INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2026IPPAX023

Thèse de doctorat

Sampling and Transport with Score-based Diffusion Models : Opportunities and Methodological Advances

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à l'École polytechnique

École doctorale n°574 École doctorale de mathématiques Hadamard (EDMH)
Spécialité de doctorat : Mathématiques appliquées

Thèse présentée et soutenue à Palaiseau, le 05/06/2026, par

MAXENCE NOBLE-BOURILLOT

Composition du Jury :

Josselin Garnier Professeur, Ecole polytechnique	Président
Eric Vanden-Eijnden Professeur, New York University	Rapporteur
Sebastian Reich Professeur, Universität Potsdam	Rapporteur (absent)
Christian A. Naesseth Professeur assistant, University of Amsterdam	Examinateur
Kamélia Daudel Professeur assistant, ESSEC Business School	Examinateur
Alain Oliviero Durmus Professeur, École Polytechnique	Directeur de thèse
Valentin De Bortoli Research scientist, Google Deepmind	Invité
Marylou Gabrié Professeur assistant, École Normale Supérieure	Invitée

A mon très cher grand-père, Bernard.

Remerciements

Mes premiers remerciements vont tout d'abord à mon directeur de thèse, Alain, qui m'a accompagné pendant ces quatre dernières années. Je te remercie d'avoir initié ce si long chemin à partir d'un simple coup de téléphone que je t'ai passé, à une époque où je ne connaissais rien ni personne : j'étais simplement curieux, désireux de tout apprendre, de tout comprendre. En quelques minutes à peine, je me suis engagé sur l'autoroute de la recherche à tes côtés, sans savoir alors que tout allait se dérouler à 200 à l'heure. Depuis notre toute première réunion au centre Lagrange — où j'ai appris à mes dépens que tu ne mangeais que rarement le midi... — jusqu'à nos échanges autour de notre dernier article soumis en conférence, où nous nous comprenions sans même avoir besoin de mots, je suis extrêmement ému par la tournure qu'a prise notre chemin commun. J'ai le sentiment d'avoir appris énormément dans mon domaine scientifique en travaillant avec toi, et d'avoir développé une méthode de travail rigoureuse, robuste à toute épreuve, dont tu es certainement l'une des plus grandes inspirations. Au regard de ces quatre ans, j'ai la sensation du devoir accompli. Moi qui me rêvais simplement titulaire d'une thèse de mathématiques appliquées, j'en ressors avec dix articles scientifiques coécrits avec toi, trois participations à des conférences internationales, un immense bagage scientifique, des réflexes scientifiques aigus, un cerveau qui fonctionne à plein régime... Et je sais que tout cela est né de la confiance que tu as placée en moi à l'issue de ce tout petit appel. Alors je te remercie sincèrement pour cela. Je te remercie aussi pour ta lucidité : celle de m'avoir poussé dans les moments où je n'avançais plus, et celle, au contraire, de m'avoir laissé autonome lorsque les choses allaient mieux. Récemment encore, tu m'as surpris par cette même lucidité lorsque j'étais traversé par des doutes. Si notre collaboration a su se maintenir, c'est grâce à cet équilibre d'encadrement, ni trop distant ni trop étouffant, que tu as su trouver et qui m'a parfaitement convenu tout au long de ma thèse. J'ai l'impression que nous nous sommes bien trouvés, que l'alchimie a bien opéré, et j'en suis extrêmement ravi. Honnêtement, je n'aurais pu espérer une meilleure thèse, et je t'en suis très reconnaissant. J'espère t'avoir rendu fier en tant que ton tout premier thésard officiel.

J'aimerais aussi adresser un remerciement tout spécial aux membres de mon jury. Tout d'abord, mes rapporteurs : thank you, Sebastian, for taking the time to review my manuscript — I know it was quite heavy, sorry for that. It means a lot to me that you are my reviewer, since I have developed great admiration for your work and for your team and collaborators since our paths crossed in Cambridge. Merci aussi à toi, Eric : je suis extrêmement honoré de t'avoir comme rapporteur, ayant été profondément inspiré par tes travaux importants sur les modèles génératifs tout au long de ma thèse. J'espère que notre future collaboration pourra aboutir à des travaux d'aussi grande qualité. Merci aussi aux autres membres du jury, dont la présence de chacun a une signification toute particulière pour moi. Josselin, tu as été mon tout premier professeur de mathématiques appliquées à l'X, en 1A, en 2019 ; celui dont la pédagogie m'a donné envie de poursuivre dans cette science particulière. Il me semblait naturel de boucler ce cycle de sept ans en t'incluant dans le jury. Kamélia, tu as été ma source d'inspiration francophone lorsque j'étais en toute première année de doctorat, de passage à Oxford. Je me souviens très bien des conseils que tu avais pu me donner à ce moment-là, et qui m'ont été bien utiles. Finally, Christian, whom I met at this famous "un-conference" at HEC and then several times at international conferences, I really appreciate your line of work, between theoretical and practical advancement, which is exactly the path I am trying to follow. Merci à tous.tes.

Mon voyage doctoral n'aurait pas pris une si belle allure stationnaire sans l'aide de quelques étoiles pour guider ma route. En tout premier lieu, il y a Valentin, le fameux "Schrödinger bridge guy", qui joua le rôle d'intermédiaire entre Alain et moi au tout début de ma thèse, et qui m'a lui aussi fait confiance, cette fois-ci à travers un Zoom. Tu es probablement la personne qui a eu le plus d'impact, dans mes tout premiers mois, sur ma manière d'envisager la recherche à long terme : soigner l'esthétique et la clarté des articles, me donner les clés pratiques d'une addiction à arXiv, me pousser à suivre le plus de chercheurs possible sur Twitter, me transmettre la passion immodérée du Schrödinger bridge... Je t'admire énormément pour ta capacité à rendre simples des choses compliquées, et je te remercie de m'avoir tant apporté ; l'une des plus belles récompenses de

ma thèse aura été de me faire surnommer “mini-Valentin”. Je regrette simplement de ne pas avoir davantage collaboré avec toi — en même temps, comment faire quand il y a Google dans la balance — mais je ne désespère pas pour la suite ! La deuxième étoile de cette thèse, c'est Marylou, qui est la chercheuse avec laquelle j'ai le plus collaboré après Alain. Grâce à ton regard pragmatique de physicienne, tu m'as apporté une nouvelle perspective sur la recherche en échantillonnage, beaucoup plus pratique, qui a énormément développé mon intuition sur les objets mathématiques utilisés. Tu m'as énormément inspiré par la rigueur et l'éthique de ta recherche, ainsi que par la grande qualité pédagogique que tu mets dans chacun de tes talks. Je suis très heureux d'avoir pu collaborer avec toi, et j'espère pouvoir encore contribuer aux éditions SALT si l'occasion se présente.

Viennent ensuite les étoiles qui en ont guidé bien d'autres avant moi, et qui forcent le respect par leur persistance à rayonner. Tout d'abord, il y a Eric, aka “l'entremetteur”, celui grâce à qui cette thèse a vu le jour puisqu'il m'a orienté vers Alain et Valentin en m'indiquant, je cite, “qu'on allait former une super équipe”. C'est aussi Eric qui, lors de ma 2A à l'X, m'a mis sur le chemin du ML grâce au cours MAP433, sobrement intitulé “Statistiques” : une véritable révélation pour moi, un enfer pour mes camarades. C'est à l'issue de ce cours que j'ai su que j'allais faire une thèse, et je n'ai jamais changé d'avis. Merci d'avoir toujours participé de loin au bon déroulement de ma thèse, et d'avoir des idées aussi saugrenues que de traverser la forêt le soir sans lumière pour rentrer d'HEC à Paris... Et puis, il y a Arnaud, l'homme aux mille intuitions : merci de m'avoir accueilli à Oxford, et de m'avoir mis dans la boucle des diffusion models, qui sont devenus mon objet d'étude préféré. Merci de m'avoir fait participer à une multitude d'événements publics qui m'ont permis de rencontrer d'excellents chercheurs ; merci aussi pour ta sincérité et ton enthousiasme systématique autour d'une bière.

Enfin, il y a mon étoile du Berger, celle qui fut là au tout début, avant même que la thèse ne commence, celle qui m'a mis sur la route de la recherche scientifique : je parle bien évidemment d'Aymeric. Encore une fois, un simple Zoom début 2021, entre un chercheur reconnu et un étudiant sans compétence particulière, mais animé d'une grande curiosité. Toi aussi, tu as su voir quelque chose en moi et tu m'as fait confiance, ce qui a abouti à ce que je considère encore comme le plus grand braquage scientifique de ma toute jeune carrière : un rapport de stage de M1, transformé à la hâte en article scientifique, accepté en conférence internationale, et finalement cité plus de 200 fois à ce jour ! Au-delà de cette curieuse aventure que fut ce travail, je tiens à te remercier tout particulièrement, Aymeric, pour avoir cru en moi et m'avoir permis de démarrer. Tu m'as également permis de prendre mes marques au CMAP lors de mon tout premier passage au CIRM à Marseille, une expérience enchantée au cœur de la nature, qui reste à ce stade l'un de mes meilleurs souvenirs de ce que peut offrir la recherche. Je n'étais que stagiaire, mais tu me faisais sentir ton égal. J'espère que tu ne m'en veux pas de ne pas avoir fait de thèse avec toi; si tu avais travaillé sur des modèles génératifs, j'y serais allé sans hésiter ! Tu es l'un des chercheurs avec lesquels j'ai le plus réussi à parler de tout sauf de maths — environnement, santé mentale, politique... — sans difficulté et avec la plus grande envie, et cela compte beaucoup pour moi. Merci sincèrement pour ça ; je suis sûr que nos chemins ne seront jamais vraiment lointains.

Mais si cette thèse a aussi bien marché, c'est sans aucun doute grâce à toi, Louis. Tu es arrivé à un moment très particulier, où je sentais peser sur moi la fameuse solitude du chercheur, et où j'avais besoin d'un renouveau. Tu as été le visage de cette nouvelle phase. Depuis notre rencontre à ce fameux reading group, où nous nous sommes simplement dit “tiens, il y a quelque chose à faire”, nous ne nous sommes jamais lâchés, et cela me rend de plus en plus heureux chaque jour qui passe. Tu as été présent à tous les moments charnières de ma thèse, véritable pilier inébranlable face à toutes les épreuves que j'ai traversées. Je pense que notre obsession commune pour le goût du travail bien fini, l'envie de tout comprendre et le désir de réaliser des choses impactantes se sont parfaitement emboîtés. Des moments que nous avons partagés ensemble, je garde tout : les nuits interminables en Zoom, les coups de téléphone de plusieurs heures, les périples pédestres et gustatifs en conférence, notre façon de refaire le monde, ta manière d'être sûr d'un postulat avant d'être convaincu du contraire le lendemain... Sans toi, l'expérience de la thèse aurait été bien moins stimulante, sur tous les plans. Je ne me lasse pas de nos échanges où nous ne sommes pas d'accord, et où l'on découvre finalement que l'un ou l'autre avait raison ; je pense que c'est l'une des raisons pour lesquelles notre duo a si bien fonctionné, en avançant pas à pas tout en confrontant nos idées. Nous avons tellement appris ensemble, c'est inimaginable. Aujourd'hui encore, je ne peux m'empêcher de lire un nouvel article sans vouloir te demander ton avis ensuite, tant il m'est précieux. Je suis extrêmement fier d'avoir partagé le chemin de la thèse avec toi ; mon admiration à ton égard n'a cessé de grandir, pour ton éthique, ta force de travail et tes compétences techniques, et je suis extrêmement heureux que tout cela soit aujourd'hui reconnu à sa juste valeur. Bien que tu sois actuellement à plusieurs milliers de kilomètres, je sais que nous continuerons à papoter, de science comme de la vie, et que nous continuerons à faire plein de choses ensemble, animés par cette envie de faire les choses bien et proprement, comme on se l'est toujours promis. Je ne pourrai jamais assez te remercier pour tout ce que tu m'as apporté.

Ensuite, je me dois de remercier celui qui est devenu, au fil du temps, mon “grand frère de thèse”, Yazid, que je regrette seulement de ne pas avoir appris à connaître plus tôt. Avec toi aussi, cette obsession mutuelle de comprendre les choses et de faire de belles maths a parfaitement fonctionné, et a rapidement trouvé sa place dans la sphère amicale. Je suis extrêmement admiratif de la façon dont tu comprends vite les concepts, et de la vitesse à laquelle tu les manipules. J'ai l'impression de toujours apprendre à tes côtés ; c'est incroyable de faire des maths avec toi. Je te remercie d'avoir répondu à toutes mes questions, même les plus bêtes, sur tous les sujets, et de m'avoir accompagné dans notre projet maudit de Schrödinger Bridge, qui aura tout de même duré plus de deux ans et demi... Tu peux revenir à la maison quand tu veux pour plancher sur des calculs horribles ! Merci aussi pour tous les conseils que tu m'as prodigués, surtout à la fin de la thèse ; ils m'ont tous été utiles. Je n'ai qu'une seule hâte : pouvoir collaborer avec toi dans le futur, écrire des papiers et profiter des conférences ensemble. Sur le même plan, j'aimerais aussi te remercier, Pierre, mon second “grand frère de thèse”. Bien que nous ayons toujours été éloignés thématiquement, nous avons réussi à construire une véritable amitié, notamment à travers ces fameux TDs du MVA du vendredi matin, qui se concluaient en beauté par un kebab. Merci d'avoir été là en dehors de la thèse, lors de multiples soirées tardives où nous nous sommes toujours bien amusés. J'admire chez toi ton goût immodéré pour la fête et le lâcher-prise, que je ne trouve nulle part chez un autre chercheur, et que je jalouse parfois. Merci à vous deux, Yazid et Pierre, pour avoir été présents lors de ma dernière année de thèse, sûrement la plus solitaire de toutes ; merci de m'avoir traité comme votre égal, merci pour ces Picon-bières, ces Sonny's pizzas, ces discussions à n'en plus finir sur le petit monde de la recherche jusqu'au bout de la nuit. Chacun de ces moments m'a fait beaucoup de bien.

J'aimerais aussi remercier chaleureusement les autres doctorants avec qui j'ai eu l'opportunité de mener un projet commun : Giacomo, Thibaut, Marie. Merci d'avoir supporté mon caractère parfois froid et intransigeant lors de l'écriture de nos articles respectifs, merci d'avoir enduré mes allers-retours scientifiques lorsque je n'étais pas satisfait, ainsi que les Zooms tardifs lorsque la deadline approchait. Je suis sûr que vous réaliserez de beaux travaux de recherche par la suite.

J'aimerais adresser un remerciement général à tous mes collègues du CMAP. En premier lieu, ceux que j'ai rencontrés il y a bien longtemps, en 2021, alors que je n'étais qu'en stage de M1, notamment lors de la retraite SIMPAS au CIRM : Achille — merci notamment de m'avoir encouragé à me lancer dans la thèse —, Corentin, Guillaume, Clément, Margaux, Manon, Benjamin. J'adresse un remerciement tout particulier à Constantin et Baptiste, qui furent mes “grands frères de stage” et m'ont permis de me sentir extrêmement bien intégré dans le groupe de recherche ; merci pour vos discussions-débats hors maths, sur tous les sujets possibles, c'était toujours une bouffée d'air frais. Merci à ceux que j'ai ensuite rencontrés à l'X dès le début de ma thèse : Jessie, Antoine, Maxime, Wassim, Orso, Grégoire, Armand, Jean... et j'en oublie sûrement d'autres. Merci de m'avoir toujours bien accueilli, même si je ne venais pas souvent à Palaiseau. Enfin, il y a les rencontres plus récentes : Mahmoud, Renaud, Louis, Dario, Antoine, Samson, toutes des personnes aussi sympathiques que douées en recherche, à qui je souhaite le meilleur pour la suite de leur parcours. Au-delà des doctorants, j'aimerais remercier particulièrement les chercheurs permanents du CMAP — Rémi, Mahdi, Luiz — ainsi que du CSD — Kimia, Bruno, Gabriel, Giulio —, qui m'ont beaucoup encouragé, ont toujours montré une grande bienveillance à mon égard, et dont la force de volonté et le talent m'impressionnent énormément et m'ont inspiré tout au long de ma thèse.

J'aimerais également remercier une multitude de groupes de jeunes chercheurs, avec qui j'ai développé de très belles amitiés, et qui m'ont tous fait grandir grâce à leur bienveillance pendant ma thèse. Tout d'abord, merci à l'équipe du centre Lagrange — Louis, Tom, Mehdi, Pablo, Thomas, Valentin, Gabriel —, qui fut d'un précieux soutien au tout début de ma thèse, lorsque j'avais tant de choses à apprendre. Viennent ensuite les doctorantes de l'INRIA Paris-Saclay — Julia, Virginie, Julie —, avec qui j'ai partagé des moments chaleureux lors de notre passage en conférence à Vienne. Et puis, il y a également l'équipe de Neurips@Paris — Maxime, Ambroise, Edwyge, Linus, Pierre, Raphaël, Lucas, encore Virginie et Julie —, qui partage avec moi cette même envie de faire prospérer l'échange entre chercheurs et fait preuve d'une force de volonté rare pour y parvenir. J'espère pouvoir continuer avec vous à maintenir et faire grandir ce concept.

Pendant ma thèse, j'ai aussi eu l'occasion de côtoyer différents milieux de recherche en dehors de l'univers académique de l'X, de l'ENS ou de Paris-Saclay, et de faire des rencontres qui m'ont particulièrement inspiré et enrichi personnellement. Je pense notamment à l'équipe de RD de Jasper, qui m'a accueilli en césure de thèse lors de ma troisième année : Clément — un merci tout particulier à ton égard pour ton encadrement très bienveillant et tes encouragements constants —, Benjamin, Damien, Onur, Gonzalo, Urszula, Fabienne. Cette coupure de six mois dans mes projets classiques de thèse m'a permis d'explorer l'aspect industriel d'articles scientifiques que je voulais absolument comprendre, ce qui était exactement ce que je recherchais à ce moment-là. Je vous remercie de m'avoir si bien intégré et d'avoir eu confiance en moi en me donnant carte blanche,

même lorsque mon approche semblait trop théorique. I would also like to thank two people in particular whom I met thanks to Valentin — James and Emile —, two researchers I admire a lot, very kind, and whom I had the chance to meet outside the pure research scope. Thank you also to the great researchers who inspired me specifically on my research topic — Denis, Julius, Lorenz, Francisco, Carles —, whom I met on several occasions at conferences. Thank you for taking the time to answer all my questions, even the dumbest, and for showing such kind consideration toward me, even though I think I am not in your league. I want to give special thanks to Michael as well: although we really discussed only once, you gave me valuable and truthful advice, and you inspired me so much by your balance between pure talent and chillness. I would have loved doing a postdoc with you under other geopolitical circumstances. Finally, thank you to all the great researchers I met at conferences, who were kind enough to answer all my questions — Joey, Alex, Kirill, Rob —, as well as the talented PhD students I met through Louis, and with whom I shared friendly moments — Jiajun, Tony, Luhuan.

Au-delà de l'univers de la recherche, je me dois de remercier tous mes proches, qui m'ont vu prendre le chemin de la thèse et m'y épanouir. En premier lieu, ma famille : Papa, Maman, Emmeline, Aurianne, Ambroise, Annabelle, Papé, Mano, et bien sûr Papy, à qui je dédie cette thèse. Je vous remercie de m'avoir soutenu dans ma démarche académique, même si vous n'en comprenez pas forcément le contenu ou la forme ; je suis fier de tenir le rôle de "scientifique" de la famille. Merci notamment à mon père et à ma mère, qui ont posé les jalons de ma toute jeune carrière scientifique en m'inscrivant en prépa et en m'encourageant fortement pendant cette période ; merci ensuite de m'avoir fait confiance pour la suite, une fois que j'ai intégré l'X et que je savais quel chemin je voulais emprunter.

Viennent ensuite mes ami.e.s, qui furent présents aux étapes les plus importantes. Tout d'abord, celle de l'adolescence: merci à mes ami.e.s de très longue date — Éléonore, Jean, Mag, Maude, Gabrielle, Pauline, Raphaël, Pierre, Suzanne, Tanisha, Abel, June, Jeanne, Martin, Vincent —, qui sont là depuis si longtemps et qui, je l'espère, m'accompagneront encore après. Merci de m'avoir sorti de mon caractère si sérieux et de m'avoir montré qu'il était possible de se réaliser et de s'épanouir autrement que par le travail ; je me rends compte désormais à quel point cela fut précieux. Merci à mes amis de l'X, qui furent présents lors de mon tout premier contact avec la recherche en 2021 — Antoine B., Antoine D., Julien, Tristan. J'ai adoré cette colocation à Champs-sur-Marne pendant le confinement, où régnait un subtil mélange de rigueur scientifique et de détente absolue ; merci de m'avoir laissé mettre mon bureau dans le salon. Cet article à 200 citations, c'est le vôtre aussi ! Merci à mes colocataires de Fresnes pendant la période du MVA — Cyril, Pierre, Clément — : bien que ce fût une période de ma vie très intense en travail, passer cette année avec vous m'a énormément épanoui et m'a laissé beaucoup de très bons souvenirs. Merci aussi à Alice, qui m'a accompagné sur une partie de mon chemin, en trouvant les mots justes pour m'encourager lorsque ça n'allait pas, et en me faisant sortir maintes fois de mes obsessions de recherche quand celles-ci prenaient trop de place. Merci à mes ami.e.s de mon quotidien parisien : Hanane, pour ces moments de détente entre voisins ; Matteo, Maggy, Victor, pour me faire oublier les maths pendant les claquettes ; et Emmanuelle, pour ces séances de cinéma auxquelles je tiens tant pour me vider la tête. Merci aussi aux ami.e.s de l'X qui sont resté.e.s autour de moi depuis la sortie de l'école — Kanza, Nissim, Léonie — et qui me procurent beaucoup de joie chaque fois que je les revois. Merci à toi, Arthur, d'être présent depuis la prépa, de partager avec moi la passion du cinéma et de discuter pendant des heures de tout sauf de maths. Merci à toi, Julien B., d'être aussi obsessionnel que moi, et de partager cette envie commune de mettre de l'énergie dans les maths et le sport — désolé qu'on ne se comprenne pas plus souvent sur les objets mathématiques, cela viendra peut-être un jour. Merci à mes ami.e.s nageurs.es. — Elie, Marius, Gabriel, Luca, Henri, Lucie, Julien A., Enzo, Tristan, Gaspard —, qui m'ont soutenu tout au long de cette thèse grâce à leur gentillesse et leur bienveillance; je vous dois beaucoup.

J'aimerais adresser un remerciement tout particulier à ceux qui ont vécu cette thèse au plus proche de moi, dans ma vie de tous les jours. Merci à toi, Tristan, mon ultime colocataire, qui as passé en tout quatre ans à vivre à mes côtés ; merci d'avoir respecté mon rythme de travail parfois chaotique, merci de m'avoir encouragé dans les moments les plus durs, merci aussi de m'avoir rendu la vie plus légère tout au long de notre cohabitation : cela me fut d'une très grande aide.

Merci à toi aussi, Gaspard, mon frère : merci pour tes discussions tardives, tes élucubrations intellectuelles, tes débats sémantiques... Je te remercie sincèrement de me tirer constamment de ma tendance à m'enfermer dans mon obsession scientifique lorsque celle-ci devient trop forte. Sans toi, je pense que je serais devenu ce cliché du chercheur incapable de se mêler au vrai monde, obnubilé par l'idée constante de réussir. J'espère pouvoir t'apporter autant que ce que tu m'apportes. Promis, je vais essayer de diminuer le rythme après la thèse, et de ne plus chercher à maximiser tout ce que je touche du doigt. Rassure-toi, je pense avoir trouvé ce que j'étais venu chercher, et je compte bien en profiter dorénavant.

Enfin, il y a toi, Sofia, qui es arrivée dans le tout dernier axe de ma thèse et qui, pourtant, as eu un impact extrêmement bénéfique. J'aimerais te remercier de m'accepter tel que je suis, et non tel que tu voudrais que je sois, avec ma folie, mon obsession et mon intransigeance. Cette dernière année de thèse fut sûrement la plus intense en matière de quantité de travail, et dans chacun des moments les plus durs, tu as été là, avec une gentillesse sans limites, alors même que je pouvais être mutique ou irascible. Sincèrement, je ne pense pas que j'aurais aussi bien tenu ces derniers mois si tu n'avais pas été là, et je t'en suis extrêmement reconnaissant. Merci de me soulager lorsque j'ai des doutes, et de partager tous ces petits moments hors du temps, hors des maths, juste toi et moi, comme ce que nous avons vécu en Corse. Je te promets que nous en aurons d'autres, et plus encore. Merci, tout simplement, d'être là.

Si cette expérience de doctorat s'est réalisée pour le mieux, sans encombre majeure ni difficulté infranchissable, c'est grâce à vous toutes et tous, chacun à votre manière et à des degrés divers. Comme je l'ai dit très tôt à Alain, alors empli d'une assurance à toute épreuve : "si j'en avais l'occasion, je ferais bien cinq ans de thèse à l'anglo-saxonne" ; au terme de ces quatre magnifiques années, mon sentiment n'a pas changé d'un iota. Cette fin de thèse, c'est aussi l'achèvement d'une période de sept ans passés à l'X, bien plus que ce à quoi je m'attendais. Tourner la page de l'X, c'est aussi m'engager dans une nouvelle phase dont je ne connais ni les contours ni le contenu ; même si je suis empli de doutes, je suis sûr que ce nouveau cycle, celui du monde "en dehors", sera des plus stimulants, grâce à tout ce que l'X m'a donné. Je n'en serai jamais assez reconnaissant.

Contents

1 Introduction **1**
1.1 Introduction générale . . . 1
1.2 General introduction . . . 3
1.3 Sampling . . . 4
1.3.1 General framework . . . 5
1.3.2 Importance Sampling . . . 6
1.3.3 Variational Inference . . . 7
1.3.4 Markov Chain Monte Carlo . . . 9
1.3.5 Annealed Sampling . . . 13
1.4 Diffusion models . . . 16
1.4.1 Diffusion models for generative modeling . . . 16
1.4.2 Diffusion models for sampling . . . 20
1.5 Optimal Transport . . . 26
1.5.1 Static formulation . . . 26
1.5.2 Dynamical formulation . . . 29
1.6 Outline and contributions of this thesis . . . 33

Bibliography **42**

I On Fundamental Challenges in Sampling **69**

2 Unbiased constrained sampling with Self-Concordant Barrier Hamiltonian Monte Carlo **71**
2.1 Introduction . . . 71
2.2 Setting and Background . . . 72
2.2.1 Riemannian Manifold Hamiltonian dynamics . . . 72
2.2.2 Self-concordance and regularity . . . 73
2.3 The n-BHMC algorithm . . . 74
2.3.1 Hamiltonian integrators of n-BHMC . . . 74
2.3.2 Steps of the algorithm . . . 75
2.3.3 About the usefulness of the "involution checking step" . . . 76
2.4 Theoretical results . . . 77
2.4.1 From implicit to numerical integrators . . . 77
2.4.2 Reversibility results . . . 77
2.5 Related work . . . 78
2.6 Numerical experiments . . . 79
2.7 Discussion . . . 80

3 Improving the evaluation of samplers on multi-modal targets **81**
3.1 Introduction . . . 81
3.2 What is hard in sampling from multi-modal distributions? . . . 82
3.3 Towards a systematic experiment for evaluating samplers . . . 82
3.3.1 Description of the experimental setting . . . 83
3.3.2 Evaluation on various samplers . . . 83
3.4 Perspectives . . . 87

II Sampling methodologies based on diffusion models 89

4 Stochastic Localization via Iterative Posterior Sampling 91
4.1 Introduction . . . 91
4.2 Background on SL for sampling . . . 92
4.3 SL with flexible denoising scheduling . . . 93
4.3.1 A more general observation process . . . 93
4.3.2 Hyper-parameter selection and SNR profile . . . 94
4.4 SLIPS sampling algorithm . . . 95
4.4.1 Handling the discretization error . . . 95
4.4.2 Estimating the denoiser via MCMC . . . 96
4.4.3 Duality of log-concavity is all you need . . . 97
4.4.4 Implementation of SLIPS. . . . 98
4.5 Related work . . . 99
4.6 Numerical experiments . . . 100
4.7 Discussion . . . 101

5 Learned Reference-based Diffusion Sampling for multi-modal distributions 103
5.1 Introduction . . . 103
5.2 Reference-based Diffusion Sampling . . . 104
5.2.1 Theoretical framework . . . 104
5.2.2 Reference-based Diffusion Sampling in practice . . . 106
5.3 Learned RDS for multi-modal distributions . . . 107
5.3.1 On the importance of the reference distribution . . . 107
5.3.2 Gaussian Mixture Model LRDS . . . 108
5.3.3 Energy-Based Model LRDS . . . 108
5.4 Related works . . . 110
5.5 Numerical evaluation of RDS . . . 110
5.6 Discussion . . . 112

6 Diffusion-based Annealed Boltzmann Generators : benefits, pitfalls and hopes 115
6.1 Introduction . . . 115
6.2 Background . . . 117
6.2.1 Diffusion models . . . 117
6.2.2 Standard Monte Carlo & Annealed sampling . . . 119
6.3 Diffusion-based aMC as a Boltzmann Generator backbone : benefits and pitfalls . . . 122
6.3.1 Of the interest of diffusion-based density paths . . . 122
6.3.2 Review of existing diffusion-based aMC-BGs . . . 123
6.3.3 Current approaches fail due to first-order approximations of DM stochastic dynamics . . . 123
6.4 Exploiting deterministic transitions of DMs in aMC methods : a new hope ? . . . 124
6.4.1 General methodology . . . 124
6.4.2 The key components needed for efficient implementation . . . 125
6.4.3 Empirical comparison between DM-based stochastic and deterministic transitions in aMC samplers . . . 127
6.5 Related Works . . . 128
6.6 Numerical experiments in a realistic setting . . . 130
6.6.1 Log-density and score learning framework . . . 130
6.6.2 DM-BGs via aMC seem inherently limited by log-density approximation . . . 130
6.7 Conclusion & Limitations . . . 134

III Optimal transport methodologies based on diffusion models 135

7 Tree-based Diffusion Schrödinger Bridge with applications to Wasserstein Barycenters 137
7.1 Introduction . . . 137
7.2 Background and setting . . . 138
7.3 Tree-based Diffusion Schrödinger Bridge . . . 140
7.4 Theoretical properties of mIPF . . . 142
7.5 Application to Wasserstein barycenters . . . 143

7.6 Related work . . . 143
7.7 Experiments . . . 144
7.8 Discussion . . . 146

8 Twisted Schrödinger Bridge Matching : a unified continuous- and discrete-time framework 147
8.1 Introduction . . . 147
8.2 Background . . . 149
8.3 Continuous-time Twisted Schrödinger Bridge . . . 150
8.3.1 Introduction to the Twisted Schrödinger Bridge . . . 150
8.3.2 Computation of the Reciprocal projection . . . 151
8.3.3 Computation of the Markovian projection . . . 152
8.3.4 Comparison between TSBM and GSBM . . . 153
8.4 Discrete-time Twisted Schrödinger Bridge . . . 154
8.4.1 Setting and ideal IMF scheme . . . 154
8.4.2 Computation of the Markovian projection . . . 156
8.4.3 Computation of the Reciprocal projection . . . 157
8.5 Conclusion & Perspectives . . . 159

Appendices 161

A Appendix of Chapter 2 163
A.1 Useful facts and lemmas . . . 163
A.2 Details on Riemannian metrics . . . 164
A.3 Properties of self-concordance . . . 165
A.4 C-BHMC: Algorithm and Results . . . 167
A.5 Proofs of Appendix A.4 . . . 168
A.5.1 Existence, uniqueness and explosion time of Hamiltonian dynamics . . . 168
A.5.2 Proof of reversibility in Algorithm 2 . . . 171
A.6 Numerical integration in HMC . . . 174
A.7 Proofs of Section 2.4.1 . . . 174
A.8 Modification of n-BHMC algorithm with step-size conditioning . . . 187
A.9 Proofs of Section 2.4.2 . . . 188
A.9.1 Expression and properties of $r^\star$ . . . 188
A.9.2 Markov kernels of Algorithm 1 . . . 190
A.9.3 Proof of reversibility in Algorithm 3 . . . 190
A.10 Comparison with Kook et al. [2022]. . . . 194
A.10.1 General framework . . . 194
A.10.2 Theoretical gaps in the reversibility of CRHMC Kook et al. [2022] . . . 195
A.11 Experimental details . . . 196

B Appendix of Chapter 3 197
B.1 Details on considered samplers . . . 197
B.2 Implementation details . . . 200

C Appendix of Chapter 4 203
C.1 Preliminaries . . . 203
C.2 Details on Stochastic Localization and our extension . . . 205
C.2.1 Reminders and intuition on generalized stochastic localization . . . 205
C.2.2 Theoretical results. . . . 205
C.2.3 Alternative denoising approach to `SLIPS` . . . 209
C.3 Detailed computations for Gaussian mixtures . . . 210
C.4 Sampling via Stochastic Localization in an ideal setting . . . 211
C.4.1 Exploring new integration schemes . . . 212
C.4.2 Studying the impact of the SNR-adapted time discretization . . . 213
C.4.3 Studying the impact of the Gaussian approximation in the initialization . . . 214
C.4.4 Studying the impact of the number of discretization steps . . . 214
C.5 Theoretical results on the duality of log-concavity . . . 214
C.6 Details on `SLIPS` algorithm . . . 218
C.6.1 Details on the implementation of `SLIPS` . . . 218

C.6.2 Ablation study . . . 219
C.7 Details on related work . . . 219
C.7.1 Reverse Diffusion Monte Carlo Huang et al. [2024a] . . . 221
C.7.2 Multi-Noise Measurements sampling Saremi and Srivastava [2022], Saremi et al. [2023, 2024] 222
C.8 Details on numerical experiments . . . 223
C.8.1 The failure of MCMC methods when targeting multi-modal distributions . . . 223
C.8.2 Algorithms and hyper-parameters . . . 223
C.8.3 Target distributions and metrics . . . 225
C.8.4 Empirical complexity of SLIPS . . . 226

D Appendix of Chapter 5 **229**
D.1 Pseudo-codes of RDS-based algorithms . . . 229
D.2 Preliminaries . . . 230
D.3 Details on noising diffusion processes . . . 232
D.3.1 General noising scheme . . . 232
D.3.2 Pinned Brownian motion . . . 233
D.3.3 Variance-Preserving diffusion . . . 233
D.3.4 Additional examples of noising diffusion processes . . . 234
D.4 Variational objectives for diffusion-based methods . . . 235
D.4.1 Continuous time formulation . . . 235
D.4.2 Discrete time formulation . . . 237
D.4.3 General RDS setting . . . 237
D.4.4 DIS setting . . . 241
D.4.5 CMCD setting . . . 243
D.5 Annealed MCMC samplers . . . 244
D.6 Energy-based Models . . . 249
D.6.1 Single-Level EBMs . . . 249
D.6.2 Multi-level EBMs . . . 249
D.7 Additional metrics for variational diffusion-based methods . . . 251
D.7.1 'Reverse' performance criteria . . . 252
D.7.2 'Forward' performance criteria . . . 253
D.8 Implementation details . . . 254
D.8.1 Details on target distributions . . . 254
D.8.2 Computational comparison of the presented methods . . . 256
D.8.3 Hyper-parameters of each sampling algorithm . . . 257
D.9 Additional results . . . 258
D.9.1 Link between the target and the reference processes . . . 258
D.9.2 Optimal setting of isotropic Gaussian reference distribution . . . 258
D.9.3 Failure of local MCMC samplers on multimodal distributions . . . 259
D.9.4 Limitations of Gaussian Mixture Models . . . 260
D.9.5 Further experimental results . . . 260

E Appendix of Chapter 6 **269**
E.1 Preliminaries . . . 269
E.1.1 Useful lemmas . . . 269
E.1.2 Discrete time setting for diffusion models . . . 270
E.1.3 General results on SDE/ODE integration . . . 270
E.1.4 Review of score and energy matching methods . . . 271
E.2 Details on (de)noising diffusion processes . . . 274
E.2.1 General noising scheme . . . 274
E.2.2 Variance-Preserving diffusion . . . 276
E.2.3 Variance-Exploding diffusion . . . 278
E.3 Proofs of Section 6.4 . . . 281
E.4 Experimental details . . . 282
E.4.1 Target details . . . 282
E.4.2 Training and sampling parameters . . . 283
E.4.3 Additional targets and metrics . . . 284
E.4.4 Ablation studies . . . 308

F Appendix of Chapter 7 **315**
F.1 Additional notation . . . 315
F.2 Introduction to trees . . . 315
F.3 Background on martingale problems . . . 317
F.4 Theoretical results on Tree Schrödinger Bridges . . . 317
F.4.1 Proofs of Section 7.3 . . . 318
F.4.2 Proofs of Section 7.4 . . . 319
F.4.3 Proof of Section 7.5 . . . 329
F.4.4 Comparison with Haasler et al. [2021] . . . 331
F.5 Further results on TreeSB . . . 333
F.6 Algorithmic techniques . . . 334
F.7 Additional experimental results and details . . . 335
F.7.1 General experimental setup . . . 335
F.7.2 Details on the experiments . . . 337

G Appendix of Chapter 8 **347**
G.1 Continuous-time setting . . . 347
G.1.1 Results on twisted measures . . . 347
G.1.2 Results on the Reciprocal projection . . . 349
G.1.3 Results on the Markovian projection . . . 351
G.2 Discrete-time setting . . . 353
G.2.1 Results on twisted measures . . . 353
G.2.2 Results on the Twisted SB problem . . . 354
G.2.3 Results on the Markovian projection . . . 355

H Fast, faithful and photorealistic diffusion-based image super-resolution with enhanced Flow Map models **357**
H.1 Introduction . . . 358
H.2 Related works . . . 359
H.3 Preliminaries on Flow Map models . . . 360
H.3.1 Background on Flow Matching . . . 360
H.3.2 Background on Flow Map models . . . 361
H.3.3 Generalization of CFG to the Flow Map setting . . . 362
H.4 Methodology . . . 364
H.4.1 Base implementation of FlowMapSR . . . 364
H.4.2 Enhancing FlowMapSR via positive-negative prompting guidance . . . 365
H.4.3 Enhancing FlowMapSR via LoRA-based adversarial fine-tuning . . . 367
H.5 Experiments . . . 368
H.5.1 Experimental setup . . . 368
H.5.2 Main results . . . 369
H.5.3 Ablation studies . . . 371
H.6 Conclusion . . . 374
H.7 Supplementary material : Theoretical details . . . 375
H.7.1 Background on Flow Matching and Flow Map models . . . 375
H.7.2 Generalization of CFG to the Flow Map setting . . . 376
H.8 Supplementary material : Implementation details . . . 377
H.8.1 General settings . . . 377
H.8.2 Training procedure . . . 377
H.8.3 Unsuccessful enhancement strategies . . . 378
H.9 Supplementary material : Additional results . . . 379

Acronyms

AIS Annealed Importance Sampling.
aMC Annealed Monte Carlo.

BM Bridge Matching.

CE Continuity Equation.

DBC Detailed Balance Condition.
DDIM Denoising Diffusion Implicit Model.
DM Diffusion Model.
DSB Diffusion Schrödinger Bridge.
DSBM Diffusion Schrödinger Bridge Matching.
DSM Denoising Score Matching.

EBM Energy-Based Model.
EI Exponential Integration.
EmOT multi-marginal EOT.
EOT Entropy-regularized Optimal Transport.
ESS Effective Sample Size.

FM Flow Matching.
FPE Fokker-Planck Equation.

GMM Gaussian Mixture Model.
gSB Generalized Schrödinger Bridge.
GSBM Generalized Schrödinger Bridge Matching.

HMC Hamiltonian Monte Carlo.

i.i.d. independent and identically distributed.
IM Implicit Midpoint.
IMF Iterative Markovian Fitting.
IPF Iterative Proportional Fitting.
IS Importance Sampling.

KL Kullback-Leibler.

LV Log-Variance.

MALA Metropolis-adjusted Langevin.
MC Monte Carlo.
MCMC Markov Chain Monte Carlo.
MD Molecular Dynamics.
MH Metropolis-Hastings.

NF Normalizing Flow.

ODE Ordinary Differential Equation.
OT Optimal Transport.

PDE Partial Differential Equation.
PT Parallel Tempering.

RE Replica Exchange.
RMHMC Riemannian Manifold Hamiltonian Monte Carlo.

SB Schrödinger Bridge.
SDE Stochastic Differential Equation.
SI Stochastic Interpolant.
SMC Sequential Monte Carlo.
SNR Signal-to-Noise Ratio.
SOC Stochastic Optimal Control.

ULA Unadjusted Langevin Algorithm.

VI Variational Inference.

WB Wasserstein Barycenter.

Chapter 1

Introduction

This introduction presents the main scientific themes developed throughout the manuscript. We begin by discussing the broader context in which this thesis is situated, with a particular emphasis on the opportunities offered by the widespread use of score-based diffusion models, notably in sampling and optimal transport. Then, we provide an overview of the three core topics studied during the Ph.D. (sampling, diffusion models, and optimal transport) highlighting the main existing methodologies in each area and their interdependence. Finally, we conclude with a summary of the chapter contents, outlining the contributions of this thesis.

1.1 Introduction générale

Dans de nombreuses applications réelles, les observations de phénomènes physiques ne sont pas purement aléatoires, mais sont plutôt supposées être des réalisations d'une distribution de probabilité sous-jacente, qui encapsule la variabilité et l'incertitude ; dans ce cadre, les observations sont appelées *échantillons*. Nous considérons ici deux cadres complémentaires. Soit la distribution est décrite par une fonction de densité explicite, c'est à dire une fonction à valeurs réelles qui quantifie la vraisemblance d'une observation donnée sous la distribution ; soit la distribution n'est accessible qu'au travers d'un grand ensemble d'échantillons déjà tirés. Une tâche fréquente consiste alors à générer de nouveaux échantillons. Dans le premier cadre, cette tâche est traditionnellement appelée *échantillonnage* ; dans le second, on parle de *modélisation générative*. Ci-dessous, nous illustrons chacune des ces deux perspectives à l'aide d'un cas d'usage représentatif.

Un exemple fondamental d'échantillonnage apparaît dans l'étude de la liaison protéine-ligand, dans la perspective de découvrir de nouveaux médicaments. L'objectif est de quantifier la force avec laquelle une molécule candidate se lie à une protéine cible, puisque l'affinité de cette liaison est directement liée à son efficacité pratique. En termes thermodynamiques, cette affinité est capturée par le concept d'*énergie libre*, qui résume à la fois la favorabilité des interactions moléculaires et le nombre de configurations microscopiques compatibles avec celles-ci. À une constante additive près, l'énergie libre peut s'exprimer comme l'opposé du logarithme d'une densité de probabilité. Par conséquent, générer des conformations moléculaires plausibles (celles de faible énergie libre, ou de façon équivalente, de forte densité) peut être vu comme un problème d'échantillonnage. En pratique, un système protéine-ligand solvaté contient typiquement entre 50,000 et 100,000 atomes, soit entre 150,000 et 300,000 degrés de liberté positionnels, en faisant un problème de très haute dimension. De plus, l'ensemble lié est souvent multi-modal : plusieurs poses de liaison distinctes ou conformations locales peuvent coexister. Des exemples concrets incluent des systèmes où deux poses liées sont traitées comme des états concurrents, conduisant à une distribution bi-modale ; le défi pratique consiste alors à obtenir des échantillons issus de chaque mode avec des proportions relatives correctes, bien que celles-ci soient inconnues a priori.

L'outil computationnel standard pour générer des échantillons dans ce contexte est appelé "Molecular Dynamics" (MD). Cette technique simule des mouvement atomiques en intégrant numériquement des équations dynamiques censées converger vers la distribution d'équilibre associée au paysage d'énergie libre. Sous cet angle, la MD peut être vue comme un cas particulier au sein de la famille plus large des méthodes de Monte Carlo. La difficulté principale provient des barrières d'énergie libre, des obstacles énergétiques qu'il faut franchir pour passer d'une configuration métastable à une autre. Le franchissement de ces barrières peut nécessiter des réarrangements coordonnés du ligand, des chaînes latérales de la protéine et du solvant environnant, rendant ces transitions rares. Or, observer de telles transitions est essentiel pour estimer les poids relatifs des modes et, in

fine, caractériser correctement les probabilités conformationnelles. Comme la MD utilise typiquement des pas de temps de l'ordre de la femtoseconde, atteindre des microsecondes de temps simulé peut nécessiter des milliards d'étapes d'intégration et des milliers d'heures-GPU, tandis que des transitions biologiquement pertinentes peuvent survenir sur des échelles de temps encore plus longues [Shaw et al., 2010]. Ce coût computationnel élevé est souvent irréaliste en pratique, ce qui motive le développement de méthodes d'échantillonage plus rapides, adaptées à la multi-modalité et à la haute dimension.

En ce qui concerne la modélisation générative, un exemple majeur en biologie computationnelle est le séquençage de l'ARN à l'échelle de la cellule ("single-cell RNA sequencing" en anglais, abrégé en "scRNA-seq"). Il fournit des clichés instantanés en haute dimension de milliers à millions de cellules à des temps successifs, capturant la diversité cellulaire au sein d'un tissu et révélant des états rares ou transitoires qui seraient invisibles dans des mesures moyennées. Cependant, le processus de mesure est essentiellement destructif : une fois une cellule séquencée, elle ne peut plus être mesurée, de sorte qu'il est impossible de suivre une même cellule au cours du temps. Ce que l'on observe n'est donc pas une collection de trajectoires individuelles, mais des distributions de plus en plus changeantes d'états cellulaires au fil du temps. Une question naturelle apparaît alors : comment la population à un temps initial se transforme-t-elle en la population observée ultérieurement ?

Le Transport Optimal (OT) est particulièrement pertinent car il formalise précisément cette question au niveau des distributions : il cherche une manière générale d'apparier la "masse" d'une distribution précoce à une distribution plus tardive tout en favorisant des changements petits et progressifs plutôt que des sauts irréalistes, respectant ainsi la géométrie de l'espace d'expression génique. En pratique, toutefois, la modélisation est difficile et computationnellement coûteuse aux échelles modernes de scRNA-seq. Des méthodes telles que Waddington-OT [Schiebinger et al., 2019] (démontrée sur environ 315,000 cellules sur 18 jours) nécessitent de résoudre de grands problèmes de transport en présence de bruit, de dynamiques avec embranchements et de tailles de population variables (puisque les cellules se divisent). À mesure que le nombre de cellules augmente, les calculs deviennent coûteux à la fois en mémoire et en temps de calcul. Ainsi, bien que l'OT fournisse un cadre mathématique rigoureux, le faire passer à l'échelle de manière robuste et l'intégrer à des contraintes biologiques réalistes demeure non trivial en pratique.

Au cours de la dernière décennie, la modélisation générative profonde a connu des progrès remarquables, portés par les avancées de l'apprentissage profond à grande échelle, tant en milieu académique qu'industriel. Sur des modalités aussi diverses que les images, l'audio et le texte, les modèles génératifs ont atteint une fidélité et une diversité sans précédent. En particulier, les modèles de diffusion fondés sur le score [Ho et al., 2020, Song et al., 2021b], ou simplement modèles de diffusion, déployés dans des systèmes à grande échelle tels que Stable Diffusion [Rombach et al., 2022] pour la génération d'image, ont montré que des distributions complexes et multi-modales peuvent être apprises et échantillonnées efficacement à partir de données disponibles.

Conceptuellement, les modèles de diffusion apprennent à transformer progressivement du bruit en données structurées au travers d'une succession de petites étapes de débruitage, apprenant ainsi à inverser un processus stochastique de corruption. Ce cadre est naturellement relié à la fois à l'échantillonnage et au transport optimal. En échantillonnage moléculaire, les modèles de diffusion peuvent être entraînés sur des données structurées afin d'apprendre des distributions d'équilibre complexes, accélérant potentiellement l'exploration de l'espace des configurations et gérant la multi-modalité sans simuler explicitement chaque transition rare. Lorsque des données suffisamment représentatives sont disponibles, ils peuvent offrir une voie guidée par les données pour contourner certaines limitations en échelle de temps des approches MD traditionnelles. Dans le cadre du transport optimal, les modèles de diffusion définissent implicitement une transformation continue de distributions de probabilité, transportant la masse d'une distribution de référence simple vers une distribution cible. Une question naturelle consiste alors à guider ou contraindre ce transport appris afin qu'il ne soit pas seulement plausible, mais optimal au regard d'un coût biologiquement ou physiquement pertinent. En ce sens, les modèles de diffusion constituent un socle computationnel prometteur, sur lequel peuvent être construits à la fois de l'échantillonnage accéléré et du transport optimal spécifique.

Cette thèse vise à contribuer à résoudre des formulations génériques d'échantillonnage et de transport optimal en développant des méthodes computationnelles fondées sur les modèles de diffusion. Dans la Partie I, qui inclut les Chapitres 2 et 3, nous mettons en évidence des défis fondamentaux de l'échantillonnage. Dans le Chapitre 2, nous abordons le défi des contraintes géométriques et proposons une méthode de Monte Carlo avancée, assortie de garanties théoriques. Dans le Chapitre 3, nous proposons une analyse détaillée des difficultés de l'échantillonnage de distributions multi-modales, fondée sur une métrique interprétable permettant d'évaluer les performances des algorithmes d'échantillonnage dans ce régime. Dans la Partie II, qui inclut les Chapitres 4 à 6, nous proposons plusieurs méthodologies d'échantillonnage basées sur les modèles de diffusion, conçues spécifiquement pour traiter la multi-modalité. Le Chapitre 4 présente une méthode ne nécessitant pas d'apprentissage automatique

et reposant uniquement sur des techniques de Monte Carlo, applicable dans un cadre général. Le Chapitre 5 introduit une méthode fondée sur l'apprentissage, construite autour d'un objectif variationnel qui exploite des informations additionnelles sur la distribution cible, telles que la localisation de zones de fortes densités. Enfin, le Chapitre 6 propose une méthodologie permettant d'intégrer un modèle de diffusion pré-entraîné au sein d'un algorithme de Monte Carlo avancé afin d'obtenir des échantillons non biaisés et de réduire l'erreur statistique induite par l'étape d'apprentissage. Dans la Partie III, qui inclut les Chapitres 7 et 8, nous présentons des méthodologies utilisant les modèles de diffusion comme colonne vertébrale pour résoudre des problèmes de transport optimal. Ces deux chapitres resposent tous deux sur le concept du pont de Schrödinger : le Chapitre 7 le reformule comme un problème de transport optimal multi-marginal régularisé, avec le calcul de barycentres de Wasserstein comme application principale, tandis que le Chapitre 8 étudie son extension dans le cadre de problèmes de contrôle optimal stochastique avec contraintes marginales aux extrémités et fonctions de coût d'état non triviales, pour la modélisation d'inférence de trajectoires.

1.2 General introduction

In many real-world applications, observations are not assumed to be purely random, but rather realizations of an underlying probability distribution, which encapsulates variability and uncertainty; in this setting, the observations are referred to as *samples*. We consider two complementary frameworks. Either the distribution is specified through an explicit density function, *i.e.*, a real-valued function that quantifies how likely a given observation is under the distribution; or the distribution is only accessible through a large collection of samples already drawn from it. A common goal is then to generate additional samples. In the first framework, this task is traditionally called *sampling*; in the second, it is referred to as *generative modeling*. We illustrate both perspectives with one representative examples for each below.

A concrete example of sampling arises in protein-ligand binding, a central problem in drug discovery. The aim is to quantify how strongly a candidate drug molecule binds to a target protein, since binding affinity is directly related to practical efficacy. In thermodynamic terms, this affinity is captured by the *free energy*, which summarizes both the favorability of molecular interactions and the number of microscopic configurations consistent with them. Up to an additive constant, free energy can be expressed as the negative logarithm of a probability density. Consequently, generating likely molecular conformations (those with low free energy, equivalently high density) can be viewed as a sampling problem. In practice, a solvated protein-ligand system typically contains $50,000$ to $100,000$ atoms, *i.e.*, $150,000$ to $300,000$ positional degrees of freedom, placing the problem firmly in a high-dimensional regime. Moreover, the bound ensemble is often multi-modal: several distinct binding poses or local conformations may coexist. Concrete examples include systems where two bound poses are treated as competing states, yielding a bi-modal distribution; the practical challenge is to recover samples from each mode with correct relative proportions, which are not known a priori.

The standard computational tool for generating samples in this setting is Molecular Dynamics (MD), which simulates atomic motion by numerically integrating dynamical equations and is expected to converge to the equilibrium distribution associated with the free-energy landscape. In this respect, MD can be seen as a particular instance within the broader family of Monte Carlo methodologies. The key difficulty comes from *free-energy barriers*, *i.e.*, energetic obstacles that must be crossed to move from one metastable configuration to another. Crossing these barriers may require coordinated rearrangements of the ligand, protein side chains, and surrounding solvent, making transitions rare. Yet observing such transitions is essential to estimate the relative weights of the modes and, ultimately, to characterize conformational probabilities accurately. Since MD typically uses femtosecond time steps, reaching microseconds of simulated time can require billions of integration steps and thousands of GPU-hours, while biologically relevant transitions may occur on even longer timescales [Shaw et al., 2010]. This high computational cost is often infeasible in realistic settings, motivating the development of faster sampling methods tailored to multi-modality and high dimensionality.

Turning to generative modeling, a fundamental example in computational biology is single-cell RNA sequencing (scRNA-seq). It provides high-dimensional snapshots of thousands to millions of cells at successive time points, capturing cellular diversity within a tissue and revealing rare or transient states that would be invisible in averaged measurements. However, the measurement process is essentially destructive: once a cell is sequenced, it cannot be measured again, so the same cell cannot be followed over time. What we observe is therefore not trajectories of individuals, but evolving distributions of cell states across time. A natural question emerges: how does the population at an early time transform into the population observed later?

Optimal Transport (OT) is compelling precisely because it formalizes this question at the level of distributions: it seeks a principled way to match "mass" from an earlier snapshot to a later one while favoring small, gradual

changes over implausibly large jumps, thereby respecting the geometry of gene-expression space. In practice, however, modeling is challenging and computationally demanding at modern scRNA-seq scales. Methods such as Waddington-OT [Schiebinger et al., 2019] (demonstrated on around $315,000$ cells across 18 days) require solving large transport problems under noise, branching dynamics, and changing population sizes (since cells divide). As cell numbers grow, computations can become expensive both in memory and runtime. Hence, although OT provides a rigorous mathematical framework, scaling it robustly and integrating it with realistic biological constraints remains nontrivial in practice.

Over the last decade, deep generative modeling has undergone remarkable progress, driven by advances in large-scale deep learning from both academia and industry. Across images, audio, and text, generative models have achieved unprecedented fidelity and diversity. In particular, score-based diffusion models [Ho et al., 2020, Song et al., 2021b], or simply diffusion models, which have been developed in large-scale systems such as Stable Diffusion [Rombach et al., 2022] for image generation, have demonstrated that complex, multi-modal distributions can be learned and sampled efficiently from available data.

Conceptually, diffusion models learn to gradually transform noise into structured data through a sequence of small denoising steps, effectively learning how to reverse a stochastic corruption process. This framework is naturally connected to both sampling and optimal transport. In molecular sampling, diffusion models can be trained on structural data to learn complex equilibrium distributions, potentially accelerating exploration of configuration space and handling multi-modality without explicitly simulating every rare transition. When sufficiently representative data are available, they may offer a data-driven route to bypass some of the timescale limitations of traditional MD-based approaches. In the context of optimal transport, diffusion models implicitly define a continuous transformation of probability distributions, transporting mass from a simple reference distribution to a target data distribution. A natural question is then how to guide or constrain this learned transport so that it is not merely plausible, but optimal with respect to a biologically or physically meaningful cost. In this sense, diffusion models represent a promising computational backbone upon which both accelerated sampling and principled optimal transport may be built.

This thesis aims to contribute to solving generic formulations of sampling and optimal transport by developing computational methods based on diffusion models. In Part I, which includes Chapters 2 and 3, we highlight fundamental challenges in sampling. In Chapter 2, we address the challenge of geometric constraints and propose an advanced Monte Carlo method with theoretical guarantees. In Chapter 3, we provide a detailed analysis of the difficulties of sampling from multi-modal distributions, based on an interpretable metric to evaluate sampler performance in this regime. In Part II, which includes Chapters 4 to 6, we propose several diffusion-based sampling methodologies designed specifically to handle multi-modality. Chapter 4 presents a method that does not require machine learning and relies solely on Monte Carlo techniques, applicable in a general setting. Chapter 5 introduces a learning-based method built on a variational objective that exploits additional information about the target distribution, such as the location of high density regions. Finally, Chapter 6 proposes a methodology to incorporate a pretrained diffusion model into an advanced Monte Carlo algorithm in order to obtain unbiased samples and reduce statistical error arising from the learning stage. In Part III, which includes Chapters 7 and 8, we present methodologies that use diffusion models as a backbone for solving optimal transport problems. These chapters share a focus on the Schrödinger bridge: Chapter 7 reframes it as multi-marginal entropic-regularized OT, with Wasserstein barycenters as a main application, while Chapter 8 investigates its extension to the formulation of stochastic optimal control problems with endpoint marginal constraints and non-trivial state cost functions for trajectory inference modeling.

1.3 Sampling

In this section, we introduce the key concepts underlying the generic sampling problem (Section 1.3.1), together with the standard methodologies used to address it (Sections 1.3.2 to 1.3.5). This introduction is intentionally non-technical and informal: its purpose is to highlight the main challenges of sampling, to provide intuition for the mathematical objects involved throughout this manuscript, and to discuss the advantages and limitations of the presented sampling approaches. We refer the reader to Robert and Casella [2010] for a complementary overview from a more technical perspective.

Unless otherwise specified, we consider sampling problems on $(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$, where $\mathcal{B}(\mathbb{R}^d)$ denotes the Borel σ-field of $\mathbb{R}^d$, and we systematically assume that all considered probability distributions admit densities with respect to the Lebesgue measure. Accordingly, we use the same notation to refer interchangeably to a probability distribution and its density. Similarly, we adopt this abuse of notation for transition kernels.

Notation. We denote by $\mathscr{P}(\mathbb{R}^d)$ the space of probability measures defined on $(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$. For any distribution $\mu \in \mathscr{P}(\mathbb{R}^d)$ and any μ-integrable function $f : \mathbb{R}^d \to \mathbb{R}$ (referred to as a μ-test function), we denote the expectation of f with respect to μ by $\mu(f) = \int_{\mathbb{R}^d} f(x)\,\mu(\mathrm{d}x)$. For any $x \in \mathbb{R}^d$, δ_x will denote the Dirac mass at x. For any transition kernel $q : (x, \mathcal{A}) \in \mathbb{R}^d \times \mathcal{B}(\mathbb{R}^d) \mapsto q(\mathcal{A}|x) \in [0, 1]$ and any $n \geq 1$, we define recursively the distribution $q^n\mu \in \mathscr{P}(\mathbb{R}^d)$ by $q^n\mu(\mathcal{A}) = \int_{\mathbb{R}^d} q(\mathcal{A}|x)(q^{n-1}\mu)(\mathrm{d}x)$ with $q^0\mu = \mu$. For any continuously differentiable diffeomorphism $\mathrm{T} : \mathbb{R}^d \to \mathbb{R}^d$, we denote by $J_{\mathrm{T}}(x)$ the Jacobian matrix of T evaluated at x, and we denote by $\mathrm{T}_{\#}\mu \in \mathscr{P}(\mathbb{R}^d)$ the push-forward distribution of μ through T, defined by $\mathrm{T}_{\#}\mu(\mathcal{A}) = \mu(\mathrm{T}^{-1}(A))$. In particular, sampling $Y \sim \mathrm{T}_{\#}\mu$ amounts to sampling $X \sim \mu$ and setting $Y = \mathrm{T}(X)$. By the change-of-variable formula, the density of $\mathrm{T}_{\#}\mu$ is given by $\mathrm{T}_{\#}\mu(y) = \mu(\mathrm{T}^{-1}(y))\,|\det J_{\mathrm{T}^{-1}}(y)|$. We finally denote by $x_{0:K}$ the $(K+1)$-tuple $(x_0, x_1, \ldots, x_K) \in (\mathbb{R}^d)^{K+1}$.

1.3.1 General framework

What is sampling? The fundamental sampling problem can be formulated as follows: generate samples, *i.e.*, realizations, ideally independent and identically distributed (i.i.d.) from a probability distribution π, referred to as the *target distribution*, defined on $\mathbb{R}^d$ and described solely through its density, which takes the form

$$\pi : x \mapsto \frac{\gamma(x)}{\mathcal{Z}}\,, \quad \text{where } \mathcal{Z} = \int_{\mathbb{R}^d} \gamma(x)\,\mathrm{d}x\,.$$

Here, the function $\gamma : \mathbb{R}^d \to \mathbb{R}_+$ is referred to as the *unnormalized density* of π, which is assumed to be pointwise evaluable, while $\mathcal{Z}$ denotes the associated normalizing constant, which is typically intractable. This setting motivates the common statement in the sampling literature that "π is known up to a normalizing constant", an apparent difficulty that sampling methods are specifically designed to handle.

Let $\mathrm{Supp}(\pi)$ denote the support of π, *i.e.*, the set of points $x \in \mathbb{R}^d$ such that $\pi(x) > 0$. We may define the (unnormalized) *log-density* of π as $x \in \mathrm{Supp}(\pi) \mapsto \log \gamma(x)$, or equivalently its (unnormalized) *energy* function $x \in \mathrm{Supp}(\pi) \mapsto -\log \gamma(x)$. In most settings, γ is assumed to be at least once continuously differentiable, which allows one to define the associated *score* function $x \in \mathrm{Supp}(\pi) \mapsto \nabla \log \gamma(x)$. Importantly, this quantity is uniquely defined with respect to π and does not depend on the normalizing constant.

The canonical Bayesian example. A standard application of sampling arises in Bayesian statistics [Liu and Liu, 2001, Kroese et al., 2011], where the goal is to infer an unknown parameter x assumed to have generated observed data y, while explicitly quantifying uncertainty. Prominent applications include signal processing, chemistry, and biology. More specifically, the Bayesian framework assumes access to a statistical model that specifies how y is generated from x through a tractable *likelihood* function $p(y \mid x)$, together with *prior* knowledge about x encoded in a tractable prior distribution $p(x)$. Bayesian inference consists in updating this prior belief using the observed data, *i.e.*, sampling from x given y via Bayes' rule, which defines the target *posterior* distribution $p(x \mid y)$ with unnormalized density

$$\gamma(x) = p(x)\,p(y \mid x)\,.$$

In this setting, the normalizing constant $\mathcal{Z} = p(y)$, known as the *evidence*, quantifies how well the model explains the data and is intractable.

Beyond Bayesian applications, target distributions may exhibit specific mathematical structures. In this manuscript, we focus in particular on two such properties: constrained distributions and multi-modal distributions, as discussed below.

Geometric constraints are ubiquitous. A particularly important case arises when $\mathrm{Supp}(\pi)$ is a constrained subset of $\mathbb{R}^d$. Such situations are common in biological applications, where constraints are imposed through mathematical formulations of real-world considerations; see, for instance, Lewis et al. [2012], Thiele et al. [2013]. Typical constraints are expressed through systems of equality and inequality relations. When these constraints are linear, the support takes the form of a polytope, which constitutes the main class of targets considered in Chapter 2. Sampling from such distributions requires balancing two competing objectives regarding $\mathrm{Supp}(\pi)$: efficient exploration of high-density regions and strict confinement to this constrained subspace, in order to avoid inefficient or biased sampling. Achieving this balance is challenging for standard samplers, which are often designed under the assumption that $\mathrm{Supp}(\pi) = \mathbb{R}^d$. A substantial body of work has therefore focused on incorporating geometric information into classical sampling methods to adapt them to constrained settings, as detailed in Section 1.3.4. However, such algorithmic modifications may introduce significant bias, which must in turn be quantified and corrected, making constrained sampling particularly delicate.

Multi-modality is ubiquitous. Another fundamental class of target distributions encountered throughout scientific applications, notably in statistical mechanics [Krauth, 2006] and molecular dynamics [Stoltz et al., 2010], consists of *multi-modal* distributions. These are distributions for which the unnormalized density γ admits multiple local maxima, referred to as *modes*, located in distinct regions of the space and separated by low-probability regions. Importantly, these modes may differ substantially in their relative importance, commonly referred to as their *weights*. For example, π may describe the plausibility of molecular conformations under given external conditions such as temperature or pressure, in which case several configurations may be plausible, with markedly different probabilities.

For multi-modal targets, two complementary aspects are crucial in sampling: (A) *local accuracy*, meaning the ability to explore individual modes effectively, and (B) *global accuracy*, meaning the ability to represent the relative weights of the modes correctly in the generated samples. As discussed in the following subsections, classical sampling methods developed over the past several decades are generally effective at achieving (A), but often struggle with (B), leading to misrepresentation of mode weights. An extreme manifestation of this issue is *mode-blindness*, where all modes are locally well sampled but appear with identical weights in the output. This difficulty stems from the fact that mode weights are typically unknown to the user: the density γ may be available only as a black box or be too complex to analyze directly. As a result, samplers must recover global information about mode weights using only pointwise evaluations of γ or related quantities, even though these weights correspond to integrals of γ over disjoint regions of space. The challenge is further exacerbated in applications featuring a large number of modes, highly non-uniform weight distributions, and pathological structures such as regions of zero-probability mass separating certain modes.

In this manuscript, we place particular emphasis on the challenge of multi-modality in sampling. Despite its ubiquity, this challenge remains relatively ill-defined and largely unexplored from a general methodological perspective, with most existing work focusing on problem-specific targets. For instance, there is currently no standard sampling benchmark designed to assess the robustness of algorithms across different regimes of multi-modality. This thesis aims to help bridge the gap between practitioners and method developers by addressing the following question.

(**Q1**) What fundamental difficulties does multi-modality introduce for sampling? Can these difficulties be quantified numerically? How can the ability of sampling algorithms to handle multi-modal distributions be evaluated experimentally in an interpretable manner? We address those questions in Chapter 3.

In the following subsections, we review the main classes of classical sampling methods, which will serve as background for the contributions presented in Parts I and II. At a high level, we distinguish two broad categories: *zeroth-order* methods (Sections 1.3.2 to 1.3.4), which rely solely on pointwise evaluations of the unnormalized density γ, and *first-order* methods (Sections 1.3.4 and 1.3.5), which additionally require evaluations of the score $\nabla \log \gamma$, when it is well defined.

1.3.2 Importance Sampling

Principle. Importance Sampling (IS) is one of the earliest sampling methodologies, with roots dating back more than fifty years [Goertzel, 1949]. It assumes access to an auxiliary probability distribution $\nu \in \mathscr{P}(\mathbb{R}^d)$, referred to as the *proposal*, which satisfies the following conditions: (i) $\mathrm{Supp}(\pi) \subset \mathrm{Supp}(\nu)$, (ii) its (un)normalized density is exactly known and can be evaluated pointwise, and (iii) it is straightforward to sample from ν.

Given such a proposal, one defines the importance weight function $w : x \in \mathrm{Supp}(\nu) \mapsto \pi(x)/\nu(x)$, and observes that, for any π-test function f, the change-of-measure formula $\pi(\mathrm{d}x) = w(x)\nu(\mathrm{d}x)$ yields

$$\pi(f) = \int_{\mathbb{R}^d} f(x) w(x)\, \nu(\mathrm{d}x) \ .$$

Drawing N i.i.d. samples $\{X^i\}_{i=1}^N$ from ν (commonly called *particles*), and defining the empirical measure

$$\hat{\pi}_{\mathrm{IS}}^N = \frac{1}{N} \sum_{i=1}^N w(X^i)\, \delta_{X^i} \ ,$$

one obtains an unbiased Monte Carlo (MC) estimator of $\pi(f)$ given by $\hat{\pi}_{\mathrm{IS}}^N(f)$. This naturally suggests $\hat{\pi}_{\mathrm{IS}}^N$ as a particle-based approximation of π. Hence, sampling from π via IS amounts to sampling from the particle set $\{X^i\}_{i=1}^N$ according to a multinomial distribution with weights $\{w(X^i)\}_{i=1}^N$.

However, the importance weights are themselves not tractable, since the normalizing constant of π (and possibly ν) is unknown. To circumvent this issue, one instead considers the empirical distribution

$$\hat{\pi}^N_{\mathrm{SNIS}} = \frac{1}{N}\sum_{i=1}^{N} w_{\mathrm{SN}}(X^i)\,\delta_{X^i}\ ,$$

where the self-normalized (SN) weights are defined as

$$w_{\mathrm{SN}}(X^i) = \frac{w(X^i)}{\sum_{j=1}^N w(X^j)} = \frac{\tilde{w}(X^i)}{\sum_{j=1}^N \tilde{w}(X^j)}\ ,$$

where $\tilde{w}(x) = \gamma(x)/\nu(x)$, thereby only requiring evaluation of the unnormalized densities. The resulting MC estimator $\hat{\pi}^N_{\mathrm{SNIS}}(f)$ is biased, but consistent as $N \to \infty$ [Chatterjee and Diaconis, 2018].

Setting & known limitations. The central IS challenge of is to ensure sampling efficiency, *i.e.*, that the empirical approximation $\hat{\pi}^N_{\mathrm{SNIS}}$ remains close to the target distribution π for a moderate sample size N. This efficiency can be quantified through the bias and variance of the associated Monte Carlo estimators. In particular, Agapiou et al. [2017] derives worst-case bounds over the class of bounded π-test functions f, given by

$$\text{Bias:}\quad \sup_{|f|<1} \left|\mathbb{E}\big[\hat{\pi}^N_{\mathrm{SNIS}}(f) - \pi(f)\big]\right| \leq \frac{12}{N}\frac{\mathbb{E}_\nu[\tilde{w}(X)^2]}{\mathbb{E}_\nu[\tilde{w}(X)]^2} = \frac{12\exp\mathcal{R}_2(\pi\|\nu)}{N}\ , \tag{1.1}$$

$$\text{Variance:}\quad \sup_{|f|<1} \mathbb{E}\Big[\big(\hat{\pi}^N_{\mathrm{SNIS}}(f) - \pi(f)\big)^2\Big] \leq \frac{4}{N}\frac{\mathbb{E}_\nu[\tilde{w}(X)^2]}{\mathbb{E}_\nu[\tilde{w}(X)]^2} = \frac{4\exp\mathcal{R}_2(\pi\|\nu)}{N}\ , \tag{1.2}$$

where $\mathcal{R}_2(\pi\|\nu)$ denotes the 2-Rényi divergence [Rényi, 1961], quantifying the discrepancy between π and ν.

These bounds lead to two key observations:

- For a fixed sample size N, controlling the quantity $\mathbb{E}_\nu[\tilde{w}(X)^2]\,\mathbb{E}_\nu[\tilde{w}(X)]^{-2}$ entirely determines both the bias and variance of the SNIS estimator. While this quantity is not directly accessible, its inverse can be approximated using the available samples $\{X^i\}_{i=1}^N$ through the Effective Sample Size (ESS), defined by

$$\mathrm{ESS} = \frac{\left(\sum_{i=1}^N \tilde{w}(X^i)\right)^2}{\sum_{i=1}^N \tilde{w}(X^i)^2} = \left(\sum_{i=1}^N w_{\mathrm{SN}}(X^i)^2\right)^{-1} \in (0, N].$$

  In practice, this provides a computable diagnostic of IS performance, with values close to N indicating that all generated samples contribute equally, yielding best sampling performance.

- For a fixed proposal ν, the sample size N required to achieve accurate estimation grows exponentially with the divergence between ν and π.

Together, these observations imply that an effective proposal ν must concentrate its mass on the same high-probability regions as π, and ideally be as close as possible to π. This has motivated the development of adaptive IS methods, which iteratively refine the proposal distribution during the sampling procedure [Bugallo et al., 2017, Elvira and Martino, 2021]. Nevertheless, determining an appropriate sample size N remains challenging. In high dimensions, N must typically scale exponentially with d to compensate for the term $\exp\mathcal{R}_2(\pi\|\nu)$, which highlights the fundamental limitation of IS: even a mild mismatch between π and ν can require a prohibitive increase in computational effort when dimension increases, thus highlighting the curse of dimensionality.

Following the same principle as IS, Variational Inference aims to alleviate some of the evoked limitations by introducing parameterized proposal families that can be optimized to better align with π using machine learning techniques.

1.3.3 Variational Inference

Principle. Variational Inference (VI) [Jordan et al., 1998, Wainwright and Jordan, 2008, Blei et al., 2017] departs from the classical IS setting, where the proposal distribution is typically chosen to be simple, of limited complexity, and often hand-designed by the practitioner, which restricts its ability to accurately match the target distribution π. Instead, VI aims at approximating π directly in distribution space by selecting, within a parameterized family $\mathcal{P}^\Theta = \{p^\theta\}_{\theta\in\Theta} \subset \mathscr{P}(\mathbb{R}^d)$, referred to as the *variational family*, the distribution that is

closest to π with respect to a prescribed divergence D acting on $\mathscr{P}(\mathbb{R}^d) \times \mathscr{P}(\mathbb{R}^d)$. As such, VI is formulated as an optimization problem rather than a direct sampling method, but it remains closely related to IS due to the two key requirements imposed on elements of $\mathcal{P}^\Theta$: (i) each p^θ must be easy to sample from, and (ii) each p^θ must admit a tractable and normalized density.

Under these conditions, the VI optimization problem is defined as

$$\arg\min_{\theta \in \Theta} \mathrm{D}(p^\theta \mid \pi) \, .$$

Once an optimal parameter $\theta^\star$ is obtained, approximate samples from π can be generated simply by sampling from $p^{\theta^\star}$. Importantly, common choices of variational families enable *amortized sampling*, *i.e.*, the parallel generation of a large number of i.i.d. samples in a single forward pass, which explains the strong interest of VI among practitioners. Within this framework, two main sources of error may arise: an *estimation error*, due to imperfect optimization of $\theta^\star$, and an *approximation error*, stemming from the potential mismatch between π and the variational family $\mathcal{P}^\Theta$.

A second appealing aspect of VI is that it can be combined with IS. Once the variational approximation $p^{\theta^\star}$ has been obtained, it may be used as a proposal distribution within an IS scheme. While this may appear redundant at first sight, since $p^{\theta^\star}$ already provides approximate samples from π, this combination plays a crucial role in correcting estimation and approximation errors, acting as a statistical calibration mechanism for the variational model. In Chapter 6, we notably study an extension of this VI-based calibration strategy to joint distributions π and $p^{\theta^\star}$ that exhibit a Markov structure.

Setting & known limitations. When adopting VI for sampling, two design choices are central: the selection of the divergence D and the construction of the variational family $\mathcal{P}^\Theta$.

Regarding D, a common choice is the reverse Kullback-Leibler (KL) divergence defined by

$$\mathrm{KL}(p^\theta \mid \pi) = \mathbb{E}_{p^\theta}\left[\log \frac{p^\theta}{\pi}\right] = \int_{\mathbb{R}^d} \log\left(\frac{p^\theta(x)}{\gamma(x)}\right) p^\theta(\mathrm{d}x) + \log \mathcal{Z},$$

or the Log-Variance (LV) divergence [Nüsken and Richter, 2021], given by

$$\mathrm{LV}(p^\theta \mid \pi) = \mathrm{Var}_\nu\left[\log \frac{p^\theta}{\pi}\right] = \int_{\mathbb{R}^d} \left[\log\left(\frac{p^\theta(x)}{\gamma(x)}\right)\right]^2 \nu(\mathrm{d}x) - \left[\int_{\mathbb{R}^d} \log\left(\frac{p^\theta(x)}{\gamma(x)}\right) \nu(\mathrm{d}x)\right]^2,$$

where $\nu \in \mathscr{P}(\mathbb{R}^d)$ is an arbitrary auxiliary distribution. In practice, VI objectives are minimized using stochastic gradient descent[1] or related optimization schemes. A key distinction between these divergences is that, while the LV objective involves samples independent of θ, the KL objective requires sampling from p^θ, which may *a priori* complicate gradient computation. This difficulty is typically addressed using the reparameterization trick [Rezende et al., 2014, Kingma and Welling, 2014], which rewrites samples from p^θ as transformations of auxiliary random variables. A common choice for the LV divergence is to take $\nu = p^{\mathrm{sg}(\theta)}$, where sg denotes the stop-gradient operator.

Regarding the choice of $\mathcal{P}^\Theta$, a standard option consists in Gaussian distributions [Agrawal et al., 2020], or mixtures of Gaussians [Jaakkola and Jordan, 1998], which are better suited to represent multi-modal target distributions. In these families, both means and covariances are typically optimized. To reduce computational complexity, practitioners often adopt the mean-field assumption, which enforces coordinate-wise independence in the variational distribution. While this choice enables efficient gradient computation and favorable scaling with dimension, it can induce significant approximation errors by failing to capture correlations between variables.

The limitations of mean-field approximations have motivated the development of more expressive variational families with tractable densities, among which the Normalizing Flow (NF) [Rezende and Mohamed, 2015] is particularly prominent. NFs construct variational distributions as push-forwards of a simple base distribution, typically a standard Gaussian $\rho = \mathrm{N}(0, \mathrm{I}_d)$, through a parameterized continuously differentiable diffeomorphism $\mathrm{T}^\theta : \mathbb{R}^d \to \mathbb{R}^d$, *i.e.*, we have $p^\theta = \mathrm{T}^\theta_\# \rho$. Sampling from p^θ is then straightforward, since a realization is obtained by setting $x = \mathrm{T}^\theta(z)$ with $z \sim \rho$. Importantly, as soon as the Jacobian matrix of the inverse map $(\mathrm{T}^\theta)^{-1}$ is known and its log-determinant can be easily computed, the density of p^θ is tractable, as guaranteed by the change-of-variables formula. To satisfy this technical requirement, T^θ is chosen within specific neural

[1] Note that the constant $\log \mathcal{Z}$ expressed in the KL divergence does not affect VI gradient-based optimization, since it appears as an additive constant and is therefore canceled out when taking the gradient with respect to θ.

network architectures that allow both efficient inversion and computation of the Jacobian determinant, such as RealNVP [Dinh et al., 2017] or Masked Autoregressive Flows [Papamakarios et al., 2021]. In practice, NFs provide substantially greater expressiveness than standard variational families, even for multi-modal targets, while retaining stable training properties. The combination of VI and IS for this class of models has led to the extensive literature on *Boltzmann generators* [Noé et al., 2019].

Despite sustained efforts to enhance the expressiveness of variational families, a fundamental limitation of the VI framework persists, particularly in multi-modal settings. This limitation is linked to the *mode-seeking* behavior of divergences such as KL or LV, which structurally encourage p^θ to assign negligible probability mass to regions where π is small or vanishes (*i.e.*, between modes). As a result, the variational approximation tends to concentrate on a single mode of π, rather than distributing mass across all modes, even when p^θ is itself multi-modal. This phenomenon, known as *mode collapse*, is well documented in the VI literature [Regli and Silva, 2018, Jerfel et al., 2021]. Although alternative divergences, such as α-divergences, have been proposed to mitigate this issue [Hernandez-Lobato et al., 2016, Daudel et al., 2023], mode collapse remains a major limitation of VI in practice.

1.3.4 Markov Chain Monte Carlo

Principle. Markov Chain Monte Carlo (MCMC) methods constitute one of the most fundamental algorithmic approaches for obtaining approximate samples from a target distribution π, and play a central role in many of the contributions presented in this thesis. They are grounded in the concept of the Markov chain, namely a discrete-time stochastic process $\{X^n\}_{n\in\mathbb{N}}$ defined through a transition kernel $q : (x, \mathcal{A}) \in \mathbb{R}^d \times \mathcal{B}(\mathbb{R}^d) \mapsto q(\mathcal{A} \mid x) \in [0, 1]$, referred to as a Markov kernel, which satisfies the Markov property

$$\mathbb{P}(X^{n+1} \in \mathcal{A} \mid X_0 = x_0, X_1 = x_1, \dots, X_n = x_n) = q(\mathcal{A} \mid x_n) \, ,$$

for any measurable set $\mathcal{A} \in \mathcal{B}(\mathbb{R}^d)$ and any $(x_0, x_1, \dots, x_n) \in (\mathbb{R}^d)^{n+1}$. In particular, Markov chains are entirely characterized by their associated transition kernel. Informally, given the state x_n at step n, the next state at step $n+1$ depends only on x_n and is conditionally independent of the past history. As a result, simulating a Markov chain amounts to iteratively sampling from the kernel q, which is typically straightforward to implement.

In contrast to IS and VI, which generate approximate samples from π in parallel, MCMC methods are inherently sequential. The objective is to construct a Markov chain whose empirical distribution $\hat{\pi}^N = \frac{1}{N}\sum_{i=1}^N \delta_{X^i}$ converges to π as $N \to \infty$. Formally, this convergence is expressed through the consistency condition $\hat{\pi}^N(f) \to \pi(f)$ for any π-test function f, starting from an initial distribution $X_0 \sim \nu$, where $\nu \in \mathscr{P}(\mathbb{R}^d)$ may hold certain assumptions related to π. Such convergence is for instance guaranteed by Birkhoff's ergodic theorem provided that π is the unique invariant distribution of q, that is, $q\pi = \pi$. Ensuring invariance with respect to π is therefore a cornerstone of MCMC design. A sufficient condition for invariance is *reversibility* with respect to π, formulated via the Detailed Balance Condition (DBC), defined by

$$q(\mathrm{d}y \mid x)\, \pi(\mathrm{d}x) = q(\mathrm{d}x \mid y)\, \pi(\mathrm{d}y) \, .$$

While constructing a reversible kernel may initially appear challenging, Metropolis et al. [1953] introduced a simple and general mechanism to achieve this goal. The resulting procedure, known as Metropolis-Hastings (MH) calibration, transforms any proposal kernel q into a π-reversible kernel q^{MH} defined as

$$q^{\mathrm{MH}}(\mathrm{d}y \mid x) = \alpha^{\mathrm{MH}}(x, y)\, q(\mathrm{d}y \mid x) + \left(1 - \int_{\mathbb{R}^d} \alpha^{\mathrm{MH}}(x, y)\, q(\mathrm{d}y \mid x)\right) \delta_x(\mathrm{d}y) \, ,$$

where the acceptance probability $\alpha^{\mathrm{MH}} : \mathbb{R}^d \times \mathbb{R}^d \to [0, 1]$ is given by

$$\alpha^{\mathrm{MH}}(x, y) = \min\left(1, \frac{q(x \mid y)\, \pi(y)}{q(y \mid x)\, \pi(x)}\right) = \min\left(1, \frac{q(x \mid y)\, \gamma(y)}{q(y \mid x)\, \gamma(x)}\right) \, .$$

Crucially, this acceptance probability does not depend on the unknown normalizing constant $\mathcal{Z}$, which makes the procedure fully tractable in practice. Simulation under q^{MH} is implemented via an accept–reject step: a proposal $\tilde{y} \sim q(\cdot \mid x)$ is generated, then accepted with probability $\alpha^{\mathrm{MH}}(x, \tilde{y})$, or rejected in which case the chain remains at x. Many classical MCMC algorithms rely on this calibration mechanism. When the proposal kernel is symmetric, meaning $q(y \mid x) = q(x \mid y)$, the acceptance probability simplifies to the density ratio $\gamma(y)/\gamma(x)$.

The MH calibration can also be applied to deterministic proposals. If $q(\cdot \mid x) = \delta_{\mathrm{T}(x)}$ for a diffeomorphism $\mathrm{T} : \mathbb{R}^d \to \mathbb{R}^d$ satisfying the involution condition, *i.e.*, $\mathrm{T} \circ \mathrm{T} = \mathrm{Id}$, then the acceptance probability becomes

$$\alpha^{\mathrm{MH}}(x) = \min\left(1, \frac{\mathrm{T}_{\#}\pi(x)}{\pi(x)}\right) = \min\left(1, \frac{\gamma(\mathrm{T}(x))\,|\det J_{\mathrm{T}}(x)|}{\gamma(x)}\right).$$

When the map T is additionally volume preserving, the Jacobian determinant equals one and the acceptance probability thus reduces to the density ratio $\gamma(\mathrm{T}(x))/\gamma(x)$.

Although convergence guarantees are asymptotic, practical performance of MCMC methods hinges on how rapidly this convergence occurs. This is quantified through the mixing time, defined for a tolerance level $\varepsilon > 0$, as

$$\tau^{\varepsilon} = \inf\left\{n \in \mathbb{N} : \|P^n \nu - \pi\|_{\mathrm{TV}} \leq \varepsilon\right\} ,$$

where $\|\cdot\|_{\mathrm{TV}}$ denotes the total variation distance

$$\|\nu_1 - \nu_2\|_{\mathrm{TV}} = \sup_{A \in \mathcal{B}(\mathbb{R}^d)} |\nu_1(A) - \nu_2(A)| .$$

Smaller mixing times correspond to faster convergence of the Markov chain toward π. This criterion is extensively analyzed in the theoretical literature on Markov chains as a measure of sampling efficiency.

Zeroth order designs. Early MCMC designs relied on the Metropolis-Hastings mechanism and used only evaluations of the unnormalized target density γ together with simple proposal kernels q. A canonical example is Random-Walk Metropolis-Hastings (RWMH) [Metropolis et al., 1953], where $q(\cdot|x) = \mathrm{N}(x, \lambda \mathrm{I}_d)$, which enables local exploration around the current state x through the hyperparameter $\lambda > 0$, and for which the acceptance ratio simplifies due to the symmetry of q. In particular, the optimal MH acceptance rate is discussed in Gelman et al. [1997] for a broad class of distributions, enabling principled adjustment of the hyperparameter λ within the sampling algorithm. Another prominent example is Independent Metropolis-Hastings (IMH) [Gilks et al., 1995], where $q(\cdot|x)$ does not depend on x, which can facilitate exploration of previously unvisited regions of the space, a property that is especially valuable for multi-modal distributions. However, these basic designs face important practical limitations, as their proposal kernel is not informed by the target distribution π, which often leads to inefficient exploration, particularly in high dimensions, and consequently to very slow mixing. Recent extensions of IMH, for example, propose the use of a NF as deterministic proposal map, see e.g., Gabrié et al. [2022], based on adaptive schemes in which the NF is continuously optimized using samples generated by the Markov chain. Although better suited to multi-modal targets, this approach nevertheless suffers from the curse of dimensionality, as it remains constrained by the statistical inefficiency of IMH. These limitations have motivated the development of proposal kernels that explicitly depend on π, and more precisely on its score, as discussed next.

First-order designs. MCMC methods which exploit access to the target score are currently the primary approach for addressing sampling problems of moderate difficulty. Most algorithms in this class rely on the *overdamped* Langevin diffusion, defined by the following Stochastic Differential Equation (SDE)

$$\mathrm{d}X_t = \nabla \log \gamma(X_t)\mathrm{d}t + \sqrt{2}\mathrm{d}B_t , \tag{1.3}$$

where $(B_t)_{t\geq 0}$ denotes a standard d-dimensional Brownian motion. Under mild conditions on π, it can be shown that the invariant distribution of the resulting continuous time stochastic process $(X_t)_{t\geq 0}$ is exactly π, see for instance Roberts and Tweedie [1996]. This observation has motivated the development of a widely used sampling method, known as the Unadjusted Langevin Algorithm (ULA), obtained by discretizing SDE (1.3) using a first order Euler Maruyama scheme. This leads to the Markov kernel $q(\cdot|x) = \mathrm{N}(x + \lambda \nabla \log \gamma(x), 2\lambda \mathrm{I}_d)$, where $\lambda > 0$ is the discretization step-size of the diffusion and acts as a hyperparameter. In practice, selecting λ is challenging, highly target dependent, and may require adaptation as different regions of the space are explored. As such, ULA does not provide a principled mechanism for tuning this parameter.

This limitation has led to the design of its Metropolis-Hastings correction, known as Metropolis-adjusted Langevin (MALA) [Roberts and Tweedie, 1996], where the proposal kernel coincides with the ULA transition kernel, thus exploiting both γ and $\nabla \log \gamma$ at each step. This calibration allows for continuous adjustment of λ over time by targeting a prescribed acceptance rate, for which optimal values are derived in the theoretical literature [Roberts and Rosenthal, 1998]. Despite their simplicity, ULA and MALA have been shown to be practically efficient for log-concave distributions, that is when $\log \gamma$ is concave, with sharp non asymptotic

guarantees [Durmus and Moulines, 2015, Dalalyan, 2017, Durmus and Moulines, 2017]. In particular, the efficiency and simplicity of MALA motivate its use as a backbone for many more advanced sampling algorithms, see Section 1.3.5. Following this perspective, it is also extensively exploited in the methodologies developed in Part II.

Another important class of first-order sampling algorithms relies instead on the *underdamped* Langevin diffusion, given by the SDE

$$\begin{aligned} \mathrm{d}X_t &= \mathrm{M}^{-1} P_t \mathrm{d}t \\ \mathrm{d}P_t &= \nabla \log \gamma(X_t)\mathrm{d}t - \eta \mathrm{M}^{-1} P_t \mathrm{d}t + \sqrt{2\eta}\mathrm{d}B_t \ , \end{aligned} \tag{1.4}$$

which depends on two hyperparameters, namely the *mass* $\mathrm{M} \in \mathbb{R}^{d\times d}$, a positive definite matrix, and the *friction* $\eta > 0$. Intuitively, $(X_t, P_t)_{t\geq 0}$ represent the position and momentum of a particle, carrying richer dynamical information than in the overdamped case described by SDE (1.3), which corresponds to the high friction regime where $\eta \to \infty$. Under mild conditions on π, the invariant distribution of SDE (1.4) is the extended target $\bar{\pi}(x,p) = \pi(x) \otimes \mathrm{N}(p; 0, \mathrm{M})$ [Pavliotis, 2014], which marginalizes to π with respect to the position variable. In other words, the underdamped approach operates in an augmented state space and recovers samples of π through the marginal in x. Although this construction may appear counterintuitive at first, it yields smoother trajectories for $(X_t)_{t\geq 0}$ than in the overdamped setting, since the stochasticity is carried by the momentum variable. As a consequence, for log-concave targets, underdamped dynamics converge faster than overdamped dynamics, particularly in worst case regimes such as high dimension [Cheng et al., 2018].

A notable special case of underdamped Langevin dynamics arises when $\eta = 0$, corresponding to the absence of friction. In this regime, SDE (1.4) reduces to an Ordinary Differential Equation (ODE)

$$\begin{aligned} \mathrm{d}X_t &= \partial_p H(X_t, P_t)\mathrm{d}t = \mathrm{M}^{-1} P_t \mathrm{d}t \\ \mathrm{d}P_t &= -\partial_x H(X_t, P_t)\mathrm{d}t = \nabla \log \gamma(X_t)\mathrm{d}t \ , \end{aligned} \tag{1.5}$$

with the *Hamiltonian* energy function $H(x,p) = -\log \gamma(x) + \frac{1}{2} p^\top \mathrm{M}^{-1} p$ that verifies $\bar{\pi}(x,p) \propto \exp[-H(x,p)]$. Importantly, the Hamiltonian is invariant under momentum reversal, that is $H(x,p) = H(\mathrm{s}(x,p))$, where $\mathrm{s} : (x,p) \mapsto (x,-p)$ denotes the momentum reversal operator. The ODE (1.5), referred to as *Hamiltonian dynamics*, plays a central role in many areas of physics. It enjoys three key properties: (a) Hamiltonian dynamics is volume preserving, (b) reversing time is equivalent to flipping the sign of the momentum, that is considering the dynamics of $\mathrm{s}(X_t, P_t)$, due to s-invariance of the Hamiltonian, (c) the Hamiltonian energy is preserved along trajectories, that is $\mathrm{d}H(X_t, P_t)/\mathrm{d}t = 0$, which implies that $\bar{\pi}$ remains invariant under the dynamics and motivates its use in MCMC.

Given a fixed time window $[0,T]$ with horizon $T > 0$, let $\mathrm{T} : (X_0, P_0) \to (X_T, P_T)$ denote the flow associated with Hamiltonian dynamics, obtained by exact integration of ODE (1.5) on $[0,T]$. As a consequence, we may define the composite map $\mathrm{s} \circ \mathrm{T}$, that is involutive by property (b), and use it as a proposal map in the deterministic MH algorithm. In particular, by property (a), we have $|\det J_{\mathrm{s}\circ\mathrm{T}}| \equiv 1$, since $|\det J_{\mathrm{T}}| \equiv 1$ (we say that T is symplectic), and property (c) together with invariance under momentum reversal yields $H((\mathrm{s}\circ\mathrm{T})(x,p)) = H(x,p)$. As a result, the associated MH acceptance rate is constant, equal to 1, leading to systematic acceptance of proposals. The resulting MH algorithm corresponds to the ideal version of the widely used Hamiltonian Monte Carlo (HMC) method [Duane et al., 1987, Brooks et al., 2011].

This ideal setting is in fact unattainable in practice, since the flow T cannot be computed exactly and must be approximated using numerical integration schemes, see for example [Hairer et al., 2006, Chapter VI]. To remain faithful to the ideal case, practical HMC implementations rely on numerical integrators that preserve the key structural properties of the Hamiltonian flow, namely symplecticity and involution up to momentum reversal, inherited from properties (a) and (b) of the Hamiltonian dynamics. Let $\tilde{\mathrm{T}}$ denote a numerical approximation of T satisfying these properties. Using $\mathrm{s}\circ\tilde{\mathrm{T}}$ as a proposal map leads to the practical Metropolis-Hastings acceptance probability

$$\alpha^{\text{MH-HMC}} : (x,p) \mapsto \min\Big(1, \exp[-H(\tilde{\mathrm{T}}(x,p)) + H(x,p)]\Big) \ ,$$

since numerical integration no longer guarantees exact conservation of the Hamiltonian.

To construct such a numerical map $\tilde{\mathrm{T}}$, integration is performed segment by segment on $[0,T]$. Given an increasing sequence of time points $\{t_k\}_{k=0}^K$ with $t_0 = 0$, $t_K = T$, and $K \geq 1$, the interval $[0,T]$ is first partitioned into sub-intervals $I_k = [t_k, t_{k+1}]$ of width $\delta_k = t_{k+1} - t_k$. Numerical integration is then applied sequentially on each

I_k to build a sequence $\{X_k, P_k\}_{k=0}^K$ approximating (X_{t_k}, P_{t_k}), and $\tilde{\mathrm{T}}(X_0, P_0)$ is defined as (X_K, P_K). A widely used integrator in this context is the generalized Leapfrog method [Hairer et al., 2006, Theorem 3.4], also known as the Störmer Verlet integrator, which reads on I_k as

$$\begin{aligned} P_{k+1/2} &= P_k - \frac{\delta_k}{2}\partial_x H(X_k, P_{k+1/2}) \\ X_{k+1} &= X_k + \frac{\delta_k}{2}\left(\partial_p H(X_k, P_{k+1/2}) + \partial_p H(X_{k+1}, P_{k+1/2})\right) \\ P_{k+1} &= P_k - \frac{\delta_k}{2}\partial_x H(X_{k+1}, P_{k+1/2})\,. \end{aligned} \tag{1.6}$$

The appeal of this integrator lies in its explicit form for separable Hamiltonians of the form $H(x,p) = U(x) + V(p)$, as in the canonical case presented here, which enables straightforward implementation.

To further enhance exploration through the momentum variable, practical HMC implementations embed the Metropolis-Hastings procedure into a Gibbs sampling scheme. Given the current state (X^n, P^n) of the Markov chain, the algorithm performs the following sequence :

1. *sampling the momentum*: $P^{n+1} \sim \bar{\pi}(\cdot|X^n) = \mathrm{N}(0, \mathrm{M})$,
2. *sampling the position*: here, the target conditional distribution is $\bar{\pi}(x|P^{n+1}) \propto \exp(-H(x, P^{n+1}))$, which can be sampled via the deterministic MH-HMC algorithm described above with starting state (X^n, P^{n+1}).

This Gibbs scheme preserves π as the marginal target and significantly reduces mixing time compared to direct Metropolis-Hastings on $\bar{\pi}$ [Neal et al., 2011]. Among existing variants, the most widely used and efficient implementation is the No U-Turn Sampler (NUTS) [Hoffman and Gelman, 2014], which adaptively tunes HMC hyperparameters during sampling.

Incorporating geometric constraints into HMC. In some sampling problems, additional information is available about the geometry of the support of the target distribution. This information is often encoded through geometric constraints that the samples are required to satisfy. A notable case arises when a *metric* $x \mapsto \mathfrak{g}(x)$ is provided, where $\mathfrak{g}(x) \in \mathbb{R}^{d\times d}$ is a positive definite matrix, endowing the state space with a Riemannian manifold structure. Such metrics can be constructed in a tractable manner for polytopes via *self concordant logarithmic barriers* [Nesterov and Nemirovski, 1998]. A natural question is how to combine this geometric information with efficient first-order sampling methods such as HMC.

The seminal work of Girolami and Calderhead [2011] provides a principled solution by reformulating HMC with a position-dependent mass matrix $M = \mathfrak{g}(x)$, recovering the standard Euclidean case when $\mathfrak{g}(x)$ is constant. The underlying idea is that the metric guides Hamiltonian dynamics in regions where constraints are active, while still steering trajectories toward high-density regions. The resulting method is known as Riemannian Manifold Hamiltonian Monte Carlo (RMHMC), with corresponding Hamiltonian given by

$$H(x,p) = -\log\gamma(x) + \frac{1}{2}\log\det\mathfrak{g}(x) + \frac{1}{2}p^\top\mathfrak{g}(x)^{-1}p\,. \tag{1.7}$$

Importantly, the general Gibbs sampling structure of HMC remains unchanged. Given the curent state (X^n, P^n) of the Markov chain, the algorithm performs the same sequence:

1. *sampling the momentum*: $P^{n+1} \sim \bar{\pi}(\cdot|X^n) = \mathrm{N}(0, \mathfrak{g}(X^n))$,
2. *sampling the position*: here, the target conditional distribution remains $\bar{\pi}(x|P^{n+1}) \propto \exp(-H(x, P^{n+1}))$, which can still be sampled via the deterministic MH-HMC algorithm with starting state (X^n, P^{n+1}) and proposal map $\tilde{\mathrm{T}}$ now approximating the Riemannian Hamiltonian dynamics.

Remarkably, the generalized Leapfrog integrator remains applicable in this setting. However, unlike the Euclidean case, it becomes **implicit**, since X_k and $P_{k+1/2}$ are defined implicitly in (1.6) due to the non-separability of the Riemannian Hamiltonian. To address this issue, Girolami and Calderhead [2011] propose solving the implicit equations using a prescribed number of fixed-point iterations. This approach has also been adopted in recent work Kook et al. [2022], which revisits RMHMC for polytopes and additionally introduces linear algebra decompositions enabling efficient computation in high dimensions. However, the theoretical analysis of these algorithms typically overlooks the fact that implicit integration schemes may admit multiple exact solutions. In practice, this ambiguity can only be further exacerbated by the additional approximation layer introduced by fixed-point iterations. This raises the question of whether such approaches may introduce bias with respect to the target distribution, as discussed below.

(**Q2**) Existing practical variants of RMHMC typically rely on the tacit assumption that numerical implicit integration schemes discretizing Riemannian Hamiltonian dynamics operate exactly, independently of problem specific hyperparameters, as in the Euclidean HMC setting. However, their implicit nature introduces additional sources of complexity that are difficult to characterize and are largely overlooked in the literature, despite their potential for additional bias. For the case of polytopes, can we empirically demonstrate the existence of such bias ? If so, how can the design of RMHMC be improved to mitigate it ? What theoretical guarantees can be established for the resulting procedure ? These questions are addressed in Chapter 2.

Curse of multi-modality. Returning to the Euclidean setting, the standard MCMC algorithms described above are known to perform very well for log-concave target distributions, supported by a substantial body of theoretical results. Their success stems from efficient local exploration of high-density regions, driven by access to the score of the target, which provides rich information about π. However, when π is multi-modal and log-concavity no longer holds, this local exploration paradigm becomes a limitation. In such cases, Markov chains tend to remain trapped within a single mode: when modes are well separated, transitions between them become rare events, leading to dramatically increased mixing times. The resulting bias manifests as accurate local exploration within individual modes, but poor global exploration, yielding incorrect estimates of relative mode weights.

To address multi-modality, one approach is to consider time inhomogeneous Markov kernels, whose invariant distribution evolves over time, starting from an easy-to-sample distribution and gradually transforming into the target. This strategy, known as *annealed sampling*, aims to alleviate mode trapping and is described below through three main methodological approaches, all of which build upon the sampling methods introduced above as fundamental components.

1.3.5 Annealed Sampling

Principle. Annealed Monte Carlo (aMC) methods form a family of sampling algorithms built upon traditional MCMC approaches that specifically aim at sampling from multi-modal distributions, where standard MCMC methods are inefficient. Overall, these methods rely on a sequence of $(K+1)$ intermediate distributions $\{p_k\}_{k=0}^K \in (\mathscr{P}(\mathbb{R}^d))^{K+1}$, with $K > 1$, that gradually interpolate between the target $p_0 = \pi$ and an easy-to-sample distribution $p_K = \pi^{\text{base}}$ (e.g., a Gaussian distribution), thereby defining an interpolation path in distribution space. For this reason, they are also referred to as *bridging* distributions. The main objective of annealing is to decompose the original sampling problem into a sequence of simpler sub-problems, for which traditional MCMC applies, starting from $k = K$ and decreasing k, thus considering intermediate targets of increasing complexity. The specificity of each aMC method lies in how smoothly and in a controlled manner transitions between successive sampling problems are performed.

To fit within the standard sampling framework, each p_k is defined up to a normalizing constant, most commonly through the *geometric interpolation* scheme

$$p_k(x) \propto \gamma(x)^{\beta_k} \left(\pi^{\text{base}}(x)\right)^{1-\beta_k} , \tag{1.8}$$

where the so-called *tempering schedule* $\{\beta_k\}_{k=0}^K$ is strictly decreasing, with $(\beta_0, \beta_1) = (1, 0)$, thereby enforcing boundary conditions on the density path. A major advantage of this construction is its computational efficiency, as it allows for simple evaluation of the intermediate scores $\{\nabla \log p_k\}_{k=0}^K$, required by first-order MCMC methods, through linear combinations of $\nabla \log \gamma$ and $\nabla \log \pi^{\text{base}}$.

Annealed Importance Sampling. A natural annealed sampling strategy is Annealed Importance Sampling (AIS) [Neal, 2001], which can be interpreted as an instance of Importance Sampling on an extended space. Specifically, the target $\pi \in \mathscr{P}(\mathbb{R}^d)$ is replaced by an extended target $\bar{\pi} \in \mathscr{P}\left((\mathbb{R}^d)^{K+1}\right)$, defined as

$$\bar{\pi}(x_{0:K}) = \pi(x_0) \prod_{i=0}^{K-1} q_{k+1|k}(x_{k+1}|x_k) , \tag{1.9}$$

where $\{q_{k+1|k}\}_{k=0}^{K-1}$ denotes a collection of transition kernels, referred to as *forward kernels*. By construction, $\bar{\pi}$ marginalizes to π in the x_0 variable. Therefore, sampling from $\bar{\pi}$ is sufficient to obtain samples from π by retaining the x_0 component of the generated samples. To remain consistent with the prescribed density path, the forward kernels are typically chosen to be p_{k+1}-reversible, and hence p_{k+1}-invariant, most commonly implemented using MALA kernels.

Similarly, the IS proposal $\nu \in \mathscr{P}(\mathbb{R}^d)$ is replaced by an extended proposal $\bar{\nu} \in \mathscr{P}\left((\mathbb{R}^d)^{K+1}\right)$, defined as

$$\bar{\nu}(x_{0:K}) = \pi^{\text{base}}(x_K) \prod_{i=0}^{K-1} q_{k|k+1}(x_k|x_{k+1}) \,, \tag{1.10}$$

where $\{q_{k|k+1}\}_{k=0}^{K-1}$ is a collection of *backward kernels*, which are required to be easy to sample from (for now left flexible). In this extended setting, applying the IS paradigm with target $\bar{\pi}$ and proposal $\bar{\nu}$ amounts to simulating N i.i.d. particle trajectories $\{X_{0:K}^i\}_{i=1}^N$ from $\bar{\nu}$, generated in a Markovian manner from $k = K$ down to $k = 0$, and constructing the empirical measure

$$\hat{\pi}_{\text{AIS}}^N = \frac{1}{N} \sum_{i=1}^N \bar{w}(X_{0:K}^i)\, \delta_{X_0^i} \,,$$

as a particle-based approximation of π, where the annealed importance weight is defined as $\bar{w} : x_{0:K} \mapsto \bar{\pi}(x_{0:K})/\bar{\nu}(x_{0:K})$. To ensure good overlap between $\bar{\pi}$ and $\bar{\nu}$, which is crucial as discussed in Section 1.3.2, backward kernels are chosen such that the marginals of $\bar{\nu}$ also align with the bridging distributions. In particular, Neal [2001] proposes to set $q_{k|k+1}$ as the same p_{k+1}-invariant kernel as $q_{k+1|k}$. As a consequence, the detailed balance condition implies

$$q_{k+1|k}(x_{k+1}|x_k) p_{k+1}(x_k) = q_{k|k+1}(x_k|x_{k+1}) p_{k+1}(x_{k+1}) \,,$$

which allows the annealed importance weight to simplify as

$$\bar{w}(x_{0:K}) = \frac{\pi(x_0)}{p_1(x_0)} \cdot \frac{p_1(x_1)}{p_2(x_1)} \cdots \frac{p_{K-1}(x_{K-1})}{\pi^{\text{base}}(x_{K-1})} \,. \tag{1.11}$$

A key advantage of this choice is that computing $\bar{w}$ does not require access to the normalizing constants of the intermediate distributions $\{p_k\}_{k=1}^{K-1}$, due to the cancellations in (1.11). The remaining intractability stems from the normalizing constant $\mathcal{Z}$ of π, which is easily handled through the standard self-normalization trick.

Despite improving upon standard importance sampling, which suffers from poor alignment between the target π and the proposal $\nu = \pi^{\text{base}}$, AIS retains a fundamental limitation, namely *weight degeneracy*. This phenomenon arises when sampled trajectories contribute unevenly to the approximation of π after reweighting, leading to weights with high variance. As the dimensionality of the state space increases, the number of particles required to counteract this effect may grow dramatically, especially in large ambient dimension d, resulting in significant inefficiency. The method presented next builds upon AIS to specifically address this limitation.

Sequential Monte Carlo. To address the inefficiency of AIS, a natural strategy is to further enforce alignment with the bridging distributions along the sequential procedure by introducing intermediate resampling steps, as proposed in Sequential Monte Carlo (SMC) [Del Moral et al., 2006]. This methodology exploits the autoregressive structure of the AIS importance weights,

$$\bar{w}(x_{0:K}) = \prod_{k=0}^{K-1} w_k(x_k) \text{ where } w_k(x) = \frac{p_k(x)}{p_{k+1}(x)} \,, \tag{1.12}$$

which naturally defines a collection of K marginal importance weight functions $\{w_k\}_{k=0}^{K-1}$. In particular, for each $k \in \{K-1, \dots, 0\}$, assuming access to particles $\{X_{k+1}^i\}_{i=1}^N$ that are accurately distributed according to p_{k+1}, simulating the next particle states in AIS via $X_k^i \sim q_{k|k+1}(\cdot|X_{k+1}^i)$ for $i \in \{1, \dots, N\}$ yields a particle-based approximation of p_k given by

$$\hat{p}_k^N = \frac{1}{N} \sum_{i=1}^N w_k(X_k^i)\, \delta_{X_k^i} \,.$$

Based on this observation, SMC improves upon AIS by additionally performing, for each $k \in \{K-1, \dots, 0\}$, a particle resampling step after the generation of $\{X_k^i\}_{i=1}^N$, according to the obtained empirical distribution $\hat{p}_k^N$ (using the self-normalized version of the importance weight w_k). The resampled particles are then assigned uniform weights, which by construction prevent weight degeneracy, by discarding particles poorly aligned with p_k (low w_k) and replicating those with good alignment (high w_k). Moreover, the resampling step at iteration k may be followed by additional MCMC moves using a p_k-invariant Markov kernel. This optional rejuvenation step preserves the form of the intermediate importance weights while further improving alignment with p_k, thereby helping to alleviate degeneracy.

In practice, more sophisticated SMC variants are employed to handle challenging multi-modal targets, see for example Chopin et al. [2020] for a comprehensive overview. For instance, one may apply adaptive resampling, *i.e.*, performing resampling only at carefully selected steps $k \in \{K-1, \dots, 0\}$, by aggregating importance weights across iterations. Commonly, resampling is triggered when the aggregated ESS falls below a predefined threshold, indicating poor performance of the underlying AIS scheme. Another approach consists in adaptive scheduling, that is selecting the tempering parameters β_k so as to maximize the ESS associated with w_k, at the cost of not being able to fix K in advance. Although these strategies mitigate weight degeneracy, they do not completely eliminate it. In practice, the most effective remedy is often to design a density path tailored to the specific target distribution, thereby departing from a fully generic formulation.

Replica Exchange. The third main annealed sampling methodology is Replica Exchange (RE), also known as Parallel Tempering (PT) [Swendsen and Wang, 1986, Geyer et al., 1991], which was originally developed in statistical physics and predates both AIS and SMC. In contrast to AIS and SMC, which operate sequentially to sample from the prescribed bridging distributions, RE proceeds in parallel. More precisely, RE also considers sampling on the extended state space $(\mathbb{R}^d)^{K+1}$, but defines the target distribution as $\bar{\pi} = \otimes_{k=0}^{K} p_k$. Unlike AIS and SMC, this construction does not induce a Markovian structure across levels, and instead focuses on designing a $\bar{\pi}$-preserving Markov kernel $\bar{q}$. By simulating a multi-state Markov chain with this kernel, one asymptotically obtains samples from π through the first level.

The design of the kernel $\bar{q}$ specifically targets multi-modality by allowing for both local and global exploration. Local exploration is achieved by associating to each level $k \in \{0, \dots, K\}$ a p_k-invariant Markov kernel q_k, typically chosen as MALA due to its efficiency. These kernels define altogether an extended Markov kernel which, for any $\mathcal{A}_{0:K} \in \mathcal{B}((\mathbb{R}^d)^{K+1})$, is given by

$$\bar{q}_{\text{local}}(\mathcal{A}_{0:K} | x_{0:K}) = \otimes_{k=0}^{K} q_k(\mathcal{A}_k | x_k) \, ,$$

and operates independently and in parallel across all levels. However, this local kernel alone is insufficient, since choosing $\bar{q} = \bar{q}_{\text{local}}$ would lead to parallel instances of standard MCMC, with limited global exploration at every annealing level.

Global exploration is enabled by combining $\bar{q}_{\text{local}}$ with a deterministic map $\bar{\mathrm{T}} : (\mathbb{R}^d)^{K+1} \to (\mathbb{R}^d)^{K+1}$ that performs *swaps* between adjacent levels k and $k+1$, for selected indices $k \in \{0, \dots, K-1\}$. To enhance communication between levels, either even or odd indices are typically considered, leading to the following composite swap maps, assuming for simplicity that K is odd,

$$\bar{\mathrm{T}}^{\text{even}}(x_{0:K}) = (x_1, x_0, x_3, x_2, \dots, x_{K-2}, x_{K-3}, x_K, x_{K-1}) \tag{1.13}$$

$$\bar{\mathrm{T}}^{\text{odd}}(x_{0:K}) = (x_0, x_2, x_1, x_4, \dots, x_{K-4}, x_{K-1}, x_{K-2}, x_K) \, . \tag{1.14}$$

Each of these maps is involutive and symplectic, and can therefore be easily incorporated into a deterministic Metropolis-Hastings procedure. Due to their structure in local swaps, each global move decomposes into $\lfloor K/2 \rfloor$ parallel deterministic MH updates, each operating locally between levels k and $k+1$, with k even for $\bar{\mathrm{T}}^{\text{even}}$ or odd for $\bar{\mathrm{T}}^{\text{odd}}$. For both cases, the corresponding acceptance probability is given by

$$\alpha_k^{\text{MH-RE}} : (x_k, x_{k+1}) \mapsto \min\left(1, \frac{p_k(x_{k+1}) p_{k+1}(x_k)}{p_k(x_k) p_{k+1}(x_k)}\right) \, ,$$

which again does not require knowledge of the normalizing constants of the intermediate distributions.

Overall, the RE algorithm alternates between applying $\bar{q}_{\text{local}}$, typically for several iterations, and performing global swap moves explained above, using either $\bar{\mathrm{T}}^{\text{even}}$ or $\bar{\mathrm{T}}^{\text{odd}}$, selected according to a Bernoulli distribution. This MCMC procedure is $\bar{\pi}$-reversible and enables efficient global exploration contrary to standard MCMC. More recently, Okabe et al. [2001], Syed et al. [2022] proposed refinements of the classical RE framework, in which $\bar{\mathrm{T}}^{\text{even}}$ and $\bar{\mathrm{T}}^{\text{odd}}$ are applied deterministically in alternation, leading to reduced mixing times. Although the resulting Markov kernel is no longer $\bar{\pi}$-reversible, it remains $\bar{\pi}$-invariant and is therefore valid.

Despite its widespread use and its reputation as a gold standard for sampling from multi-modal distributions, the performance of RE strongly depends on the success of the swap moves. In practice, this is governed by the acceptance probabilities $\{\alpha_k^{\text{MH-RE}}\}_{k:\text{ even}}$ and $\{\alpha_k^{\text{MH-RE}}\}_{k:\text{ odd}}$, which may deteriorate when the overlap between successive bridging distributions is insufficient. As in AIS and SMC, this necessitates careful tuning of the tempering schedule $\{\beta_k\}$ with respect to π, for which no generic and principled solution currently exists.

1.4 Diffusion models

Generative modeling shares a common objective with sampling: generating new samples from a target probability distribution π. While, in sampling, π is specified exactly through an unnormalized density, the situation is fundamentally different in generative modeling, where, π is only accessible through a finite collection of samples $\mathcal{D} = \{x_1, \dots, x_n\}$, assumed to be i.i.d. realizations. In real-world settings, where these samples may consist of images, text, or audio, the underlying distribution π is typically highly complex. Generative modeling therefore aims to leverage machine learning techniques to approximate π from the dataset $\mathcal{D}$ by means of a parameterized probability distribution p^θ, capable of generating new samples autonomously, ideally in an efficient manner.

Among the various approaches developed for this purpose, the framework of Diffusion Model (DM) has emerged as one of the most prominent in recent years. DMs have indeed achieved state-of-the-art performance across a wide range of data modalities [Rombach et al., 2022, Ho et al., 2022, Abramson et al., 2024], thereby triggering a surge of research activity, both in their development and in extensions beyond pure generation tasks. This thesis fits naturally within this line of work, as we focus on DM applications to sampling and optimal transport. To provide context for these research fields, we first recall the fundamental principles underlying DMs for generative modeling in Section 1.4.1. Then, we review in Section 1.4.2 their more recent use in the specific setting of sampling. We refer to Section 1.5.2 for a brief overview of DM extensions to optimal transport.

1.4.1 Diffusion models for generative modeling

Diffusion models are relatively recent, having emerged less than a decade ago, with seminal contributions Sohl-Dickstein et al. [2015], Ho et al. [2020], Song et al. [2021b] that profoundly reshaped the landscape of generative modeling. Since then, numerous follow-up works have been proposed, including notable developments on closely related techniques, commonly referred to as Flow Matching (FM) [Lipman et al., 2023, Liu et al., 2023]. In this section, we present the principles of both DM and FM within the unifying data-to-Gaussian framework originally proposed by Albergo et al. [2023], with the goal of providing a concise and self-contained overview. This perspective allows us to introduce the key theoretical elements in a streamlined manner, and naturally prepares the ground for the discussion in Section 1.4.2.

From data interpolation to generative dynamics. The core idea underlying DMs and FMs consists of three successive steps summarized below.

(1) *Interpolation*: defining a continuous-time distribution path $(p_t)_{t\in[0,T]}$ over a time interval $[0, T]$ with $T > 0$, smoothly bridging $\pi = p_0$ to an easy-to-sample base distribution $\pi^{\text{base}} = p_T$;

(2) *Markovian projection*: constructing a continuous-time Markovian dynamics on $[0, T]$ whose marginals coincide with the prescribed density path;

(3) *Inference*: generating samples from these dynamics in the backward time direction, that is from $t = T$ to $t = 0$, in order to recover samples from π at $t = 0$.

In most cases (including those studied in this thesis), π^{base} is chosen as the standard Gaussian distribution $\mathrm{N}(0, \mathrm{I}_d)$, and the family of distributions $(p_t)_{t\in[0,T]}$ required at stage (1) is induced by the non-Markovian stochastic process

$$I_t = \alpha_t X_0 + \sigma_t Z, \quad X_0 \sim \pi, \quad Z \sim \mathrm{N}(0, \mathrm{I}_d)\,, \tag{1.15}$$

where X_0 and Z are independent, $\alpha : [0, T] \to \mathbb{R}$ and $\sigma : [0, T] \to \mathbb{R}$ are smooth interpolation coefficients, commonly referred to as a noising schedule. These coefficients are chosen such that the Signal-to-Noise Ratio (SNR) function, defined by $\mathrm{SNR}(t) = (\alpha_t/\sigma_t)^2$, is strictly decreasing, with boundary behaviors $\mathrm{SNR}(t) \to \infty$ as $t \to 0$ and $\mathrm{SNR}(t) \to 0$ as $t \to T$. Intuitively, the SNR controls the relative contributions of data and noise in the interpolant over the time window $[0, T]$ (full data at $t = 0$, full noise at $t = T$), and plays a central role in the effectiveness of DMs, as shown in Chapter 4.

As a consequence, the marginal density p_t is generally intractable and is given by

$$p_t(x) = \int p_{t|0}(x \mid x_0)\, \mathrm{d}\pi(x_0)\,. \tag{1.16}$$

Here, $p_{t|0}(\cdot \mid x_0) = \mathrm{N}(\alpha_t x_0, \sigma_t^2 \mathrm{I}_d)$ denotes the conditional distribution of I_t given $X_0 = x_0$. In other words, p_t is obtained by convolving π with a Gaussian kernel whose variance increases with t, progressively destroying the data signal as noise becomes dominant.

Once the distribution interpolation has been specified, the next question is how to carry out stage (2) in order to generate new samples at stage (3), *i.e.*, how to turn the non-Markovian formulation (1.15) into a Markov process that preserves the prescribed marginals. In practice, this requires the construction of appropriate Markov kernels, starting from $t = T$ and moving towards $t = 0$, commonly referred to as "denoising" transitions. It is at this point that the DM and FM formulations diverge in their characterization of the underlying dynamics: DMs exhibit a Stochastic Differential Equation (SDE) (*i.e.*, *stochastic* denoising transitions), whereas FMs are based on a Ordinary Differential Equation (ODE) (*i.e.*, *deterministic* denoising transitions).

Close-up on Diffusion models. As observed in Karras et al. [2022], the interpolant defined in (1.15) admits the same marginal distributions as the solution of the following linear SDE, commonly referred to as the *noising* or *forward* SDE,

$$\mathrm{d}X_t = f(t)X_t\mathrm{d}t + g(t)\mathrm{d}B_t, \quad X_0 \sim \pi \ , \tag{1.17}$$

where $(B_t)_{t\in[0,T]}$ denotes a standard Brownian motion, and where the coefficients $f : [0,T] \to \mathbb{R}$ and $g : [0,T] \to \mathbb{R}_+$ satisfy

$$f(t) = \frac{\dot\alpha_t}{\alpha_t} \quad \text{and} \quad g(t) = \sqrt{2\dot\sigma_t\sigma_t - 2\frac{\dot\alpha_t}{\alpha_t}\sigma_t^2} \ . \tag{1.18}$$

In this formulation, standard choices for the noising schedule include variants of the *Variance Preserving* scheme [Ho et al., 2020, Song et al., 2021b], which enforces $\alpha_t^2 + \sigma_t^2 = 1$, the *Variance Exploding* scheme, characterized by $\alpha_t = 1$ and rapidly growing σ_t [Song et al., 2021b, Karras et al., 2022], as well as the *Cosine schedule* [Nichol and Dhariwal, 2021], which is commonly used in image generation. Overall, this construction yields a Markovian formulation that enables stochastic "noising" transitions from $t = 0$ to $t = T$ (*i.e.*, from data to noise) by simulating the SDE, while marginally matching the interpolant introduced at step (1).

To enable stage (2), one would ideally reverse the time direction in the SDE (1.17) in order to simulate the associated time-reversed process, which we denote by $(Y_t)_{t\in[0,T]}$. A classical result due to Anderson [1982] shows that, under mild regularity conditions on the noising schedule and on π, the process $(Y_t)_{t\in[0,T]}$ remains Markovian and is itself governed by an SDE, referred to as the *reverse* or *denoising* SDE, given by

$$\mathrm{d}Y_t = -\left[f(T-t)Y_t - g^2(T-t)\nabla \log p_{T-t}(Y_t)\right]\mathrm{d}t + g(T-t)\mathrm{d}W_t, \quad Y_T \sim \pi^{\mathrm{base}} \ , \tag{1.19}$$

where $(W_t)_{t\in[0,T]}$ is another standard Brownian motion. By construction, $\mathcal{L}(Y_t) = p_{T-t}$, and therefore simulating this SDE starting from samples drawn from π^{base} allows one to generate new samples from π at the last step of the dynamics.

This reverse stochastic dynamics crucially depends on the family of score functions $\{\nabla \log p_t\}_{t\in[0,T]}$, which are as intractable as the marginal densities themselves. The core idea of DMs is to approximate this entire family using a single neural network $\mathbf{s}^\theta : [0,T] \times \mathbb{R}^d \to \mathbb{R}^d$, parameterized by $\theta \in \Theta$, and to substitute this approximation into the dynamics (1.19). To embed this construction into a scalable machine learning framework, DMs exploit Tweedie's formula [Robbins, 1992], which expresses the score as a conditional expectation: for any $x \in \mathbb{R}^d$,

$$\nabla \log p_t(x) = \mathbb{E}\left[\nabla \log p_{t|0}(x \mid X_0) \mid I_t = x\right] \ . \tag{1.20}$$

Here, the expectation is taken with respect to the conditional distribution of X_0 given $I_t = x$, denoted by $p_{0|t}(\cdot \mid x)$. This distribution aims at predicting the terminal state X_0 of the reverse dynamics given an input (t, x), and is commonly referred to as the *posterior* distribution, as it admits the following density expression via Bayes' rule:

$$p_{0|t}(x_0 \mid x) \propto p_{t|0}(x \mid x_0)\, \pi(x_0) \ . \tag{1.21}$$

By definition of the interpolant (1.15), the score admits the equivalent expressions

$$\nabla \log p_t(x) = \frac{-x + \alpha_t \mathbb{E}\left[X_0 \mid I_t = x\right]}{\sigma_t^2} = \frac{\mathbb{E}\left[Z \mid I_t = x\right]}{\sigma_t} \ . \tag{1.22}$$

Hence, by forming suitable linear combinations of these expressions, learning $\nabla \log p_t(x)$ with $\mathbf{s}_t^\theta(x)$ amounts to matching $\mathbb{E}\left[s_t(X_0, Z) \mid I_t = x\right]$, where $s_t(x_0, z) = x_0$ (data-prediction) [Karras et al., 2022], $s_t(x_0, z) = z$ (noise-prediction) [Song et al., 2021b], or $s_t(x_0, z) = \alpha_t z - \sigma_t x_0$ (v-prediction) [Salimans and Ho, 2022]. Exploiting

the properties of conditional expectations, score learning can be formulated as the mean squared optimization problem

$$\arg\min_{\theta\in\Theta} \mathbb{E}\left[\left\|\mathbf{s}_t^\theta(I_t) - s_t(X_0, Z)\right\|^2\right] , \tag{1.23}$$

where the expectation is taken over $(X_0, Z) \sim \pi \otimes \pi^{\text{base}}$ and $t \sim \mathrm{U}[0, T]$. In practice, this objective is approximated empirically by sampling X_0 from the dataset $\mathcal{D}$, and stochastic gradient methods are used to carry out the optimization. Importantly, this training objective is computationally efficient, since it only requires independent sampling from Gaussian distributions and from the available dataset $\mathcal{D}$ to generate realizations of the interpolant I_t. Variants of this loss are commonly referred to as Denoising Score Matching (DSM) losses. A substantial body of recent work has focused on developing variance-stabilized and large-scale variants of these objectives, see for instance the preconditioning guidelines in Karras et al. [2022] and the fine-grained weighting schemes proposed in Karras et al. [2024].

Once training of the DSM objective is complete, the remaining step (3) consists in simulating the denoising dynamics. This is achieved by numerically solving an approximate version of the reverse SDE (1.19), in which the true score is replaced by $\mathbf{s}^\theta$, using suitable integration schemes. Given a time discretization grid of $[0, T]$, typically consisting of on the order of 100 to 1000 steps, one commonly resorts to first-order methods such as the Euler–Maruyama scheme, or to the more refined Exponential Integration (EI) scheme [Durmus and Moulines, 2015], which is closer to the original sampling procedure introduced in Ho et al. [2020] and theoretically guarantees improved integration accuracy. In the contribution of this thesis, we will systematically rely on the latter.

Close-up on Flow Matching models. FMs rely on an observation closely related to that underlying DMs, as they also exploit the existence of marginal-preserving dynamics, albeit in a deterministic setting. More precisely, Lipman et al. [2023], Albergo and Vanden-Eijnden [2023] show that the interpolant defined in (1.15) admits the same marginal distributions as the Markov process given by the solution of the following ODE,

$$\mathrm{d}X_t = v_t^{\mathrm{FM}}(X_t)\mathrm{d}t,\ X_0 \sim \pi, \quad \text{where} \quad v_t^{\mathrm{FM}}(x) = \mathbb{E}[\dot\alpha_t X_0 + \dot\sigma_t Z \mid I_t = x] \ . \tag{1.24}$$

In contrast to the DM framework, the forward transitions in this formulation are not tractable. The most commonly used noising schedule in this setting is the *Linear* interpolation $\alpha_t = 1 - t/T$ and $\sigma_t = t/T$. Although not originally coined as FM, a deterministic formulation of marginal-preserving dynamics derived from the interpolant was first explored in Song et al. [2021b], who introduced an alternative dynamics, referred to as the Probability Flow ODE, and defined as

$$\mathrm{d}X_t = v_t^{\mathrm{PF}}(X_t)\mathrm{d}t,\ X_0 \sim \pi, \quad \text{where} \quad v_t^{\mathrm{PF}}(x) = f(t)x - \frac{1}{2}g(t)^2\nabla \log p_t(x) \ . \tag{1.25}$$

Here, the coefficients f and g are defined in 1.18. While these expressions may appear to correspond to distinct dynamics with identical marginals, they in fact coincide, since $v^{\mathrm{PF}} = v^{\mathrm{FM}}$, as can be verified using the relations in (1.18) and (1.22). This direct connection between DMs and FMs makes it possible to reuse the same score-matching techniques introduced previously to learn the velocity field v, without modifying the training stage.

The key difference lies in the inference procedure (3): the denoising dynamics is now deterministic, and is obtained by simulating the reverse ODE on $[0, T]$, given by

$$\mathrm{d}Y_t = -v_{T-t}^{\mathrm{PF}}(Y_t)\mathrm{d}t,\ Y_0 \sim \pi^{\text{base}} \ . \tag{1.26}$$

Since this ODE cannot be solved analytically in general, one resorts to numerical integration methods, in close analogy with the DM case. The most commonly used approaches are first-order schemes, in particular the Euler method, which also admits an EI analogue (see Chapter 6 for related computations) that provides finer control over the integration error and corresponds to the fully deterministic variant of Denoising Diffusion Implicit Model (DDIM) [Song et al., 2021a].

While deterministic FM dynamics produce a single denoised output x_0 for a given input x_T, stochastic DM dynamics can, by construction, generate more diverse samples. Nevertheless, FMs are often preferred to DMs in practical applications, especially in image generation, as they empirically exhibit a lower sensitivity to reductions in computational budget at inference time (*i.e.*, a smaller number of discretization steps), as already observed in the seminal work of Song et al. [2021a].

Translation in (log-)density space. The equivalence between DM and FM formulations becomes particularly transparent when considering the evolution of the density functions p_t in time and space through a Partial Differential Equation (PDE). Exploiting the noising SDE (1.17), one can show that the family of marginal densities $(p_t)_{t\in[0,T]}$ satisfies the so-called Fokker-Planck Equation (FPE) [Risken, 1989], which uniquely characterizes the evolution of the marginals of a Markov process. More precisely, for any $(t,x) \in [0,T]\times\mathbb{R}^d$, the density p_t is governed by

$$\partial_t p_t(x) + \operatorname{div}\left(p_t(x) f(t) x\right) = \frac{g(t)^2}{2}\Delta p_t(x)\,. \tag{1.27}$$

Here, div denotes the divergence operator. Using the identity $\Delta p_t(x) = \operatorname{div}(p_t(x)\nabla \log p_t(x))$ together with the expression of v_t^{PF} given in (1.25), a direct manipulation of (1.27) yields the strictly equivalent PDE

$$\partial_t p_t(x) + \operatorname{div}\left(p_t(x) v_t^{\mathrm{PF}}(x)\right) = 0\,. \tag{1.28}$$

This equation is commonly referred to as the Continuity Equation (CE), and, as such, uniquely characterizes the evolution of the marginal densities associated with the ODE (1.24), or equivalently (1.25) [Evans, 2010]. Hence, this direct correspondence in density space between the FPE and the CE provides a natural bridge between DM and FM formulations, which originally operate at the level of generated particles.

In practice, it is often convenient to consider the evolution of the log-densities $(\log p_t)_{t\in[0,T]}$ instead. By straightforward manipulations of (1.27) and (1.28), one obtains the logarithmic counterparts of the FPE and the CE, respectively referred to as the log-FPE and log-CE. For any $(t,x)\in[0,T]\times\mathbb{R}^d$, they are given by

$$\text{log-FPE:}\quad \partial_t \log p_t(x) = \mathcal{F}^{\mathrm{FPE}}[\log p_t, f_t](x), \tag{1.29}$$

$$\text{log-CE:}\quad \partial_t \log p_t(x) = \mathcal{F}^{\mathrm{CE}}[\log p_t, v_t^{\mathrm{PF}}](x)\,, \tag{1.30}$$

where we denote $f_t : x \mapsto f(t)x$, and where the associated functionals $\mathcal{F}^{\mathrm{FPE}}$ and $\mathcal{F}^{\mathrm{CE}}$ are defined as

$$\mathcal{F}^{\mathrm{FPE}}[U_t, b_t](x) = -\operatorname{div} b_t(x) - b_t(x)^\top \nabla U_t(x) + \frac{g(t)^2}{2}\left(\|\nabla U_t(x)\|^2 + \Delta U_t(x)\right), \tag{1.31}$$

$$\mathcal{F}^{\mathrm{CE}}[U_t, v_t](x) = -\operatorname{div} v_t(x) - v_t(x)^\top \nabla U_t(x), \tag{1.32}$$

for any potential function $U : [0,T]\times\mathbb{R}^d \to \mathbb{R}$ and vector-valued functions $b : [0,T]\times\mathbb{R}^d \to \mathbb{R}^d$ and $v : [0,T]\times\mathbb{R}^d \to \mathbb{R}^d$.

Generalization to arbitrary data translation problems. In pure generative modeling, the objective is to generate new samples from the target distribution π; as a result, the choice of the noise distribution π^{base} is largely immaterial, and Gaussian noise is typically adopted for practical convenience. A natural question is whether the framework introduced above can be extended to construct transport maps between arbitrary distributions π_0 and π_T, for which only samples are available. These samples may be paired or unpaired, and are assumed to be i.i.d. realizations from an initial coupling $\pi_{0,T}^{\mathrm{init}} \in \Pi_{0,T}$, where $\Pi_{0,T}$ denotes the set of joint distributions on $\mathbb{R}^d\times\mathbb{R}^d$ with marginals π_0 and π_T. Ultimately, the goal is to generate new samples from π_0 conditioned on samples from π_T, and vice versa, that is, to sample from a coupling $\pi_{0,T} \in \Pi_{0,T}$ using a diffusion-based framework.

This objective can be achieved within the general framework of Stochastic Interpolant (SI) [Albergo et al., 2023], which follows steps analogous to (1)-(2)-(3). Specifically, SI considers a family of intermediate distributions $(p_t)_{t\in[0,T]}$ that interpolate between $p_0 = \pi_0$ and $p_T = \pi_T$, induced by the stochastic process

$$I_t = a_t X_0 + b_t X_T + c_t Z, \quad (X_0, X_T) \sim \pi_{0,T}^{\mathrm{init}}, \quad Z \sim \mathrm{N}(0, \mathrm{I}_d)\,, \tag{1.33}$$

where Z is independent of (X_0, X_T). Here, $a : [0,T]\to\mathbb{R}$, $b : [0,T]\to\mathbb{R}$, and $c : [0,T]\to\mathbb{R}$ are smooth interpolation coefficients, chosen such that a is decreasing with $(a_0, a_T) = (1,0)$, b is increasing with $(b_0, b_T) = (0,1)$, and $(c_0, c_T) = (0,0)$. A standard choice for the schedule is $a_t = 1 - t/T$, $b_t = t/T$, and $c_t = \sqrt{t(T-t)/T}$. Marginally, this interpolant corresponds to a standard Brownian bridge on $[0,T]$, conditioned on X_0 at $t=0$ and on X_T at $t=T$. In this particular case, SI approach is equivalently referred to as Bridge Matching (BM) in literature.

For this interpolant, Albergo et al. [2023] derive marginal-preserving SDE and ODE dynamics on $[0,T]$, starting from $X_0 \sim \pi_0$, given by

$$\text{SI-SDE:}\quad \mathrm{d}X_t = \left[v^{\mathrm{SI}}(t, X_t) + \frac{g(t)^2}{2}\nabla \log p_t(X_t)\right]\mathrm{d}t + g(t)\mathrm{d}B_t\,, \tag{1.34}$$

$$\text{SI-ODE:}\quad \mathrm{d}X_t = v^{\mathrm{SI}}(t, X_t)\mathrm{d}t\,, \tag{1.35}$$

where $(B_t)_{t\in[0,T]}$ denotes a Brownian motion, $g : [0,T] \to \mathbb{R}_+$ is a user-defined non-negative weighting function, and the velocity field v^{SI} and the score $\nabla \log p_t$ are expressed as conditional expectations:

$$v^{\mathrm{SI}}(t,x) = \mathbb{E}\Big[\dot{a}_t X_0 + \dot{b}_t X_T + \dot{c}_t Z \mid I_t = x\Big] \ , \qquad \nabla \log p_t(x) = \mathbb{E}\big[-c_t^{-1} Z \mid I_t = x\big] \ . \tag{1.36}$$

As in the DM/FM setting, backward dynamics starting from $X_T \sim \pi_T$ can be constructed via time reversal of SDE (1.34) (using score-based formula) and ODE (1.35), and score-matching techniques can be employed to learn v^{SI} and, if required, $\nabla \log p_t$, using data- or noise-prediction parameterizations.

A crucial aspect of the SI framework is that simulating either the stochastic dynamics (1.34) or the deterministic dynamics (1.35) results in sampling from a coupling $\pi^{\mathrm{SI}}_{0,T} \in \Pi_{0,T}$ that may differ substantially from the initial coupling $\pi^{\mathrm{init}}_{0,T}$. While this discrepancy may appear undesirable, it can in fact be leveraged to accelerate inference in diffusion-based models. For instance, in the noise-free linear case ($c_t = 0$), Liu [2022] show that iteratively retraining the SI model using interpolant couplings obtained by simulating the previously learned ODE leads to increasingly straight trajectories. In the limit, this procedure converges to the Optimal Transport (OT) plan for the quadratic cost between π_0 and π_T. From a theoretical perspective, this would require only a single inference step—corresponding to constant-velocity trajectories— to generate new samples in either direction. This idea, coined as *Rectified Flows*, has recently attracted significant attention [Liu et al., 2023, Wang et al., 2025a, Lee et al., 2024, Esser et al., 2024] as a means to enable few-step generation. When the interpolant follows a Brownian bridge, analogous techniques yield entropy-regularized OT plans between π_0 and π_T, as detailed in Section 1.5.2. These results will be key for understanding the contributions presented in Chapters 7 and 8.

Finally, in some applications it is desirable to preserve the initial coupling $\pi^{\mathrm{init}}_{0,T}$ throughout the SI dynamics, leading to so-called coupling-preserving approaches. As shown in [Bortoli et al., 2023, Albergo et al., 2024], this can be achieved by additionally conditioning the expectations in (1.36) on the initial state of the dynamics, that is, by fixing X_0 deterministically in the forward dynamics or X_T in the backward dynamics.

1.4.2 Diffusion models for sampling

Tempering vs diffusion density path. To sample from a target distribution π with a complex energy landscape (typically multi-modal), classical sampling approaches have motivated the use of the so-called *annealing* paradigm. As recalled in Section 1.3.5, this aims at decomposing the original sampling problem into a sequence of sampling sub-problems, given by $(K+1)$ intermediate distributions $\{p_k\}_{k=0}^K$ that gradually interpolate between the target $p_0 = \pi$ and an easy-to-sample distribution $p_K = \pi^{\mathrm{base}}$ (e.g., a Gaussian distribution). In practice, standard approaches include AIS, SMC, and RE, and the bridging distributions are built via the geometric interpolation defined in (1.8). By taking the limit $K \to \infty$, the tempering construction leads to a continuous-time density path $(p_t)_{t\in[0,T]}$ of the form

$$p_t(x) \propto \pi(x)^{\beta_t} \left(\pi^{\mathrm{base}}(x)\right)^{1-\beta_t} \ , \tag{1.37}$$

where the so-called *annealing schedule* $\beta : [0,T] \to [0,1]$ is continuous and decreasing, with $(\beta_0, \beta_T) = (1,0)$. In the following, we refer to this density path as the **tempering path**. In this setting, each intermediate density is explicitly defined up to a normalizing constant.

It is natural to compare the tempering path (1.37) with the interpolation density path defined in (1.16), induced by DMs, which also bridges π to π^{base}. We refer to the latter as the **diffusion path**. By introducing a suitable time discretization of $[0,T]$, denoted by $\{t_k\}_{k=0}^K$, one can define a sequence of $(K+1)$ intermediate distributions $\{p_k\}_{k=0}^K$ from the diffusion path by setting $p_k = p_{t_k}$, in direct analogy with classical annealed sampling problems. This naturally suggests the use of diffusion models within a sampling framework, despite their original motivation in generative modeling. A key distinction, however, is that the intermediate densities along the diffusion path are no longer tractable. As a consequence, standard annealed sampling methods cannot be applied directly, which has motivated the development of novel sampling algorithms, as discussed below.

With these two density paths in mind, a natural question arises: why consider the diffusion path rather than the tempering path for sampling when dealing with intractable densities? The answer lies in a fundamental structural difference between the two paths, particularly in the presence of multi-modal targets π. On the one hand, tempering paths are prone to *mass teleportation* (also referred to as mode switching in the literature), which corresponds to abrupt shifts of probability mass between modes along the interpolation [Máté and Fleuret, 2023]. In practice, such abrupt changes violate the assumed proximity between successive bridging distributions, making it difficult to ensure smooth transitions and often requiring careful tuning of the annealing schedule or

very large values of K, which may be prohibitive or infeasible in some applications. On the other hand, diffusion paths do not suffer from this issue, thanks to the smoothing effect of the Gaussian convolution induced by the noising process, which in principle yields smoother transitions between intermediate densities.

A second motivation for using diffusion paths lies in the possibility of exploiting pre-trained DMs to facilitate sampling through guidance mechanisms [Ho and Salimans, 2022]. This aspect plays a central role in Chapter 5. In contrast, tempering paths are agnostic to any external knowledge, such as prior samples, and rely solely on evaluations of intermediate densities. That said, we emphasize that substantial prior work on advanced annealed sampling methods based on tempering paths has focused on incorporating historical information about previously visited states, particularly in applications from biology and chemistry where multi-modality is pervasive; see, for instance, Stoltz et al. [2010] for a comprehensive overview. However, as these methods are often highly problem-specific, their detailed study lies beyond the scope of this thesis.

Overview of existing diffusion-based sampling methods. The use of diffusion model frameworks for sampling is a relatively recent line of research, with the first influential contributions appearing less than four years ago. Since then, a substantial body of work has emerged, including the methods presented in Chapters 4 to 6. To situate this thesis within this rapidly growing literature, we provide below a comprehensive survey of diffusion-based sampling methodologies developed in recent years, building on the theoretical foundations introduced in Section 1.4.1. Our goal is to highlight the main challenges addressed by the different classes of methods, their respective advantages and limitations, and the ways in which they complement or have inspired the approaches proposed in this thesis.

All of the methods reviewed here share a common annealing-based perspective: to generate new samples from a target distribution π, they aim to simulate a continuous-time *backward* dynamics—either stochastic (*i.e.*, SDE) or deterministic (*i.e.*, ODE)— defined on the time interval $[0, T]$, which transports π^{base} at time $t = T$ to π at time $t = 0$, in the spirit of DM/FM formulations. Another key commonality is that the drift governing this dynamics is intractable and must be estimated, either prior to or during the sampling procedure. This estimation problem constitutes the central challenge of all diffusion-based sampling methods, since, in contrast to generative modeling, one does not have direct access to samples from π, for which drift estimation is straightforward, as discussed in Section 1.4.1. In the literature, these approaches are commonly grouped under the umbrella term of **diffusion-based sampling methods**.

These methods can be broadly categorized into three groups, depending on the nature of the density path $(p_t)_{t\in[0,T]}$ induced by the transport dynamics:

- *Denoising samplers*: naturally inherited from the generative modeling perspective, these methods mimic the denoising SDE (1.19) or ODE (1.26), so that the target density path coincides with the diffusion path.
- *Escorted Langevin samplers*: these methods extend the DM/FM framework to construct SDEs or ODEs whose marginal densities follow the tempering path (1.37) along the transport.
- *General bridge samplers*: representing the most general extension of the DM/FM framework, these methods construct SDEs or ODEs that match a flexible, typically learnable, density path along the transport, subject only to the boundary constraints imposed by π and π^{base}. In this case, the resulting density path is, in general, neither a diffusion path nor a tempering path.

In the following, we carefully distinguish between these three methodological families.

We further classify the methods into two main classes. The first class comprises approaches that require learning an approximation of the intractable drift, typically using a neural network parameterized by $\theta \in \Theta$ (categories A, B, C below). In this case, the resulting dynamics are referred to as neural SDEs or ODEs. A key advantage of these methods is that they enable *amortized sampling*, that is, the ability to generate large numbers of samples efficiently, in parallel, in contrast to traditional sequential MCMC methods. We refer the reader to the empirical survey in He et al. [2025b] to complement the presentation made below. The second class consists of methods that do not rely on learning to estimate the drift (category D below), and instead fully exploit sampling subroutines based on techniques reviewed in Section 1.3. Although these methods are not amortized, they may nevertheless be computationally attractive overall, as they avoid a potentially expensive learning phase.

In what follows, we adopt a unified continuous-time framework to facilitate comparison between methods. For clarity of exposition, we deliberately omit the time discretization aspects required for practical simulation, as they do not introduce conceptual difficulties. Throughout the remainder of this section, we consider a target distribution of the form $\pi(x) = \gamma(x)/\mathcal{Z}$, where γ denotes an unnormalized density and $\mathcal{Z}$ the corresponding normalizing constant.

A. Annealed Variational Inference methods. In the absence of ground-truth samples from the target distribution, the earliest diffusion-based sampling methods naturally adapted the variational inference (VI) paradigm, originally designed to push a base distribution toward π through a direct parameterized mapping, to the annealing setting. Formally, these methods aim to learn a neural SDE with associated path measure $\mathbb{P}^\theta$ by solving the optimization problem

$$\arg\min_{\theta\in\Theta} \mathrm{D}\big(\mathbb{P}^\theta \mid (\mathbb{P}^\star)^R\big) \ , \tag{1.38}$$

where D and R denote, respectively, a precribed divergence and the time-reversal operator acting on the space of path measures, and where $\mathbb{P}^\star$ is the path measure associated with a "forward" stochastic dynamics transporting π to π^{base}. Interestingly, this formulation can be recast as a stochastic optimal control problem [Tzen and Raginsky, 2019]. At optimality in (1.38), the forward and backward dynamics are guaranteed to be time reversals of one another, thereby ensuring that $\mathbb{P}^\theta_T = \pi$, which allows one to generate samples from π using the optimized neural SDE. In contrast to score-matching objectives, this variational formulation is not simulation-free, as it requires repeated simulations of the full neural SDE at each evaluation of the loss.

Within this variational framework, *denoising samplers* typically choose the noising SDE (1.17) as the forward dynamics, with various possible noising schedules. Most commonly, the divergence D is taken to be the Kullback–Leibler divergence [Zhang and Chen, 2022, Berner et al., 2023, Vargas et al., 2023b,a] or the Log–Variance divergence [Richter et al., 2023], which is computationally more lightweight in practice. Recent work has established strong theoretical foundations for these approaches [Reu et al., 2024, Berner et al., 2026], although empirical performance may still suffer from classical VI shortcomings, such as mode collapse. Over time, this original formulation has been extended in several directions. For instance, Blessing et al. [2025b] propose to automatically tune the base distribution π^{base} during variational training to improve transport efficiency, Blessing et al. [2025a] introduce an underdamped variant expected to smooth the backward dynamics and improve sampling efficiency, and Heng et al. [2024] and Tamogashev and Malkin [2025] concurrently investigate the possibility of recovering optimal transport plans by embedding the neural SDE within a sequential training procedure. While not directly following this formulation, He et al. [2025a] and Wang et al. [2025b] explore the possibility of bypassing neural SDEs altogether by relying on a neural generator to model p^θ, in the spirit of normalizing flows, within a variational framework. In this case, the generator is trained by minimizing a sum of marginal KL divergences between annealed variants of π and p^θ, obtained through Gaussian convolutions at increasing scales, similarly to the DM framework, at the cost of increased computational complexity in practice.

In parallel, following the seminal works Geffner and Domke [2021], Doucet et al. [2022], Geffner and Domke [2023], *escorted Langevin samplers* aim to construct dynamics that *escort* (*i.e.*, control) the unadjusted annealed Langevin dynamics so that their marginals exactly follow the tempering path (1.37). Prominently represented by Vargas et al. [2024], these methods reinterpret the notion of guidance by parameterizing the drift of both forward and backward SDE as a controlled version of the annealed Langevin drift, thereby leveraging the tractable tempering scores. The resulting objective jointly optimizes both forward and backward dynamics so as to match the tempering path marginally, which amounts to learning two neural SDEs. To alleviate mass teleportation effects during generation, Chen et al. [2025] propose an extension based on SMC resampling, allowing the VI objective to be evaluated over sub path-measures and thereby reducing its variance.

Finally, the extension of this variational framework to *general bridge samplers* is considered in [Richter et al., 2023], who propose to optimize both forward and backward dynamics in the objective (1.38), once again involving two neural networks as in Vargas et al. [2024], but now with fully flexible parameterizations.

(**Q3**) In the particularly important case where π is multi-modal, existing annealed VI methods have so far exhibited limited sampling efficiency, largely due to recurring mode collapse. Suppose that, in addition to access to the unnormalized density of such a target, one also has access to auxiliary information, such as samples concentrated in high-density regions of π but with incorrect mode proportions, a situation frequently encountered in practical sampling applications. In this context, can one design a diffusion-based VI method that effectively incorporates this prior information through guidance mechanisms, in order to mitigate mode collapse and mode misestimation, while maintaining practical computational efficiency? To address this question, we propose an end-to-end methodology [2] in Chapter 5.

B. PINN-related methods. A second strategy to circumvent the lack of ground-truth samples from π within the DM/FM framework consists in reasoning directly in log-density space, rather than at the level of generated

[2] Note that Kim et al. [2025c] address this question using a methodology similar to ours, but published their results several months after our work and did not credit us despite prior contact with the authors.

particles. In this setting, the objective is to learn a neural ODE or SDE by enforcing that the associated density path $(p_t)_{t\in[0,T]}$ follows a prescribed dynamics, governed either by the log-FPE (1.29) (for stochastic dynamics) or by the log-CE (1.30) (for deterministic dynamics), which fully characterize the corresponding particle evolutions. Depending on the nature of the density path, three approaches can be distinguished, leading to three main classes of Physics-Informed Neural Network (PINN) losses:

- *Denoising samplers*: for those samplers, we recall that the target density path coincides with the diffusion path. Since the family $(\log p_t)_{t\in[0,T]}$ is intractable, the idea is to approximate it using a real-valued neural network $\mathbf{U}^\theta : [0,T]\times\mathbb{R}^d \to \mathbb{R}$[3]. For instance, when targeting stochastic dynamics as proposed in Shi et al. [2024], this leads to the optimization problem

$$\text{log-FPE-PINN (diffusion):} \quad \operatorname*{arg\,min}_{\theta\in\Theta} \mathbb{E}\Big[\big\|\partial_t \mathbf{U}_t^\theta(X_t) - \mathcal{F}^{\mathrm{FPE}}[\mathbf{U}_t^\theta, f_t](X_t)\big\|^2\Big] \,, \tag{1.39}$$

  where the expectation is taken over $t \sim \mathrm{U}[0,T]$ and over so-called *collocation points* $(X_t)_{t\in[0,T]}$ sampled in $\mathbb{R}^d$. Note that, in this setting, the corresponding log-CE-PINN objective would be strictly equivalent.

- *Escorted Langevin samplers*: here, the density path is prescribed as the tempering path, for which each marginal density is explicitly defined up to a normalizing constant, denoted by $\tilde{p}_t$. The time-dependent normalizing constant is learned using a real-valued neural network $\mathbf{Z}^\theta : [0,T] \to \mathbb{R}$. The main approach in this category relies on the deterministic log-CE formulation, as proposed in Albergo and Vanden-Eijnden [2025], where the velocity field of the generative dynamics is learned via a neural network $\mathbf{v}^\theta : [0,T]\times\mathbb{R}^d \to \mathbb{R}^d$ by solving

$$\text{log-CE-PINN (tempering):} \quad \operatorname*{arg\,min}_{\theta\in\Theta} \mathbb{E}\Big[\big\|\partial_t \log\tilde{p}_t(X_t) - \partial_t \mathbf{Z}_t^\theta - \mathcal{F}^{\mathrm{CE}}[\log\tilde{p}_t, \mathbf{v}_t^\theta](X_t)\big\|^2\Big] \,. \tag{1.40}$$

- *General bridge samplers*: generalizing the two previous approaches, this setting aims to learn both the density path, represented through the normalized log-density $x \mapsto \mathbf{U}_t^\theta(x) - \mathbf{Z}_t^\theta$, and the driving velocity field $\mathbf{v}^\theta$ of the neural ODE [Máté and Fleuret, 2023, Tian et al., 2024, Chemseddine et al., 2025]. This significantly increases the complexity of the problem. To ensure validity of the density path, boundary constraints are imposed, namely $\mathbf{U}_0^\theta = \log\gamma$ and $\mathbf{U}_T^\theta = \log\pi^{\mathrm{base}}$, and the optimization problem writes

$$\text{log-CE-PINN (general):} \quad \operatorname*{arg\,min}_{\theta\in\Theta} \mathbb{E}\Big[\big\|\partial_t \mathbf{U}_t^\theta(X_t) - \partial_t \mathbf{Z}_t^\theta - \mathcal{F}^{\mathrm{CE}}[\mathbf{U}_t^\theta, \mathbf{v}_t^\theta](X_t)\big\|^2\Big] \,. \tag{1.41}$$

We refer to Sun et al. [2024], Fan et al. [2024] for comprehensive and unified overviews of these approaches, which also include extensions to optimal transport.

A common limitation of all PINN-based methods is the need to compute divergences of neural-network-defined vector fields. Exact computation quickly becomes prohibitively expensive beyond low-dimensional settings, severely limiting practical applicability. Recent work Plainer et al. [2025] proposes to mitigate this issue using approximation techniques based on the Hutchinson trace estimator, at the cost of introducing non-negligible variance. In addition, to avoid expensive backpropagation through higher-order neural network derivatives during training, these approaches are often combined with stop-gradient techniques; see Chapter 6 for concrete applications.

A second major drawback of this paradigm lies in the choice of collocation points. Ideally, these points should be distributed according to the support of the target density path, so as to enforce the log-density PDEs throughout the relevant regions of the space. However, since the shape of the desired path is unknown a priori, one typically resorts to uniform sampling over large compact domains enclosing the supports of both π and π^{base}, which can be highly inefficient. The next class of diffusion-based samplers addresses this limitation by exploiting target-informed samples within diffusion-based training objectives.

C. Adaptive learning methods. To mitigate the curse of dimensionality affecting VI and PINN-based approaches, and to accelerate training procedures, a class of hybrid diffusion-based sampling methods has recently emerged. These methods, which are most naturally associated with the diffusion density path, combine variational and sample-based training objectives within an *adaptive* paradigm. Their defining feature is an

[3]Most practitioners inform the neural network of the target density by enforcing $\mathbf{U}_0^\theta = \log\gamma$. In this case, $\mathbf{U}_t^\theta$ effectively approximates the unnormalized log-density $\log\tilde{p}_t = \log p_t + \log\mathcal{Z}$. Although the learning problem is slightly modified, the log-FPE still appplies since $\nabla\log p_t = \nabla\log\tilde{p}_t$ and $\partial_t \log p_t = \partial_t\log\tilde{p}_t$, and generation remains unchanged.

iterative optimization scheme for the parameter θ, which alternates between two stages: (1) simulating the neural ODE or SDE parameterized by θ to obtain approximate samples from π, and (2) minimizing a target-informed training objective with respect to θ by leveraging the samples generated in stage (1), thereby producing an updated parameter value. The underlying rationale is heuristic but intuitive, and can be interpreted as a virtuous cycle: samples obtained in stage (1) that are closer to π reduce bias in stage (2), which in turn improves the quality of the samples generated in subsequent iterations. A key advantage of adaptive learning methods over the previously discussed approaches is that stage (2) can directly exploit the generative modeling backbone of DMs and FMs, thereby benefiting from large-scale and computationally efficient training procedures that have already been extensively studied in the literature.

For adaptive methods to be effective, it is crucial to inject information about the target distribution π during the iterative procedure. A natural strategy is to do so at stage (2) by designing target-informed, sample-based objectives, which were not originally considered in standard generative modeling. A prominent line of work in this direction exploits a target analogue of Tweedie's formula,

$$\nabla \log p_t(x) = \mathbb{E}\left[\alpha_t^{-1} \nabla \log \gamma(X_0) \mid I_t = x\right] , \tag{1.42}$$

where the expectation is still taken with respect to the posterior distribution $p_{0|t}(\cdot \mid x)$, using the notation introduced in (1.15). For instance, the seminal work Akhound-Sadegh et al. [2024] proposes, for stage (2), a score-matching objective in which the regression target (based on the conditional score $\nabla \log p_{t|0}$ in theory, see (1.23)) is replaced by a Monte Carlo estimate of (1.42). This estimate is obtained by applying importance sampling to the posterior $p_{0|t}(\cdot \mid x)$, using a Gaussian proposal with unnormalized density $x_0 \mapsto p_{t|0}(x \mid x_0)$. Building on this idea, OuYang et al. [2024], Woo and Ahn [2024] employ the same IS procedure, this time to estimate the log-density $\log p_t(x)$ via the integral in (1.16), from which score estimates are obtained by differentiation.

However, this importance-sampling-based approach suffers from an intrinsic statistical limitation. When t is close to T, the posterior $p_{0|t}(\cdot \mid x)$ becomes nearly independent of x and closely resembles the target distribution π (a phenomenon clearly identified in Chapter 4), while the Gaussian proposal distribution has very broad support. As a consequence, when π is highly multi-modal, the importance sampling estimator becomes ineffective due to severe support mismatch; see Section 1.3.2. This phenomenon hence degrades the accuracy of the estimated drift at the beginning of the backward dynamics.

In principle, the identity (1.42) can also be exploited without relying on importance sampling. Indeed, if one has access to ground-truth samples $X_0 \sim \pi$, the DM score function can be learned by a neural network $\mathbf{s}^\theta : [0,T] \times \mathbb{R}^d \to \mathbb{R}^d$ by minimizing the so-called *Target Score Matching* (TSM) objective [Bortoli et al., 2024a], given by

$$\arg\min_{\theta \in \Theta} \mathbb{E}\left[\left\|\mathbf{s}_t^\theta(I_t) - \alpha_t^{-1} \nabla \log \gamma(X_0)\right\|^2\right] . \tag{1.43}$$

This observation motivated earlier works Phillips et al. [2024], Zhang et al. [2026], and more recently Havens et al. [2025][4], followed by the optimal transport extension in Liu et al. [2025a], to apply the objective (1.43) using samples X_0 generated in stage (1). While this yields a tractable and scalable training objective that is straightforward to implement, there is no guarantee that stage (2) effectively improves the accuracy of the driving drift, due to the bias of the training samples obtained in stage (1) with respect to π. This issue is particularly acute during early iterations, when the neural ODE or SDE remains largely uninformed about the target, and can ultimately undermine the expected virtuous cycle.

To address this limitation, several works have investigated the use of target information directly at stage (1), with the goal of generating samples that more closely resemble π through additional sampling procedures, in a manner analogous to Boltzmann generators for normalizing flows [Noé et al., 2019]. These so-called *adjusted* approaches aim to ensure, in a statistical sense, that the generated samples follow exactly the diffusion density path. Prominent examples include Phillips et al. [2024], which relies on SMC, and Zhang et al. [2026], which leverages RE. Although these methods are theoretically well grounded, they require, in practice, the estimation of intermediate diffusion log-densities. To this end, Phillips et al. [2024], Zhang et al. [2026] learn these log-densities using a real-valued neural network $\mathbf{U}^\theta : [0,T] \times \mathbb{R}^d \to \mathbb{R}$, by fitting its gradient $\mathbf{s}_t^\theta = \nabla \mathbf{U}_t^\theta$ within the TSM objective. It is nevertheless important to note that the majority of adaptive methods remain *unadjusted* (see, e.g., Akhound-Sadegh et al. [2024], OuYang et al. [2024], Woo and Ahn [2024], Havens et al. [2025], Liu

[4] Although [Havens et al., 2025] derive their training objective from stochastic optimal control arguments, it is in fact equivalent to the TSM objective up to a linear reparameterization of the score, as acknowledged by the authors.

et al. [2025a]), meaning that no explicit calibration with respect to π is performed during stage (1). While computationally lightweight—since stage (1) only involves sampling from the neural ODE or SDE— this can introduce significant bias in the generated samples relative to π, thereby weakening the intended virtuous learning cycle across iterations. This context naturally motivates the following question.

(**Q4**) Assume we are given a pretrained DM intended to generate samples from a multi-modal target distribution π, specified only through its unnormalized density. A central practical challenge is to guarantee the generation of unbiased samples from π when using such a pretrained model. For normalizing flows, this issue is commonly addressed via importance sampling enabled by tractable likelihoods, leading to so-called Boltzmann generators [Noé et al., 2019]. Assuming access to intermediate DM densities, or suitable estimators thereof, can one develop an analogous calibration framework for DMs based on standard annealing methodologies (AIS, SMC, RE), that fully exploits the sequential structure of DMs and remains robust in challenging multi-modal settings? In practice, how does such a framework interact with existing log-density estimation techniques for DMs? Generalizing prior work Phillips et al. [2024], Zhang et al. [2026], this question is addressed in Chapter 6.

Sequential learning methods: a promising alternative? Very recently, a new class of diffusion-based sampling methods [Rissanen et al., 2025, Akhound-Sadegh et al., 2025] has emerged as an alternative to the learning-based approaches discussed above. In this framework, the objective is to sample from π by following a discrete tempering path $\{p_k\}_{k=0}^K$ defined by (1.8), starting from $k = K$ with decreasing k. However, instead of learning a *single* dynamics whose marginals coincide with this density path, the core idea is to learn *multiple* neural dynamics, namely one dynamics for each predefined tempering level $k \in \{0, \dots, K\}$. For a given level k, the k-th target dynamics is defined as a denoising SDE that bridges π^{base} to p_k. Learning is performed sequentially, starting from $k = K$ and progressively decreasing k. In particular, resampling procedures between the dynamics of consecutive levels can be exploited to generate training samples, such as SMC in Akhound-Sadegh et al. [2025] or RE in Rissanen et al. [2025].

Compared to directly applying traditional annealed sampling methods along the tempering path, this hybrid strategy might benefit from better-conditioned transitions between successive levels, thereby potentially alleviating mass teleportation effects. To date, however, *this advantage has not yet been formally established.* Furthermore, as in the case of *adjusted adaptive* methods, the effectiveness of this approach critically depends on accurate diffusion-based log-density estimation, which remains non-trivial within the DM framework.

D. Methods without learning. In view of the mode collapse issues affecting variational objectives, the inefficiency of PINN-based approaches in high-dimensional settings, the statistical limitations of adaptive sample-based objectives, and, more broadly, the substantial computational cost associated with these learning-based methods, a natural question arises: can one simulate the target "backward" dynamics by relying solely on Monte Carlo estimators of the underlying drift?

Following early works Huang et al. [2021], Bernton et al. [2019a], this research direction has primarily been explored within the diffusion path framework in recent years. In this setting, drift estimation is performed on-the-fly, *i.e.*, during the execution of the sampling procedure itself. At a high level, this amounts to constructing Monte Carlo estimators of the score function, either using the *standard* formulation (1.20) or the *target-informed* formulation (1.42), both of which express the score as a conditional expectation with respect to posterior distributions $(p_{0|t})_{t\in(0,T]}$. Within this general paradigm, diffusion-based sampling methods mainly differ in the Monte Carlo techniques employed to approximate these conditional expectations.

For instance, McDonald and Barron [2022] relies on importance sampling, He et al. [2024b] on rejection sampling, and Huang et al. [2024a] on Langevin-based algorithms. More advanced schemes have also been proposed. For example, Huang et al. [2024b] build upon Huang et al. [2024a] to introduce recursive Monte Carlo estimators that are expected to become progressively easier to sample from, at the expense of increased memory requirements. In a different direction, Wu et al. [2025a] adapt the SMC-based calibration scheme of Phillips et al. [2024], combined with additional density estimation performed using Monte Carlo techniques similar to those in OuYang et al. [2024], thereby theoretically reducing bias with respect to the target distribution. Finally, Encinar et al. [2025] propose embedding the denoising dynamics within a slow–fast SDE system, in which posterior sampling is carried out internally as part of the dynamics itself.

Complementary to this line of work, other approaches have investigated alternative sampling schemes that depart from the explicit estimation of denoising diffusion dynamics. For instance, Chen et al. [2024] and Huang et al. [2024c] independently propose Gibbs-like sampling procedures that operate on joint distributions

obtained by Gaussian convolution of π at a fixed noise scale, alternating between noising steps, which are easy to sample from, and denoising posterior steps, which remain challenging. In another direction, Saremi et al. [2024] introduce a non-Markovian analogue of diffusion models, enabling sampling from π through progressive denoising steps based on importance sampling.

Overall, all these approaches aim, in one way or another, to apply Monte Carlo techniques to sample from denoising posterior distributions rather than directly sampling from the target distribution itself. However, as discussed in paragraph C, these posterior distributions may exhibit pathological behavior for Monte Carlo methods when the target distribution is highly multi-modal. Within this context, we may naturally ask the following questions.

(**Q5**) In the particularly important case where π is multi-modal and only accessible through its unnormalized density, can one design a practical and efficient Monte Carlo sampler, built upon a denoising-like diffusion dynamics, that avoids the statistical flaws introduced by posterior sampling sub-procedures? Moreover, can the effectiveness of such an approach be characterized theoretically for a broad class of multi-modal target distributions? These questions are addressed in Chapter 4.

1.5 Optimal Transport

This section provides a brief introduction to the task of Optimal Transport (OT), which will serve as useful context for the elements introduced in Part III. In the same spirit as Section 1.3, this presentation is intentionally non-technical and adopts a global perspective, with the sole purpose of building intuition around the underlying mathematical concepts. We refer the reader to Peyré et al. [2019] for a more in-depth introduction, as well as for computational and theoretical details.

General remark. In its most general form, OT is a powerful framework for comparing two probability distributions ν and μ, defined on arbitrary measurable spaces $(\mathsf{X}, \mathcal{X})$ and $(\mathsf{Y}, \mathcal{Y})$, respectively. It relies on a prescribed cost function $c : \mathsf{X} \times \mathsf{Y} \to \mathbb{R}_+$, where $c(x, y)$ quantifies the cost of transporting x to y (and conversely from y to x when the cost is symmetric, as is typically assumed). In the context of generative modeling, the goal of OT is to exhibit an "optimal transport map" $\mathrm{T} : \mathsf{X} \to \mathsf{Y}$ that transports μ to ν in the least costly manner, that is by minimizing the cost induced by the displacement. This is the perspective adopted throughout this introduction. The construction of such a map T can be approached through two OT formulations, which we present sequentially. Section 1.5.1 first describes the static formulation of OT, which is the most classical one, while Section 1.5.2 introduces the dynamical formulation, which has gained significant attention due to its connections with diffusion models, a link that we specifically emphasize here. Unless specified otherwise, we consider $\mathsf{X} = \mathsf{Y} = \mathbb{R}^d$, with two distributions μ and ν supported on $\mathbb{R}^d$. For the theoretical discussion, we assume that both admit densities with respect to the Lebesgue measure. For the practical counterpart, focused on methodologies for solving OT problems, we systematically assume that these probability measures are described only through available samples, thereby connecting OT to the broader framework of generative modeling. Although the presentation applies to general cost functions, we place particular emphasis on the Euclidean *quadratic cost* $c(x, y) = \|x - y\|^2 / 2$, associated with linear displacement in $\mathbb{R}^d$, for which several simplifications and remarkable results arise when it exists.

Notation. In the following, we denote by $\Pi(\mu, \nu)$ the set of couplings between μ and ν, that is the set of joint distributions $\pi \in \mathscr{P}(\mathbb{R}^d \times \mathbb{R}^d)$ such that $\int_{\mathbb{R}^d} \pi(\cdot, \mathrm{d}y) = \mu$ and $\int_{\mathbb{R}^d} \pi(\mathrm{d}x, \cdot) = \nu$, meaning that the marginals are μ and ν, respectively. For probability distributions $\mu, \nu \in \mathscr{P}(\mathbb{R}^d)$ that are absolutely continuous with respect to the Lebesgue measure, we denote by $\mathrm{KL}(\mu|\nu)$ the Kullback-Leibler divergence, defined as $\mathrm{KL}(\mu|\nu) = \int_{\mathbb{R}^d} \log\left(\frac{\mu(x)}{\nu(x)}\right) \mu(\mathrm{d}x)$. We also denote by $\mathrm{H}(\mu)$ the entropy of μ, defined by $\mathrm{H}(\mu) = -\int_{\mathbb{R}^d} \log(\mu(x))\mu(\mathrm{d}x)$.

1.5.1 Static formulation

Unregularized setting. The OT framework is an old computational problem, whose first formulation dates back more than two centuries [Monge, 1781]. This original formulation, known as the Monge problem, is defined as

$$\underset{\substack{\mathrm{T}:\mathbb{R}^d \to \mathbb{R}^d \\ \mathrm{T}_{\#}\mu = \nu}}{\arg\min} \int_{\mathbb{R}^d} c(x, \mathrm{T}(x))\mu(\mathrm{d}x) \ . \tag{1.44}$$

When a solution $\mathrm{T}^\star$ exists, it is referred to as the Monge map or OT map. This formulation is intuitive, as it aims to transport μ to ν while minimizing the average displacement cost induced by c. It is important to note that OT was originally conceived as a deterministic transport problem, meaning that for almost every $x \sim \mu$, there exists a unique $y \sim \nu$ realizing the minimal transport, given by $y = \mathrm{T}^\star(x)$. However, the Monge problem is ill posed in general, as such a map $\mathrm{T}^\star$ may fail to exist. This limitation motivated the introduction of a stochastic relaxation of the Monge problem, known as the *Kantorovitch relaxation* [Kantorovitch, 1958], which has since become the standard OT formulation.

Rather than seeking a deterministic map T pushing μ onto ν, the Kantorovitch problem aims to identify a coupling $\pi \in \Pi(\mu, \nu)$. This induces a fundamental change in the transport mechanism, as the transport becomes stochastic. Informally, for almost every $x \sim \mu$, mass is transported toward ν according to the conditional distribution $\pi(\cdot|x) = \frac{\pi(x,\cdot)}{\mu(x)}$, allowing in principle infinitely many possible destinations. In this setting, one may still define a deterministic map through the barycentric projection $x \mapsto \int_{\mathbb{R}^d} y\pi(\mathrm{d}y|x)$, which however does not constitute a valid push-forward from μ to ν. The Kantorovitch problem admits two equivalent formulations:

- *the primal formulation*, given by

$$\operatorname*{arg\,min}_{\pi \in \Pi(\mu,\nu)} \int_{\mathbb{R}^d \times \mathbb{R}^d} c(x, y)\pi(\mathrm{d}x, \mathrm{d}y) \;, \tag{1.45}$$

- *the dual formulation*, which recasts the OT problem as a maximization over continuous potentials $f : \mathbb{R}^d \to \mathbb{R}$ and $g : \mathbb{R}^d \to \mathbb{R}$, known as *Kantorovitch potentials*,

$$\operatorname*{arg\,max}_{f,g \in \mathcal{F}} \int_{\mathbb{R}^d} f(x)\mu(\mathrm{d}x) + \int_{\mathbb{R}^d} g(y)\nu(\mathrm{d}y) \;, \tag{1.46}$$

  where $\mathcal{F} = \{f, g : f(x) + g(y) \leq c(x, y) \;, \forall (x, y) \in \mathbb{R}^d \times \mathbb{R}^d\}$. Although less intuitive, the dual formulation is central both for theoretical analysis and practical computation in OT.

While the Monge and Kantorovitch formulations differ in nature, they are known to be equivalent when the cost function is quadratic, which is the primary case of interest considered here. In this setting, Brenier's theorem [Brenier, 1987] guarantees that (1.44) and (1.45) admit unique solutions $\mathrm{T}^\star$ and $\pi^\star$, respectively, satisfying $\pi^\star = (\mathrm{Id}, \mathrm{T}^\star)_\# \mu$. In other words, the optimal coupling is deterministic, with $\pi^\star(\cdot|x)$ equal to a Dirac mass at $\mathrm{T}^\star(x)$, so that the barycentric projection coincides with the Monge map. Moreover, the optimal value of (1.45) defines the square of a distance on the space of probability measures between μ and ν, known as the Wasserstein-2 distance and denoted by $W_2(\mu, \nu)$. Brenier's theorem further establishes the equivalence between the primal and dual Kantorovitch formulations through the decomposition $\mathrm{T}^\star = \nabla\phi^\star$, where $\phi^\star : \mathbb{R}^d \to \mathbb{R}$ is a convex function, called the Brenier potential, defined by $\phi^\star(x) = \|x\|^2/2 - f^\star(x)$, with $f^\star$ a solution to (1.46).

Building on the Kantorovitch formulation, the OT framework naturally extends to the multi-marginal setting, where one is given a collection of marginal distributions $\{\mu_i\}_{i=0}^\ell$, with $\ell > 1$, together with a generalized displacement cost function $\bar{c} : (\mathbb{R}^d)^{\ell+1} \to \mathbb{R}_+$. In this case, the primal formulation becomes

$$\operatorname*{arg\,min}_{\bar{\pi} \in \Pi(\mu_{0:\ell})} \int_{(\mathbb{R}^d)^{\ell+1}} c(x_0, x_1, \ldots, x_\ell)\bar{\pi}(\mathrm{d}x_0, \mathrm{d}x_1, \ldots, \mathrm{d}x_\ell) \;, \tag{1.47}$$

where $\Pi(\mu_{0:\ell})$ denotes the set of joint distributions $\bar{\pi} \in \mathscr{P}((\mathbb{R}^d)^{\ell+1})$ whose marginals are $\mu_0, \mu_1, \ldots, \mu_\ell$. In contrast with the bi-marginal case, the multi-marginal OT problem is less understood and remains relatively unexplored. We refer the reader to Pass [2015] for a survey of theoretical results and applications. A case of particular interest, which we study in Chapter 7, arises when the cost takes the form $\bar{c}(x_0, x_1, \ldots, x_\ell) = \inf_{y \in \mathbb{R}^d} \sum_{i=0}^\ell c(x_i, y)$, corresponding to minimizing a two-state cost between each $x_i \sim \mu_i$ and an implicit latent state y. When c is quadratic, y coincides with the barycenter of $\{x_i\}_{i=0}^\ell$, yielding the explicit expression

$$c(x_0, x_1, \ldots, x_\ell) = \sum_{i=0}^{\ell} \frac{\|x_i - \bar{x}\|^2}{2} \text{ where } \bar{x} = \frac{1}{\ell+1} \sum_{i=0}^{\ell} x_i \;.$$

In this setting, the associated multi-marginal OT problem is equivalent to the *Wasserstein barycenter problem* [Agueh and Carlier, 2011], see for instance [Carlier and Ekeland, 2010, Proposition 3], which is defined as

$$\operatorname*{arg\,min}_{\nu \in \mathscr{P}(\mathbb{R}^d)} \sum_{i=0}^{\ell+1} W_2^2(\mu_i, \nu) \;. \tag{1.48}$$

Thus, solving the multi-marginal OT problem with this barycentric cost amounts to identifying a probability measure $\nu \in \mathscr{P}(\mathbb{R}^d)$ that can be interpreted as the Wasserstein-2 barycenter of the given marginals. This formulation is particularly appealing in machine learning, where it naturally extends clustering methods when the marginals are represented by samples.

Entropy-regularized setting. Despite its conceptual appeal, the Kantorovitch formulation is challenging to solve in practice. Although the dual formulation can be cast as a finite dimensional linear program when N samples from the marginals are available, for which discrete optimization methods exist [Burkard et al., 2012], such approaches are computationally prohibitive, with complexity of order $O(N^3)$. Moreover, they may suffer from numerical instability, as optimal transport plans typically have sparse support. These difficulties motivated the introduction of an entropy-regularized variant of the Kantorovitch problem, known as the Entropy-regularized Optimal Transport (EOT) problem. This formulation augments the original objective with an entropy penalty, which encourages transport plans with more diffuse support. The strength of this regularization is controlled by a parameter $\varepsilon > 0$, with the limiting cases corresponding to no regularization, as $\varepsilon \to 0$, which recovers the original problem, and dominant regularization, as $\varepsilon \to \infty$, which amounts to maximizing entropy alone. As in the unregularized case, the entropy-regularized problem admits two equivalent formulations:

- *the primal formulation*,

$$\operatorname*{arg\,min}_{\pi \in \Pi(\mu,\nu)} \int_{\mathbb{R}^d \times \mathbb{R}^d} c(x,y)\pi(\mathrm{d}x,\mathrm{d}y) - \varepsilon \mathrm{H}(\pi) \;, \tag{1.49}$$

  which is now strongly convex in π, unlike the unregularized problem (1.45), and therefore admits a unique solution for general costs,

- *the dual formulation*, which optimizes unconstrained potentials $f : \mathbb{R}^d \to \mathbb{R}$ and $g : \mathbb{R}^d \to \mathbb{R}$, referred to as *Schrödinger potentials*,

$$\operatorname*{arg\,max}_{f,g} \int_{\mathbb{R}^d} f(x)\mu(\mathrm{d}x) + \int_{\mathbb{R}^d} g(y)\nu(\mathrm{d}y) - \varepsilon \int_{\mathbb{R}^d \times \mathbb{R}^d} \exp\left(\frac{f(x)+g(y)-c(x,y)}{\varepsilon}\right) \mu(\mathrm{d}x)\nu(\mathrm{d}y) \;, \tag{1.50}$$

  where the inequality constraints present in the unregularized dual formulation are now directly absorbed into the objective through the entropy term.

As shown in Nutz [2021], the primal and dual solutions in the entropy-regularized setting are linked by

$$\pi^\star(\mathrm{d}x,\mathrm{d}y) = \exp\left(\frac{f^\star(x)+g^\star(y)-c(x,y)}{\varepsilon}\right) \mu(\mathrm{d}x)\nu(\mathrm{d}y) \;, \tag{1.51}$$

where $\pi^\star$ denotes the unique solution of (1.49), and $(f^\star, g^\star)$ is the unique solution of (1.50). This explicit relation forms the foundation of the *Sinkhorn algorithm*, a widely used procedure for solving entropy-regularized OT problems, which we describe next.

Similarly, the multi-marginal OT problem (1.47) admits an entropy-regularized version, whose primal formulation is naturally defined as

$$\operatorname*{arg\,min}_{\bar\pi \in \Pi(\mu_{1:\ell})} \int_{(\mathbb{R}^d)^{\ell+1}} c(x_0,x_1,\dots,x_\ell)\bar\pi(\mathrm{d}x_0,\mathrm{d}x_1,\dots,\mathrm{d}x_\ell) - \varepsilon \mathrm{H}(\bar\pi) \;. \tag{1.52}$$

Finally, the Wasserstein barycenter problem also admits a standard entropy-regularized counterpart, see for instance [Benamou et al., 2015, Section 3.2], defined as

$$\operatorname*{arg\,min}_{\nu \in \mathscr{P}(\mathbb{R}^d)} \sum_{i=0}^{\ell} W_{2,\varepsilon}^2(\mu_i,\nu) \;, \tag{1.53}$$

where $W_{2,\varepsilon}^2$ denotes the entropy-regularized squared Wasserstein-2 distance, given by the optimal value of (1.49) with quadratic cost.

IPF-Sinkhorn algorithm. Although the results on the EOT problem presented above are formulated on continuous state spaces (*i.e.*, $\mathbb{R}^d$), the primal and dual formulations (1.49) and (1.50) remain valid in the particular case of discrete state spaces, by replacing the Lebesgue measure with the counting measure, so that integrals become finite sums. This setting is especially relevant when μ and ν are represented by N and M

samples. In this case, the Schrödinger potentials f and g reduce to vectors in $\mathbb{R}^N$ and $\mathbb{R}^M$, respectively, while the transport plan π becomes a matrix in $[0,1]^{N\times M}$, which is doubly stochastic in the sense that its rows and columns sum to one, reflecting the marginal constraints. Importantly, the expression of the EOT plan in terms of dual potentials given in (1.51) continues to hold in the discrete setting [Peyré et al., 2019, Proposition 4.3].

By explicitly incorporating the marginal constraints through the dual potentials, formulation (1.51) can be recast as a matrix scaling problem for arbitrary cost functions. An iterative algorithm to solve this problem was proposed early on, known as Iterative Proportional Fitting (IPF) [Deming and Stephan, 1940]. In this procedure, the vectors corresponding to f and g are optimized alternately by enforcing the associated marginal constraint (μ for f and ν for g), which amounts to simple elementwise rescaling at each iteration. The proof of linear convergence is due to Sinkhorn and Knopp [1967], giving rise to the alternative name *Sinkhorn algorithm*. In machine learning, this algorithm has gained renewed interest following the seminal work of Cuturi [2013], which demonstrated its effectiveness in large-scale settings through GPU-friendly implementations. As in the unregularized case, one may compute the barycentric projection associated with the EOT plan obtained via Sinkhorn iterations, which is shown to converge to the Monge map in the quadratic cost setting as $N, M \to \infty$ and $\varepsilon \to 0$ [Pooladian and Niles-Weed, 2021].

In continuous state spaces, the IPF algorithm was independently proposed by Fortet [1940], Kullback [1968], following similar principles of alternating updates of the Kantorovitch potentials f and g while enforcing the corresponding marginal constraints. In this setting, Ruschendorf [1995] established convergence under mild assumptions on μ, ν, and c, without specifying a convergence rate. More recently, this result was sharpened by Greco et al. [2023] for compact costs and compactly supported marginals, and further extended by Conforti et al. [2023] to more general settings, including the quadratic cost. However, although the IPF updates admit explicit expressions in continuous state spaces, they involve intractable integrals over $\mathbb{R}^d$, which restricts their use to a theoretical framework.

In both discrete and continuous settings, the IPF algorithm can be interpreted as an iterative KL projection method, by leveraging the primal formulation. Indeed, by simple manipulations of the entropy term in (1.49), one can show that solving the EOT problem is equivalent to the KL projection problem

$$\operatorname*{arg\,min}_{\pi\in\Pi(\mu,\nu)} \mathrm{KL}(\pi|\pi^{\mathrm{ref}}) \text{ where } \pi^{\mathrm{ref}}(x,y) \propto \mu(x)\exp(-c(x,y)/\varepsilon)\ , \tag{1.54}$$

which can be viewed as a variational inference problem on $\mathscr{P}((\mathbb{R}^d)^2)$, seeking the distribution closest to π^{ref} in the KL sense under the marginal constraints imposed by μ and ν. Since $\Pi(\mu,\nu)$ is the non-empty intersection of two closed convex sets in $\mathscr{P}((\mathbb{R}^d)^2)$, denoted by $\Pi(\mu)$ and $\Pi(\nu)$ for the respective marginal constraints, problem (1.54) can be solved via iterative Bregman projections [Bregman, 1967], with the KL divergence acting as the associated Bregman divergence. This yields a sequence of transport plans $\{\pi^n\}_{n\in\mathbb{N}}$ that is guaranteed to converge to the solution of (1.54). Starting from $\pi^0 = \pi^{\mathrm{ref}} \in \Pi(\mu)$, the updates read, for any $n \in \mathbb{N}$,

$$\begin{aligned}
\pi^{2n+1} &= \operatorname*{arg\,min}_{\pi\in\Pi(\nu)} \mathrm{KL}(\pi|\pi^{2n}) \quad \text{solved by} \quad \pi^{2n+1}(x,y) = \nu(y)\pi^{2n}(x|y)\ , \\
\pi^{2n+2} &= \operatorname*{arg\,min}_{\pi\in\Pi(\mu)} \mathrm{KL}(\pi|\pi^{2n+1}) \quad \text{solved by} \quad \pi^{2n+2}(x,y) = \mu(x)\pi^{2n+1}(y|x)\ .
\end{aligned} \tag{1.55}$$

Although these updates are tractable (obtained by applying the chain rule for the KL divergence), they remain non-computable in continuous settings. Informally, this procedure iteratively projects the current estimate of the EOT plan onto the marginal constraint sets under the KL divergence, and can be viewed as a fixed-point iteration. While this perspective may appear distinct from the IPF algorithm, the two procedures in fact generate identical iterates [Benamou et al., 2015, Section 3.1]. Beyond providing an alternative interpretation, this KL-based formulation naturally bridges the static and dynamic formulations of OT, the latter being closely connected to diffusion models, as discussed next.

1.5.2 Dynamical formulation

For the remainder of this section, we focus on the OT framework with quadratic cost c. In this setting, the Monge and Kantorovitch formulations admit the same solution. We now establish the connection between the standard OT formulation presented in Section 1.5.1, commonly referred to as the static formulation, and the dynamical formulation of OT, which transports μ to ν through time-inhomogeneous dynamics defined on a fixed time interval $[0,T]$, with $T > 0$. These dynamics may be deterministic in the OT setting or stochastic in the entropy-regularized case.

To introduce this framework, we first define additional notation. We denote by $\mathcal{U} = \{\mathbf{u} : (t, x) \in [0, T] \times \mathbb{R}^d \mapsto \mathbf{u}_t(x) \in \mathbb{R}^d\}$ the set of time-dependent vector fields on $[0, T]$, and by $\mathrm{C}([0, T], \mathbb{R}^d)$ the space of continuous functions from $[0, T]$ to $\mathbb{R}^d$. Hence, $\mathscr{P}(\mathrm{C}([0, T], \mathbb{R}^d))$ denotes the set of probability measures on path space.

Benamou-Brenier problem. Benamou and Brenier [2000] showed that the OT problem can be equivalently reformulated in a dynamical manner through the following variational problem, defined in terms of an ODE $(X_t^{\mathbf{u}})_{t\in[0,T]}$ with prescribed marginal constraints at times 0 and T,

$$\operatorname*{arg\,min}_{u\in\mathcal{U}} \int_0^T \frac{1}{2}\mathbb{E}[\|\mathbf{u}_t(X_t^{\mathbf{u}})\|_2^2]\mathrm{d}t \quad \text{such that} \quad \mathrm{d}X_t^{\mathbf{u}} = \mathbf{u}_t(X_t^{\mathbf{u}})\mathrm{d}t \ , \ X_0^{\mathbf{u}} \sim \mu \ , \ X_T^{\mathbf{u}} \sim \nu \ . \tag{1.56}$$

That is, dynamical OT seeks vector fields $\mathbf{u}$ whose induced flows transport μ to ν while minimizing the total kinetic energy over the time interval $[0, T]$. Under mild assumptions on μ and ν, Benamou and Brenier [2000] proves that (1.56) admits a unique solution $\mathbf{u}^\star$, which is directly related to the Monge map $\mathrm{T}^\star$ through $\mathbf{u}_t^\star(x) = (\mathrm{T}^\star(x) - x)/T$. Notably, $\mathbf{u}^\star$ does not depend on time, resulting in straight-line trajectories. From a computational perspective, this implies that, if $\mathbf{u}^\star$ were available, the OT dynamics could be simulated in a single Euler step from $t = 0$ to $t = T$, without any time discretization error, thereby recovering the OT map as $\mathrm{T}^\star(x) = x + T\mathbf{u}_t^\star(x)$.

Schrödinger Bridge problem. An analogous correspondence holds between the entropy-regularized OT problem and the dynamical Schrödinger Bridge (SB) problem [Schrödinger, 1932]. The latter is formulated as a variational problem over controlled SDEs $(X_t^{\mathbf{u}})_{t\in[0,T]}$, with marginal constraints at times 0 and T,

$$\operatorname*{arg\,min}_{u\in\mathcal{U}} \int_0^T \frac{1}{2}\mathbb{E}[\|\mathbf{u}_t(X_t^{\mathbf{u}})\|_2^2]\mathrm{d}t \quad \text{such that} \quad \mathrm{d}X_t^{\mathbf{u}} = \mathbf{u}_t(X_t^{\mathbf{u}})\mathrm{d}t + \sigma\mathrm{d}B_t \ , \ X_0^{\mathbf{u}} \sim \mu \ , \ X_T^{\mathbf{u}} \sim \nu \ , \tag{1.57}$$

where $\sigma > 0$ is a scaling parameter. By Girsanov's theorem, problem (1.57) can be reformulated as a variational inference problem on path space, with fixed initial and terminal marginals,

$$\operatorname*{arg\,min}_{\substack{\mathbb{P}\in\mathscr{P}(\mathrm{C}([0,T],\mathbb{R}^d))\\ \mathbb{P}_0=\mu,\mathbb{P}_T=\nu}} \mathrm{KL}(\mathbb{P}|\mathbb{Q}) \ , \tag{1.58}$$

where the so-called reference path measure $\mathbb{Q}$ is associated with the scaled Brownian motion $(\sigma B_t)_{t\in[0,T]}$ initialized at $X_0 \sim \mu$, and the KL divergence is defined on path space as $\mathrm{KL}(\mathbb{P}|\mathbb{Q}) = \int \log(\frac{\mathrm{d}\mathbb{P}}{\mathrm{d}\mathbb{Q}})\mathrm{d}\mathbb{P}$. Using the chain rule for the KL divergence, the objective can be decomposed as

$$\mathrm{KL}(\mathbb{P}|\mathbb{Q}) = \mathrm{KL}(\mathbb{P}_{0,T}|\mathbb{Q}_{0,T}) + \mathbb{E}_{\mathbb{P}_{0,T}}[\mathrm{KL}(\mathbb{P}_{|0,T}|\mathbb{Q}_{|0,T})] \ ,$$

where $\mathbb{P}_{|0,T}$ and $\mathbb{Q}_{|0,T}$ denote the conditional path measures given the initial and terminal states. The second term does not involve the marginal constraints and can be set to zero at optimality. As a result, the solution of (1.57) can be written as $\mathbb{P}^\star = \mathbb{P}_{0,T}^\star\mathbb{Q}_{|0,T}$, where $\mathbb{P}_{0,T}^\star$ solves the static Schrödinger bridge problem [Léonard, 2014b],

$$\operatorname*{arg\,min}_{\pi\in\Pi(\mu,\nu)} \mathrm{KL}(\pi|\mathbb{Q}_{0,T}) \ . \tag{1.59}$$

Since $\mathbb{Q}_{0,T}(x, y) = \mu(x)\mathrm{N}(y; x, \sigma^2 T\mathrm{I}_d)$, this establishes an exact equivalence between the Schrödinger bridge problem (1.59) and the entropy-regularized OT problem (1.54) with quadratic cost, by identifying $\varepsilon = \sigma^2 T$.

Solving optimal transport via diffusion models. A natural question is how the dynamical formulation can be leveraged to solve OT and EOT problems. Recent works have exploited connections with diffusion models (DMs) and flow matching models (FMs), which we briefly review below. This overview is not meant to be exhaustive, but rather to present methods that provide useful context for the contributions developed in Part III. To highlight these connections within a unified perspective, we adopt the stochastic interpolant framework introduced at the end of Section 1.4.1, whose objective is to construct deterministic or stochastic dynamics that transport $X_0 \sim \mu$ to $X_T \sim \nu$ over the time interval $[0, T]$, in either the forward or backward direction, using training samples $(X_0, X_T) \sim \pi_{0,T}^{\text{init}}$, where $\pi_{0,T}^{\text{init}} \in \Pi(\mu, \nu)$.

Unregularized setting. In the OT case, equivalently the Benamou-Brenier formulation, it can be shown that the OT plan $\pi^\star$, solution of (1.45), is a fixed point of the SI training procedure[5] when using the *linear interpolant*

[5]Here, we consider an idealized training procedure, neglecting approximation errors induced by the use of neural networks in practice.

($a_t = 1 - t/T$, $b_t = t/T$, $c_t = 0$ in (1.33)) [Tong et al., 2023, Proposition 3.4]. That is, setting $\pi^{\text{init}}_{0,T} = \pi^\star$ yields the dynamical OT solution after optimization. In theory, access to $\pi^\star$ would therefore allow one to train an FM model that generates straight trajectories, which can be simulated in a single exact discretization step and correspond to evaluations of the Monge map up to linear terms. In practice, however, the true OT plan $\pi^\star$ is unavailable. One may nonetheless consider computable approximations of $\pi^\star$ to initialize the SI training, with the expectation that the learned drift approximates the optimal one $\mathbf{u}^\star$. To this end, Pooladian et al. [2023], Tong et al. [2023] propose using minibatch approximations of the OT plan [Fatras et al., 2021], which however require very large batch sizes to closely approximate $\pi^\star$ [Bernton et al., 2019b], rendering them inaccurate.

Along similar lines, and still using the same stochastic interpolant, Liu et al. [2023], Liu [2022] introduce an iterative approach known as *Rectified flow*. This method consists of a sequence of SI training problems, where the coupling generated by the dynamics at the previous iteration is used as the training coupling for the next one. Liu et al. [2023] show that this procedure monotonically decreases the average transport cost, leading to progressively straighter trajectories. However, convergence to the optimal transport plan is not guaranteed. This issue is addressed in Liu [2022] by restricting the optimization domain to gradients of real-valued neural networks, which theoretically ensures convergence.

Regularized setting. In the entropy-regularized case, equivalently the Schrödinger Bridge problem, one can similarly show that the EOT plan $\pi^\star$ is a fixed point of the SI training procedure[5] when using the *Brownian bridge interpolant* ($a_t = 1 - t/T$, $b_t = t/T$, $c_t = \sqrt{t(T-t)/T}$ in (1.33)) [Tong et al., 2024, Proposition 3.4]. That is, choosing $\pi^{\text{init}}_{0,T} = \pi^\star$ yields the dynamical EOT solution after optimization. This result is particularly appealing in practice when the marginals μ and ν are represented by samples, since the EOT plan can be efficiently approximated using modern implementations of the Sinkhorn algorithm [Cuturi and Doucet, 2014]. When the number of samples becomes very large, however, this approach may still be computationally demanding, which has motivated the development of iterative alternatives, in the same spirit as in the unregularized case.

A natural alternative is the dynamical analogue of the IPF algorithm associated with (1.58) [Fortet, 1940], defined through iterative KL projections on path space under boundary marginal constraints imposed by μ and ν. This yields a sequence of path measures $\{\mathbb{P}^n\}_{n\in\mathbb{N}}$, initialized as $\mathbb{P}^0 = \mathbb{Q}$ with $\mathbb{P}^0_0 = \mu$, and satisfying for any $n \in \mathbb{N}$

$$\begin{aligned}
\mathbb{P}^{2n+1} &= \operatorname*{arg\,min}_{\substack{\mathbb{P}\in\mathscr{P}(\mathrm{C}([0,T],\mathbb{R}^d))\\ \mathbb{P}_T=\nu}} \mathrm{KL}(\mathbb{P}|\mathbb{P}^{2n}) \quad \text{solved by} \quad \mathbb{P}^{2n+1} = \nu\mathbb{P}^{2n}_{|T}\,, \\
\mathbb{P}^{2n+2} &= \operatorname*{arg\,min}_{\substack{\mathbb{P}\in\mathscr{P}(\mathrm{C}([0,T],\mathbb{R}^d))\\ \mathbb{P}_0=\mu}} \mathrm{KL}(\mathbb{P}|\mathbb{P}^{2n+1}) \quad \text{solved by} \quad \mathbb{P}^{2n+2} = \mu\mathbb{P}^{2n+1}_{|0}\,.
\end{aligned} \tag{1.60}$$

These dynamical updates correspond exactly to the static updates in (1.55), with the identification $\mathbb{P}^n_{0,T} = \pi^n$ for all $n \in \mathbb{N}$. At first glance, these updates appear intractable, similarly to their static counterparts. However, one can observe that, by recursion, each $\mathbb{P}^n$ is Markov, inheriting this property from $\mathbb{Q}$, and that the associated stochastic dynamics corresponds to the time-reversal of the dynamics of the previous iterate $\mathbb{P}^{n-1}$, initialized from the marginal imposed at step n, namely μ for even n and ν for odd n. This naturally defines two opposite directions for the dynamics, a forward one associated with $\{\mathbb{P}^n\}_{n \text{ odd}}$, and a backward one associated with $\{\mathbb{P}^n\}_{n \text{ even}}$.

This observation constitutes the key insight underlying Diffusion Schrödinger Bridge (DSB) [Bortoli et al., 2021], which implements the dynamical IPF procedure by learning each $\mathbb{P}^n$ numerically through a score matching objective, using samples generated from the previous iterate $\mathbb{P}^{n-1}$. In particular, the first iteration of DSB corresponds to training a diffusion model with a Variance Exploding-like noise schedule. From the second iteration onward, DSB aims to learn the time reversal of neural dynamics, which falls outside the standard DM framework. More precisely, the transition kernels of these dynamics are not tractable, making the direct application of the Tweedie formula (1.20) infeasible. To overcome this limitation, Bortoli et al. [2021] introduce an alternative objective, known as the *Mean Matching* loss. This loss applies the Tweedie formula locally in time intervals where the transition kernels of the learned dynamics can be approximated by Gaussians, thus recovering tractability. While effective in practice, this approach has a significant computational drawback, as it is not simulation free. Indeed, computing the Mean Matching loss requires storing full neural trajectories, rather than only their endpoints as in Rectified Flows, which is computationally prohibitive.

To address this limitation, Peluchetti [2023], Shi et al. [2023] proposed a new methodology based on iterative KL projections, in the spirit of IPF, known as Iterative Markovian Fitting (IMF). This approach relies on a fundamental result from Léonard [2014b], which states that the solution to the Schrödinger Bridge problem is

the unique path measure $\mathbb{P}$ on $[0,T]$ satisfying the following four conditions (a) $\mathbb{P}$ is Markov, (b) $\mathbb{P}_{|0,T} = \mathbb{Q}_{|0,T}$, (c) $\mathbb{P}_0 = \mu$, (d) $\mathbb{P}_T = \nu$. In the IPF algorithm, KL projections alternate between enforcing the marginal constraints (c) and (d), while the constraints (a) and (b) are automatically preserved. The IMF algorithm reverses this logic, alternating KL projections onto the constraints (a) and (b), while the marginal constraints (c) and (d) are enforced at every iteration. Concretely, it defines a sequence $\{\mathbb{P}^n\}_{n\in\mathbb{N}}$ with $\mathbb{P}^0$ chosen such that $\mathbb{P}^0_0 = \mu$ and $\mathbb{P}^0_T = \nu$, and for any $n \in \mathbb{N}$

$$\begin{aligned}\mathbb{P}^{2n+1} &= \operatorname*{arg\,min}_{\substack{\mathbb{P}\in\mathscr{P}(\mathrm{C}([0,T],\mathbb{R}^d))\\ \mathbb{P}\text{ Markov}}} \mathrm{KL}(\mathbb{P}|\mathbb{P}^{2n}) \,, \\ \mathbb{P}^{2n+2} &= \operatorname*{arg\,min}_{\substack{\mathbb{P}\in\mathscr{P}(\mathrm{C}([0,T],\mathbb{R}^d))\\ \mathbb{P}_{|0,T}=\mathbb{Q}_{|0,T}}} \mathrm{KL}(\mathbb{P}|\mathbb{P}^{2n+1}) \quad \text{solved by} \quad \mathbb{P}^{2n+2} = \mathbb{P}^{2n+1}_{0,T}\mathbb{Q}_{|0,T} \,.\end{aligned} \tag{1.61}$$

Shi et al. [2023] show that $\mathbb{P}^{2n+1}$ corresponds exactly to the Markovian projection of $\mathbb{P}^{2n}$ (mentioned in Section 1.4.1), while $\mathbb{P}^{2n+2}$ is the stochastic interpolant on $[0,T]$ induced by the Brownian bridge and the coupling $\mathbb{P}^{2n+1}_{0,T}$ (referred to as the Reciprocal projection of $\mathbb{P}^{2n+1}$). Consequently, the IMF algorithm can be viewed as the stochastic analogue of Rectified Flows, implementing an iterative SI training procedure where the coupling produced by the dynamics at one stage serves as the training coupling for the next. The resulting practical algorithm, termed Diffusion Schrödinger Bridge Matching (DSBM), only requires storing the initial and terminal points of simulated trajectories, leading to a drastic reduction in computational cost compared to DSB, while achieving improved empirical performance. Although IMF is relatively recent, it has already attracted significant attention, with theoretical guarantees, Silveri et al. [2025] showing convergence rates comparable to IPF, as well as practical refinements [Bortoli et al., 2024b, Peluchetti, 2025].

Overall, diffusion-based approaches have demonstrated strong empirical performance for OT and EOT in continuous state spaces, especially when marginals are represented by large sample sets. However, they remain primarily limited to the standard bi-marginal setting. This observation motivates the following question.

(**Q6**) How can recent diffusion-based methods developed for bi-marginal EOT with quadratic cost in continuous state spaces be extended to the multi-marginal setting ? Can we define a notion of Schrödinger bridge in this case ? In particular, how can such approaches be used to compute Wasserstein barycenters ? Beyond algorithmic design, what theoretical guarantees can be established for these methods ? These questions are addressed in Chapter 7, where we propose a multi-marginal extension of the DSB algorithm.[6]

Generalized Schrödinger Bridge. More recently, Liu et al. [2024] introduced a generalized version of the dynamical Schrödinger Bridge problem, coined Generalized Schrödinger Bridge (gSB), by incorporating an additional potential term $V : [0,T] \times \mathbb{R}^d \to \mathbb{R}$ as

$$\operatorname*{arg\,min}_{u\in\mathcal{U}} \int_0^T \mathbb{E}\left[\frac{1}{2}\left\|\mathbf{u}_t(X^{\mathbf{u}}_t)\right\|_2^2 + V_t(X^{\mathbf{u}}_t)\right]\mathrm{d}t \quad \text{such that} \quad \mathrm{d}X^{\mathbf{u}}_t = \mathbf{u}_t(X^{\mathbf{u}}_t)\mathrm{d}t + \sigma\mathrm{d}B_t \,, X^{\mathbf{u}}_0 \sim \mu \,, X^{\mathbf{u}}_T \sim \nu \,. \tag{1.62}$$

The potential V acts as an additional penalization term on the dynamics, for instance encoding repulsive or attractive regions in trajectory inference problems. The classical Schrödinger Bridge problem is recovered as a special case by taking $V \equiv 0$. As shown in Chapter 8, this problem can again be rewritten in SB form, using Girsanov's theorem, as

$$\operatorname*{arg\,min}_{\substack{\mathbb{P}\in\mathscr{P}(\mathrm{C}([0,T],\mathbb{R}^d))\\ \mathbb{P}_0=\mu,\mathbb{P}_T=\nu}} \mathrm{KL}(\mathbb{P}|\mathbb{Q}^V) \,, \tag{1.63}$$

where the reference path measure $\mathbb{Q}^V$ is defined by $\mathbb{Q}^V(x_{[0:T]}) \propto \exp(-\int_0^T \frac{V_t(x_t)}{\sigma^2}\mathrm{d}t)\mathbb{Q}(x_{[0:T]})$ and is no longer a scaled Brownian motion. Although, in principle, the IMF algorithm could be applied to this formulation to obtain an asymptotic solution of (1.62), neither the Markov nor the Reciprocal projections can be computed using the original DSBM framework, which assumes a Brownian reference measure. To address this, Liu et al. [2024] propose a method termed Generalized Schrödinger Bridge Matching (GSBM), based on diffusion-like backbones similar to those of DSBM. However, the precise relationship between this scheme and the theoretical IMF framework remains unclear, as the proposed updates act more as proxies than exact counterparts of IMF projections. This leaves room for methodological improvements and motivates the following questions.

[6] Note that, at the time of publication of the corresponding paper, the DSBM methodology was not yet available, which motivated our choice of DSB as a backbone. More recently, Howard et al. [2025] proposed an IMF-based extension of our approach, retaining our formulation of the multi-marginal dynamical SB problem while replacing DSB with DSBM, leading to improved numerical performance.

(**Q7**) Building on Liu et al. [2024], can we design an IMF-based diffusion methodology to solve the generalized Schrödinger Bridge problem with exact Markovian and Reciprocal projection steps ? Moreover, can a corresponding framework be developed directly in discrete time, an aspect that remains largely unexplored in existing Schrödinger Bridge formulations, in order to avoid discretization errors inherent in translating continuous-time models to practical implementations ? These questions are addressed in Chapter 8.

1.6 Outline and contributions of this thesis

The content of this thesis is motivated by research questions (**Q**-1-2-3-4-5-6-7), each of which is investigated in the following papers that constitute the remaining chapters of the manuscript; the superscript * denotes equal contribution.

1. Unbiased constrained sampling with Self-Concordant Barrier Hamiltonian Monte Carlo [Noble et al., 2023b].
   Maxence Noble, Valentin de Bortoli, Alain Oliviero Durmus.
   Conference on Neural Information Processing Systems 37 (NeurIPS) (2023).

2. Improving the evaluation of samplers on multi-modal targets [Grenioux et al., 2025].
   Louis Grenioux*, **Maxence Noble***, Marylou Gabrié.
   Frontiers in Probabilistic Inference: Learning meets Sampling (ICLR Workshop) (2025).

3. Stochastic Localization via Iterative Posterior Sampling [Grenioux et al., 2024].
   Louis Grenioux*, **Maxence Noble***, Marylou Gabrié, Alain Oliviero Durmus.
   International Conference on Machine Learning 41 (ICML) (2024).

4. Learned Reference-based Diffusion Sampler for multi-modal distributions [Noble et al., 2025].
   Maxence Noble*, Louis Grenioux*, Marylou Gabrié, Alain Oliviero Durmus.
   International Conference on Learning Representations 13 (ICLR) (2025).

5. Diffusion-based Annealed Boltzmann Generators : benefits, pitfalls and hopes [Noble and Grenioux, 2026].
   Louis Grenioux*, **Maxence Noble***.
   Under review.

6. Tree-based Diffusion Schrödinger Bridge with applications to Wasserstein Barycenters [Noble et al., 2023a].
   Maxence Noble, Valentin de Bortoli, Arnaud Doucet, Alain Oliviero Durmus.
   Conference on Neural Information Processing Systems 37 (NeurIPS) (2023).

7. Twisted Schrödinger Bridge Matching : a unified continuous- and discrete-time framework.
   Maxence Noble, Yazid Janati El Idrissi, Alain Oliviero Durmus.
   Preliminary work.

Although not directly aligned with the core research axes of this thesis, yet still related to diffusion models, I have first-authored the following paper[7], restated in Appendix H, which introduces a fast-inference diffusion-based methodology specifically designed for image super-resolution:

1. Fast, faithful and photorealistic diffusion-based image super-resolution with enhanced Flow Map models [Noble et al., 2026].
   Maxence Noble, Gonzalo Iñaki Quintana, Benjamin Aubin, Clément Chadebec.
   Technical report.

In addition, although not included in this thesis, I have co-authored the following papers. They concern, respectively, non-asymptotic convergence guarantees for the Sinkhorn algorithm in compact settings and an extension of the methodology introduced in Chapter 4 to Riemannian state spaces:

1. Non-asymptotic convergence bounds for Sinkhorn iterates and their gradients: a coupling approach [Greco et al., 2023].
   Giacomo Greco, **Maxence Noble**, Giovanni Conforti, Alain Oliviero Durmus
   Annual Conference on Learning Theory 36 (COLT) (2023).

2. Sampling from multi-modal distributions on Riemannian manifolds with training-free stochastic interpolants [Durmus et al., 2026].

[7] This work was carried out during a six-month internship at Jasper AI (Paris, France).

Alain Oliviero Durmus, **Maxence Noble**, Thibaut Pellerin.
Under review.

Below, we provide a summary of the contributions presented in each chapter. For clarity, the chapters are organized into three main parts, reflecting the principal research axes of this thesis:

- Part I : **On Fundamental Challenges in Sampling** (Chapters 2 and 3)
- Part II : **Sampling methodologies based on diffusion models** (Chapters 4 to 6)
- Part III : **Optimal transport methodologies based on diffusion models** (Chapters 7 and 8)

Part I - Chapter 2 / (Q1) - Unbiased constrained sampling with Self-Concordant Barrier Hamiltonian Monte Carlo. In this chapter, we investigate the fundamental problem of sampling from constrained distributions. More specifically, we aim at sampling from a target distribution π supported on M, an open, convex, and bounded subset of $\mathbb{R}^d$, assumed to be equipped with a self-concordant barrier $\phi : \mathsf{M} \to \mathbb{R}$, *i.e.*, a three-times differentiable convex function satisfying specific regularity properties on its derivatives and exhibiting a "barrier" behavior [Nesterov and Nemirovskii, 1994]. More precisely, $\phi(x)$, $\|\nabla\phi(x)\|$, and $\left\|\nabla^2\phi(x)\right\|$ tend to ∞ as x approaches the boundary of M. This setting is of particular interest since it includes polytopes, for which such a function ϕ can always be constructed [Nesterov and Nemirovski, 1998]. For instance, if $\mathsf{M} = \{x \in \mathbb{R}^d : Ax < b\}$ for a given matrix $A \in \mathbb{R}^{m\times d}$ and vector $b \in \mathbb{R}^m$, one may define the *logarithmic barrier* $\phi : x \mapsto -\sum_{i=1}^m \log(b_i - A_i^\top x)$, which is easily implementable even in high-dimensional settings. A key property of self-concordant barriers is that they naturally encode the geometry of the feasible set: indeed, the Hessian $\mathfrak{g} : x \mapsto \nabla^2\phi(x)$ defines a Riemannian metric on M, viewed as a d-dimensional submanifold of $\mathbb{R}^d$. This geometric structure can then be naturally incorporated into Monte Carlo sampling methods, which are traditionally designed for distributions supported on the full space $\mathbb{R}^d$. We follow this geometry-aware sampling approach and build upon the Riemannian instance of HMC (RMHMC) introduced by Girolami and Calderhead [2011] for a general Riemannian metric, which naturally incorporates $\mathfrak{g}$ into the Hamiltonian through (1.7). As recalled in Section 1.3.4, the standard implementation of RMHMC corresponds to a Metropolis-Hastings scheme based on a deterministic proposal map obtained by discretizing the Riemannian Hamiltonian dynamics (with an additional momentum reversal) using a symplectic and implicit numerical integrator. However, Girolami and Calderhead [2011] do not provide a detailed theoretical analysis establishing the exactness of the resulting algorithm. This chapter aims to address this limitation by designing an instance of RMHMC in the specific setting induced by self-concordant barriers, which we call BHMC, supported by both theoretical and numerical results that prove its unbiasedness with respect to the target distribution,

As a first contribution, we introduce an idealized version of BHMC, referred to as *continuous*-BHMC (c-BHMC), in which we assume that the Riemannian Hamiltonian flow can be simulated exactly. Under minimal assumptions on M and ϕ, we theoretically show that this flow is well defined up to a possible "explosion time" induced by the barrier, corresponding to the flow reaching the boundary of M when this explosion time is finite. These findings allow us to formally define c-BHMC. Building on the idealized formulation of HMC, where the use of the exact flow in the MH proposal leads to an acceptance rate identically equal to one, *i.e.*, without an actual acceptance-rejection step, we nevertheless introduce an additional acceptance-rejection mechanism to verify that the proposal map is indeed involutive (namely to ensure that the flow does not experience an explosion behavior). We refer to this mechanism as the "involution checking step", and we prove that, when it is included, c-BHMC admits π as its invariant distribution.

Although c-BHMC is not implementable in practice, it provides essential intuition for its translation to a practical setting, which constitutes the second main contribution of this chapter. To this end, we follow the same general principle as RMHMC: the MH proposal map approximates the Hamiltonian dynamics (up to momentum reversal) over small time intervals of size $h > 0$ using the Generalized Leapfrog (GL) symplectic integrator [Hairer et al., 2006, Theorem 3.4], which leads to implicit updates solved via a fixed-point scheme. In particular, such an approach was previously proposed by Kook et al. [2022] in the same self-concordant setting, with the only difference being the use of the Implicit Midpoint (IM) integrator, which is also symplectic [Hairer et al., 2006, Theorem 3.5], resulting in the Constrained RHMC (CRHMC) algorithm. However, the design of RMHMC and CRHMC relies on the tacit assumption that solving the implicit integrator, either GL or IM, yields a unique solution for any value of $h > 0$. In this chapter, we explicitly show that this assumption is false: for instance, solutions to the implicit integrator may include, with non-zero probability, points lying outside of M for certain values of h, a phenomenon that cannot occur under exact Hamiltonian integration. As a consequence, RMHMC and CRHMC may exhibit intrinsic bias when the step size h is poorly chosen, thereby leading to asymptotic bias.

Motivated by this observation, we revisit RMHMC from a novel and more principled theoretical perspective, in which the implicit GL integrator is treated in its canonical form as a set-valued map rather than a single-valued function, as implicitly assumed in previous works. Within this framework, we establish that this integrator can be *locally* identified with a continuously differentiable diffeomorphism when h is sufficiently small, thereby justifying its use within a MH calibration scheme in this regime. This guarantee, however, no longer holds when h is too large. Inspired by this theoretical insight, we introduce a substantial modification to RMHMC and CRHMC, leading to the *numerical* version of BHMC (n-BHMC). Specifically, we propose adding an acceptance-rejection step after the implicit integration has been computed via fixed-point iterations, with two objectives aligned with HMC requirements: (i) verifying that the proposed state lies in M, and (ii) ensuring that the corresponding numerical map is locally involutive. To verify condition (ii), we check the involution condition of the numerical integrator at the current state up to a prescribed tolerance threshold. Taking advantage of the Riemannian formulation, we replace the standard Euclidean norm by the norm induced by $\mathfrak{g}$ on M in the self-concordant setting, which is more sensitive to the geometric pathologies near the boundary of M. If either condition (i) or (ii) fails, the proposal is rejected. In analogy with c-BHMC, we refer to this mechanism as the "involution checking step", which is now fully tractable and implementable. With this additional step, we prove that n-BHMC admits π as its invariant distribution, thereby removing the bias induced by the use of implicit integrators, a result that is further confirmed by our numerical experiments. To the best of our knowledge, n-BHMC constitutes the first instance of RMHMC for which invariance is guaranteed while explicitly accounting for the implicit nature of Hamiltonian integrators.

Part I - Chapter 3 / (Q2) - Improving the evaluation of samplers on multi-modal targets. In this chapter, we investigate the second fundamental challenge of sampling, namely sampling from multi-modal distributions. We notably propose a general and simple framework to evaluate the quality of sampling methodologies in this setting in a way that enables fair and interpretable comparisons. This reflection-oriented contribution naturally paves the way for the diffusion-based sampling methodologies introduced in Part II. We first identify two major sources of difficulty encountered in practice when dealing with multi-modal distributions: (i) locating the modes, which is essential to prevent samplers from wasting computational effort exploring regions of negligible probability mass, and (ii) generating samples according to the true relative importance of the modes (*i.e.*, the mode weights), which is unknown in most practical settings. While the first problem is known to be NP-hard, as it amounts to identifying local maxima of a possibly high-dimensional function, this work deliberately focuses on the second difficulty, which remains challenging even when the locations of the modes are assumed to be known. In particular, we illustrate why standard MCMC methods intrinsically fail at this task, despite their widespread use in practice.

As a first contribution, we introduce a formal definition of mode weight for a given multi-modal distribution and propose to focus throughout this chapter on a simple yet informative representative case: a two-component Gaussian mixture, corresponding to two modes with unequal weights. In order to capture, at a coarse level, the main factors governing the difficulty of multi-modality in more general settings, we allow two hyperparameters to vary freely: the distance between the modes and the ambient dimension, while keeping the mode weight ratio fixed, yielding a broad family of multi-modal distributions. The interest of this family lies in its complete tractability: samples can be generated easily, and the mode weights are exactly known. We therefore propose using this family as a systematic testbed for samplers, focusing on their ability to accurately recover mode weights through Monte Carlo estimates based on the generated samples, thereby considering both estimation bias and variance as metrics. This criterion is inexpensive to compute and easy to interpret, making it a practical and controlled benchmark. Although these test distributions are considerably simpler than more realistic targets, such as distributions with an exponential number of modes in the dimension or with highly ill-conditioned modes, they constitute a necessary baseline. Indeed, samplers should be expected to succeed in such elementary scenarios before being considered reliable in more complex, high-dimensional multi-modal settings.

Based on this protocol, we propose, as a second contribution, to evaluate several widely used sampling methods along two axes of increasing difficulty for the test distributions: increasing inter-mode distance and increasing dimension. To ensure that the comparison is informative, all methods are carefully tuned using access to exact samples, a setting that is not feasible in practice, and are run under generous yet explicitly bounded computational budgets, making our evaluation intentionally optimistic. In particular, we refrain from applying method-specific tuning strategies that could potentially yield better results, in order to allow for a fair and general comparison across methods. The primary objective of this study is therefore not to establish a precise benchmark or ranking, but rather to extract methodological insights into multi-modality and to characterize the typical strengths and limitations of each sampler.

Our empirical results show that standard MCMC methods, such as MALA, consistently suffer from mode trapping, while traditional variational inference approaches based on unimodal variational families experience systematic mode collapse, leading to biased mode weight estimates with artificially low variance. On the other hand, we demonstrate that IS fundamentally struggles in high-dimensional settings due to importance weight degeneracy. This limitation also arises when using normalizing flows as proposal distributions, although their use can partially mitigate the issue. We further investigate annealed sampling approaches based on classical tempering paths, including Sequential Monte Carlo (SMC) and Replica Exchange (RE), and show that they handle multi-modality more effectively than standard approaches. Nevertheless, their performance deteriorates as the dimension increases, with RE exhibiting a slower degradation than SMC. Finally, we include two diffusion-based samplers in our comparison: DDS [Vargas et al., 2023a], an annealed variational inference method, and SLIPS, a training-free method relying solely on Monte Carlo principles, introduced in Chapter 4. While DDS performs comparably to, or slightly worse than, tempering-based methods, particularly in high-dimensional regimes, we show that SLIPS consistently outperforms all other approaches across the full range of tested conditions when optimally tuned.

The main conclusion of this chapter is that even in a highly favorable and controlled setting, characterized by careful tuning and ample computational resources, most sampling methods fail as either the dimension or the inter-mode distance increases. It is therefore reasonable to expect that, in more realistic scenarios involving a larger number of modes or more complex structures, performance would degrade even more rapidly. While our results suggest that diffusion-based annealed sampling methods are more promising for handling multi-modality at scale, a conclusion that aligns with recent developments in sampling literature, we advocate for the systematic use of this simple evaluation protocol as a means to assess the robustness and accuracy of such approaches.

Part II - Chapter 4 / (Q3) - Stochastic Localization via Iterative Posterior Sampling. In this chapter, we investigate the design of a novel diffusion-based sampling algorithm for drawing samples from an arbitrary multi-modal distribution, specified solely through its unnormalized density γ. As shown in Chapter 3, this task may pose significant challenges for standard annealed Monte Carlo methods. The originality of our approach lies in the fact that it is entirely training-free, a feature that is not commonly associated with diffusion models. In line with previous work Huang et al. [2024a], we propose to simulate a time-inhomogeneous stochastic dynamics that marginally follows a density path interpolating between an easy-to-sample distribution and the target distribution, relying exclusively on Monte Carlo techniques as its computational backbone. While Huang et al. [2024a] considers the standard diffusion density path given in (1.16), associated with the classical denoising diffusion process (1.19), we introduce a novel formulation that is more flexible, easier to implement and tune, and inspired by the literature on stochastic localization [Montanari, 2023].

We first define an *observation* stochastic process

$$Y_t = \alpha(t)X + \sigma W_t, \quad t \in [0,T], \quad T > 0\,, \tag{1.64}$$

where $X \sim \pi$, $(W_t)_{t\in[0,T]}$ is a standard Brownian motion, $\sigma > 0$ is a rescaling hyperparameter, and $\alpha : [0,T] \to \mathbb{R}_+$ is a scheduling function satisfying $\alpha(0) = 0$, with asymptotic properties $\alpha(t)t^{-1/2} \to 0$ as $t \to 0$, and $\alpha(t)t^{-1/2} \to \infty$ as $t \to T$. Under these conditions, the observation process is said to "localize" to π, in the informal sense that increasing values of t progressively reveal more information about π through Y_t. This phenomenon is formally captured by the convergence result $Y(t)/\alpha(t) \to \pi$ as $t \to T$. In principle, simulating $(Y_t)_{t\in[0,T]}$ up to time T would therefore yield samples from π; however, such a simulation is infeasible, since the definition of (1.64) requires prior access to samples from π.

Observing that it suffices to match the observation process only *marginally*, we propose to consider its Markovian projection (see Section 1.4.1), that is, the Markov process associated with the SDE

$$\mathrm{d}Y_t = \dot{\alpha}(t)u_t(Y_t)\mathrm{d}t + \sigma \mathrm{d}B_t, \quad Y_0 = 0\,, \quad \text{where} \quad u_t(y) = \mathbb{E}[X \mid Y_t = y]. \tag{1.65}$$

We prove that this process admits the same marginals as (1.64), denoted by $(p_t)_{t\in[0,T]}$. Here, $u_t(y)$ is defined as the mean of the posterior distribution associated with (1.64), given by Bayes' rule as $q_t(\cdot \mid y) \propto p_t(y \mid x)\gamma(x)$, where $p_t(\cdot \mid x) = \mathrm{N}(\alpha(t)x, \sigma^2 t \mathrm{I}_d)$ is the conditional distribution of Y_t given $X = x$. In this sense, u_t is a denoiser function and the resulting dynamics closely resembles the denoising diffusion process (1.19), where the drift term involving the score can be equivalently rewritten using a denoiser through the Tweedie formula (1.22). Nevertheless, the stochastic localization-based dynamics considered here is more interpretable and simpler to analyze, since its dependence to the signal-to-noise ratio of the observation process, verifying $\mathrm{SNR}(t) \propto \alpha(t)^2 t^{-1}$, is immediate. This allows us to consider a variety of denoising dynamics based on a flexible tuning of the SNR profile.

As in diffusion models, a key difficulty of the dynamics (1.65) is that the denoiser u_t is intractable. Leveraging its formulation as a conditional expectation, we propose to estimate $u_t(Y_t)$ on-the-fly during the simulation of the SDE, rather than learning the denoiser explicitly as in standard diffusion models. This is achieved by using MCMC methods targeting the posterior distribution $q_t(\cdot \mid Y_t)$. For this approach to be viable, the intermediate posterior sampling problems must be simpler than the original task of sampling from π. This motivates a detailed analysis of the behavior of $q_t(\cdot \mid y)$ as a function of time, which constitutes the first building block of our algorithm. As $t \to 0$, the posterior $q_t(\cdot \mid y)$ becomes increasingly independent of y (recalling that $Y_0 = 0$ is agnostic to π) and thus closely resembles to the prior π, inheriting its multi-modality, a regime in which standard MCMC methods are known to fail. In contrast, as $t \to T$, the localization phenomenon induces concentration of $q_t(\cdot \mid y)$ around $y/\alpha(t)$, rendering it log-concave and eventually collapsing to a Dirac mass at $t = T$. Exploiting this specific behavior, we propose to start the simulation of (1.65) at an intermediate time $t_0 \in (0, T)$ such that, for all $t > t_0$, the posterior $q_t(\cdot \mid y)$ is increasingly unimodal and therefore amenable to efficient sampling with standard MCMC methods.

This strategy requires the ability to accurately sample an initial condition $Y_{t_0} \sim p_{t_0}$, which constitutes the second building block of our algorithm. Although p_{t_0} is not available in closed form, we bypass this limitation by employing a first-order MCMC method that requires only access to $\nabla \log p_{t_0}$, namely the Unadjusted Langevin Algorithm (ULA). The true gradient is nevertheless replaced by a Monte Carlo estimate computed through posterior sampling, exploiting the Tweedie formula. As this procedure again relies on a standard MCMC technique (to sample from p_{t_0}), it raises the question of its validity for arbitrary choices of t_0. This leads us to analyze the complexity of sampling from p_{t_0} as a function of t_0. For small values of t_0, the Brownian term dominates the data-dependent term in (1.65), implying that p_{t_0} is log-concave. Conversely, as $t_0 \to T$, the data term dominates and p_{t_0} approaches a rescaled version of π, recovering its multi-modality. Choosing t_0 too large therefore leads to the failure of the Langevin initialization due to the multi-modality of p_{t_0}. This reveals a trade-off in the selection of t_0, which must simultaneously exploit the log-concavity of the marginal p_{t_0} and that of the posterior $q_{t_0}(\cdot \mid y)$ in order to justify the use of standard MCMC methods for both. We refer to this antagonistic phenomenon as the "duality of log-concavity", and we term the resulting algorithm SLIPS (Stochastic Localization via Iterative Posterior Sampling).

To guide the choice of the initialization time t_0, we provide a theoretical analysis of this duality for a broad class of multi-modal distributions, namely distributions that are assumed to be log-concave with Gaussian tails outside a compact set. This class includes Gaussian mixtures, which form the core test case studied in Chapter 3 where we prove they are challenging for classical sampling methods. Under these assumptions, we derive explicit expressions for a threshold time $t_p \in (0, T)$ below which p_t is log-concave, and a threshold time $t_q \in (0, T)$ above which $q_t(\cdot \mid y)$ is log-concave. Moreover, we identify a subclass of such distributions for which $t_q < t_p$, enabling the duality of log-concavity to fully apply over the time window $[t_q, t_p]$, from which a valid choice of t_0 can be made to initialize the SLIPS procedure. In addition to these theoretical results, we provide extensive empirical evidence demonstrating the performance of SLIPS across a wide range of multi-modal settings that largely extend beyond the scope of our theoretical assumptions. With appropriate tuning of t_0, we show that SLIPS significantly outperforms standard annealed Monte Carlo methods under comparable computational budgets, and is particularly robust to the multi-modality challenges highlighted in Chapter 3.

Part II - Chapter 5 / (Q4) - Learned Reference-based Diffusion Sampling for multi-modal distributions. Complementary to Chapter 4, this chapter also falls within the research axis of designing novel sampling algorithms for efficiently sampling from an arbitrary multi-modal distribution π, specified through its unnormalized density γ, using diffusion models as the main theoretical backbone. The guiding principle here is, however, different. Rather than assuming access only to γ, we adopt a setting aligned with the practical framework studied in Chapter 3, in which the locations of the modes of π, *i.e.*, the local maxima of γ, are assumed to be known. In particular, such prior knowledge allows us to run standard, and therefore inexpensive, MCMC methods within high-density regions. While these local samplers are expected to miss global information about the target, such as the relative importance of the modes, they nevertheless provide an efficient and accurate local description of the density landscape, for instance in terms of intra-mode variability. The objective of this chapter is to exploit this partial knowledge of the target distribution, in the form of uncalibrated samples from π, in order to train a diffusion model via annealed variational inference (Section 1.4.2-Paragraph A) in a way that overcomes the pervasive issue of mode collapse.

Our variational formulation follows the line of previous works Zhang and Chen [2022], Vargas et al. [2023a], Richter et al. [2023], which independently rely on a change-of-measure formula on path space of the form $\mathrm{d}\mathbb{P}^\star = \frac{\pi}{\pi^{\mathrm{ref}}} \mathrm{d}\mathbb{P}^{\mathrm{ref}}$. Here, $\mathbb{P}^\star$ and $\mathbb{P}^{\mathrm{ref}}$ denote the path measures associated with the noising diffusion process (1.17) initialized at π and at π^{ref}, respectively, where π^{ref} is a predefined Gaussian distribution, rendering $\mathbb{P}^{\mathrm{ref}}$

fully tractable. Using the KL divergence, this decomposition allows one to rewrite the variational objective (1.38) as a tractable stochastic optimal control problem, whose terminal cost is given by the log-density ratio $\log(\pi^{\text{ref}}/\gamma)$, a term that also appears when using the LV divergence [Richter et al., 2023]. A first contribution of this chapter is to show that these existing approaches can be viewed as particular instances of a more general formulation, in which π^{ref} is an arbitrary *reference* probability distribution sharing the same support as π. We prove that, provided access to the unnormalized density of π^{ref}, denoted γ^{ref}, as well as to the intermediate marginal scores $(\nabla \log p_t^{\text{ref}})_{t\in[0,T]}$, one can derive a tractable discrete-time training objective that is directly applicable in practice.

A core contribution of this chapter is to establish that the choice of the reference distribution π^{ref} plays a critical role in the success of the variational optimization when the target distribution π is multi-modal. In particular, we show that using an isotropic Gaussian reference, as commonly proposed in previous works, requires very careful hyperparameter tuning in practice, notably in the selection of the variance of π^{ref}, in order to mitigate mode collapse. Even under favorable tuning, this choice often leads to poor sampling performance after optimization. We attribute this behavior to the weak overlap between π and π^{ref}, a limitation that becomes even more pronounced for challenging multi-modal targets such as those considered in Chapter 3. To address this issue, we leverage our general reference formulation and propose to learn π^{ref} in an auto-supervised manner using the available MCMC samples. By construction, this learned reference distribution is informed by the target and therefore exhibits a significantly improved overlap with π. Moreover, the associated reference scores can be interpreted through the notion of guidance, which plays a central role in diffusion model applications, naturally steering the denoising dynamics toward high-density regions. From this perspective, the variational optimization can be understood as learning a controlled perturbation of the reference denoising process, whose purpose is to reweight particles along their trajectories so as to recover the correct mode proportions of π. Importantly, the complexity of this learning problem remains moderate, as the optimal control is close to zero in both the small- and large-time denoising regimes.

Based on the MCMC-generated *reference* samples, we propose two strategies for constructing γ^{ref} and the associated reference scores, which differ in their expressive power. First, we consider fitting a Gaussian Mixture Model (GMM) to the reference samples, allowing γ^{ref} to be obtained in closed form. This approach has the advantage that the reference scores can be computed analytically from γ^{ref}, resulting in a computationally lightweight strategy. Empirically, we show that this design achieves better performance than most diffusion-based sampling methods evaluated on targets similar to those considered in Chapter 3. Second, for target distributions whose modes deviate significantly from Gaussian shapes, we propose using a multi-level Energy-Based Model (EBM) trained on the reference samples. In this case, the reference log-densities $(\log p_t^{\text{ref}})_{t\in[0,T]}$ are learned using a neural network, providing direct access to both the scores and the reference density. A key contribution of this chapter is the introduction of a novel maximum likelihood training objective for the EBM, defined in an annealed manner over the time interval $[0, T]$ and relying on Replica Exchange as a mechanism for negative sampling. This contribution can be viewed independently of our sampling framework, as a general method for learning energy functions associated with diffusion models from data. Although computationally more demanding in practice, we demonstrate empirically that this approach yields substantially improved overlap with highly complex multi-modal distributions.

We refer to the resulting methodology as LRDS (Learned Reference-based Diffusion Sampling), which consists of the initial sampling phase using standard MCMC, followed by the construction of a reference diffusion process and the final variational optimization stage. While LRDS is more computationally intensive than the MCMC-based SLIPS approach introduced in Chapter 4, it offers the crucial advantage of requiring no target-specific hyperparameter tuning in practice, as it is fully self-supervised, thereby addressing the main limitation of SLIPS. Another important advantage of LRDS is that it completely avoids the use of target-informed neural network parameterizations in variational inference, *i.e.*, architectures explicitly conditioned on the target score $\nabla \log \gamma$. Such parameterizations have recently been shown to constitute a major limitation of previous annealed VI methods, which systematically fail when this information is not available [He et al., 2025b].

Part II - Chapter 6 / (Q5) - Diffusion-based Annealed Boltzmann Generators : benefits, pitfalls and hopes. In this chapter, we investigate the design of a novel class of Boltzmann Generators (BGs) for sampling from a multi-modal target distribution π described by its unnormalized density γ. In their most general form, BGs refer to Monte Carlo correction procedures applied to pre-trained generative models, with the objective of producing unbiased samples from the target distribution, a property that is essential for correct statistical inference. Traditionally, BGs have been associated with importance sampling applied to normalizing flows, a combination that has been shown both theoretically and empirically to scale poorly in high-dimensional settings when the π is multi-modal [Grenioux et al., 2023b].

To address these challenging regimes, we present BGs built upon diffusion models, with a representative example being the LRDS model introduced in Chapter 5, combined with annealed Monte Carlo methodologies such as AIS, SMC, and RE, reviewed in Section 1.3.5. The rationale is to leverage the complementary strengths of DMs, which currently represent the state of the art for generative tasks, and aMC methods, which are specifically designed to handle multi-modal targets for sampling tasks, while exploiting their shared sequential structure. This synergy is further motivated by the fact that diffusion density paths exhibit improved conditioning for multi-modal distributions compared to the classical tempering paths used in aMC, as they mitigate the mode-switching phenomenon [Máté and Fleuret, 2023]. Empirically, we indeed show that naively replacing tempering paths with diffusion paths in AIS, SMC, and RE already leads to improved sampling performance on challenging bi-modal Gaussian mixtures. Although prior works have incorporated denoising diffusion dynamics into AIS [Doucet et al., 2022], SMC [Phillips et al., 2024], or RE [Zhang et al., 2026], this work is, to our knowledge, the first to adopt a unified and systematic perspective on diffusion-based BGs, with the aim of assessing their strengths and limitations in challenging multi-modal settings.

The first main contribution of this chapter is to provide a high-level characterization of strategies for exploiting the correlations between noise levels in DMs in order to integrate them effectively into aMC frameworks. We identify two fundamentally different classes of approaches:

- The first strategy leverages the stochastic transition kernels naturally available in DMs, namely the noising kernels, which are tractable Gaussian distributions, and the denoising kernels, which are in principle intractable but can be approximated by Gaussian distributions. We show that these stochastic kernels can be naturally embedded into aMC schemes. In particular, for AIS and SMC, noising kernels and denoising kernels serve as natural candidates for forward and backward kernels, respectively, thereby modifying both the proposal mechanism and the associated importance weights. For RE, we follow Zhang et al. [2026] and consider a substantial algorithmic modification by replacing the classical deterministic swap move used in the Metropolis-Hastings step with a stochastic variant based on the transition kernels, recovering the original stochastic formulation of MH calibration. Compared to the traditional RE swap move, this stochastic proposal explicitly exploits correlations between consecutive noise levels and is expected to promote improved mixing of the extended Markov chain. Within each aMC variant, we further distinguish two approximation regimes for the denoising kernels: (i) *first-order* approximations, which match only the conditional mean and can be constructed using score estimates learned via score matching, as adopted in previous works [Doucet et al., 2022, Phillips et al., 2024, Zhang et al., 2026]; and (ii) *second-order* approximations, which additionally incorporate covariance information and offer improved accuracy, but are generally impractical due to the need to compute score Jacobians, motivating additional learning strategies [Ou et al., 2025].
- The second strategy, which is novel in the diffusion model literature, departs from the stochastic paradigm and instead exploits deterministic transport maps induced by DMs. This approach replaces the stochastic transition kernels in previously established diffusion-based aMC methods with deterministic counterparts that approximate the probability-flow ODE (1.24), rather than the denoising SDE (1.17). For this strategy to be valid, we identify two key requirements for the approximate transport maps: (a) they must satisfy a mutual invertibility condition between consecutive noise levels, and (b) they must allow for efficient computation of the induced push-forward densities, which appear naturally in AIS and SMC importance weights as well as in RE acceptance ratios. To ensure these properties, we propose constructing the transport maps implicitly, a novelty in the diffusion model framework, using the IM integrator combined with a fixed-point scheme. Exploiting the implicit structure of these maps, we derive training-free approximations of the push-forward densities via power series expansions, applicable in both the first-order regime, relying solely on score estimates, and the second-order regime, requiring additional access to score Jacobians.

The second main contribution of this chapter is an extensive empirical study of these two strategies across all aMC variants, namely AIS, SMC, and RE, on challenging multi-modal targets introduced in Chapter 3. We consider two experimental regimes. The first is an idealized setting in which the diffusion model is assumed to be perfectly learned, allowing us to isolate pure inference effects and to assess both first-order and second-order approximations. The second is a realistic setting in which the diffusion model is trained from biased data and only first-order approximations are available, thereby reflecting practical scenarios. To make this second regime well grounded, we rely on state-of-the-art DM density and score matching techniques, together with two widely used neural architectures proposed in recent literature. Our empirical analysis leads to the following conclusions.

- In the idealized regime, first-order denoising kernels, which are used in prior works, provide only marginal improvements over classical aMC methods based solely on diffusion density paths, revealing an under-

exploitation of inter-level correlations. In contrast, second-order kernels yield clear performance gains, with comparable results across all aMC variants, indicating near-optimal use of the available information. Moreover, we show that deterministic transport-based strategies constitute a promising alternative: our first-order deterministic scheme already matches the performance of second-order stochastic methods, demonstrating that it effectively exploits the information only contained in score estimates, while incurring only modest additional computational cost. In this deterministic, first-order regime, AIS and SMC consistently outperform RE.

- In the practical regime, we obtain the negative result that all diffusion-based BG variants exhibit significant performance degradation, achieving results comparable to those obtained with tempering paths, which are known to be suboptimal. We demonstrate that this degradation are not attributed to inaccuracies in score estimation, but rather to poor density learning, as illustrated by concrete examples. We conjecture that this behavior is caused by mode blindness in the associated training objectives, a phenomenon previously identified for score matching losses [Zhang et al., 2022], which induces mode switching in learned diffusion density paths, sometimes to a greater extent than tempering. These results show that accurate score estimation alone is insufficient to guarantee successful BG construction from diffusion models when the correction step relies on unreliable density approximations. Although these negative findings are obtained on controlled targets such as Gaussian mixtures, we argue that extending the analysis to more complex distributions is unlikely to alter our conclusions.

Part III - Chapter 7 / (Q6) - Tree-based Diffusion Schrödinger Bridge with applications to Wasserstein Barycenters. In this chapter, we investigate the design of a diffusion-based methodology for solving multi-marginal EOT (EmOT) problems in continuous state spaces, thereby opening a research direction that has remained largely unexplored so far. Existing works have either focused on the use of diffusion models in the standard bi-marginal setting, see Section 1.5.2 for a comprehensive overview, or proposed approaches for the multi-marginal setting restricted to discrete state spaces. This chapter aims to bridge this gap by providing both theoretical and computational contributions.

We consider a relaxed instance of the EmOT problem (1.52), previously studied by Haasler et al. [2021], under two main assumptions. First, the multi-state cost function $\bar{c}$ is assumed to be *tree-structured*, taking the form

$$c(x_{0:\ell}) = \sum_{\{v,v'\}\in \mathsf{E}} w_{v,v'} c(x_v, x_{v'}) \;, \tag{1.66}$$

where the structure is inherited from an undirected tree with vertex set $\mathsf{V} = \{0, \dots, \ell\}$, edge set E, and weights $w_{v,v'} > 0$. Here, c denotes a base transport cost and $\{v, v'\}$ represents an edge connecting vertices v and v'. Second, the set of prescribed marginals is assumed to coincide with the leaves of the tree, denoted by S. In order to deploy diffusion model techniques, we further restrict the analysis to the quadratic cost setting, *i.e.*, $c(x, y) = \|x - y\|_2^2$. While this formulation encompasses a wide range of modeling scenarios, it finds a particularly important application in an alternative formulation of the Wasserstein Barycenter (WB) problem. This corresponds to the case of a *star-shaped* tree, in which ℓ leaves are connected to a central node with equal weights. In this setting, each node represents a probability distribution, with the central node to be inferred from the marginals prescribed at the leaves through the dependencies induced by the cost function, and each edge encoding a coupling between two distributions.

A core contribution of this chapter is to generalize, to the multi-marginal case, the equivalence between EOT and static SB formulations, previously established in the classical bi-marginal setting. We show that the EmOT problem with the tree-structured quadratic cost $\bar{c}$ is indeed equivalent to a tree-based static SB problem of the form

$$\underset{\bar{\pi}_i = \mu_i \;, \forall i \in \mathsf{S}}{\arg\min} \; \mathrm{KL}(\bar{\pi} \mid \bar{\pi}^0) \;, \tag{1.67}$$

where the reference measure $\bar{\pi}^0$ admits a Markovian structure with Gaussian transition kernels, induced by a directed version of the underlying tree. This equivalence naturally leads us to propose solving the EmOT problem via a multi-marginal extension of the IPF algorithm, denoted mIPF. This algorithm generates a sequence of distributions $\{\bar{\pi}^n\}_{n\in\mathbb{N}}$, initialized at $\bar{\pi}^0$, and recursively defined through iterative KL projections onto the marginal constraints prescribed at the leaves. From a theoretical perspective, we prove that this sequence converges to the solution of the tree-based EmOT problem. While convergence results were previously available for bounded cost functions and compactly supported marginals [Marino and Gerolin, 2020, Carlier, 2022], we establish a more general result that includes the quadratic cost case and marginals with full support. To the

best of our knowledge, this work provides the first non-quantitative convergence guarantees for mIPF in a non-compact setting under mild assumptions. Extending results from Nutz [2021], Ruschendorf [1995] to the multi-marginal context, we notably prove that the EmOT solution is unique and admits a representation in terms of $|\mathsf{S}|$ Schrödinger potentials, analogous to (1.51). Moreover, we show that the mIPF iterates admit the same potential-based representation and that these potentials converge to the Schrödinger potentials of the solution.

Although the mIPF iterates admit closed-form expressions, they are not tractable in continuous state spaces, similarly to the bi-marginal case. This motivates the second main contribution of this chapter, in which we derive a dynamical formulation of the mIPF scheme that yields tractable updates of path measures defined over the edges of the full tree. As in the bi-marginal setting, consecutive dynamical iterates are obtained through the application of time-reversal operators, here operating independently on the edges of the tree. More precisely, for an update enforcing the marginal constraint μ_k, following a previous update on $\mu_{k'}$, we show that time reversals occur only along the edges of the unique path connecting k' to k, while the path measures on the remaining edges remain unchanged. Leveraging the Markovian structure of $\bar{\pi}^0$, we prove that each dynamical iterate actually corresponds to a collection of diffusion processes, initialized at a leaf, taken as the root of the tree, and propagated recursively along the edges toward the other leaves. The time horizon associated with each edge $\{v, v'\}$ is given by $T_{v,v'} = \varepsilon/(2w_{v,v'})$.

As a third main contribution, we propose a diffusion-based computational implementation of the resulting dynamical mIPF scheme, drawing strong inspiration from the DSB method [Bortoli et al., 2021] developed for the bi-marginal case. For each edge of the tree, we introduce two neural networks to learn the drifts of the stochastic dynamics along that edge, respectively for forward and backward direction. Similarly to Bortoli et al. [2021], we employ the Mean-Matching loss to iteratively learn the time-reversal dynamics, while storing simulated trajectories for reuse in subsequent iterations. We refer to the resulting numerical method as TreeDSB (Tree-based Diffusion Schrödinger Bridge). Although the proposed framework is, in principle, applicable to arbitrary undirected trees, our computational study focuses on the star-shaped case, where $\mathsf{S} = \{1, \dots, \ell\}$. In this setting, we show that the corresponding tree-based EmOT problem is equivalent to a doubly regularized Wasserstein Barycenter (WB) problem [Chizat, 2025] of the form

$$\arg\min_{\nu} \frac{1}{\ell} \sum_{i=1}^{\ell} W^2_{2,\ell\varepsilon}(\nu, \mu_i) + (\ell - 1)\varepsilon \mathrm{H}(\nu) \,. \tag{1.68}$$

Unlike the classical entropy-regularized WB formulation (1.53), the objective in (1.68) features an additional negative-entropy regularization term. Importantly, this term acts antagonistically to the regularization induced by $W^2_{2,\ell\varepsilon}$, thereby reducing the tendency of the barycenter to exhibit overly diffuse support and, in principle, bringing it closer to the unregularized WB solution. In practice, this doubly regularized formulation plays a crucial role in resolving the fundamental trade-off between accuracy and numerical stability, that one usually faces in EOT-based methodologies. In particular, it allows the regularization parameter ε in TreeDSB to be chosen sufficiently large to ensure stable training, without introducing excessive bias. This stands in contrast to standard entropy-regularized WB approaches, where overly small values of ε can severely degrade performance. Finally, we illustrate the performance of TreeDSB on both synthetic and real-world datasets, demonstrating its robustness to dimensionality, its ability to converge to accurate WB solutions or plausible samples thereof, and its capacity to recover high-quality samples from the leaf distributions through the learned diffusion models, a desirable property from a generative modeling perspective.

Part III - Chapter 8 / (Q7) - Twisted Schrödinger Bridge Matching : a unified continuous- and discrete-time framework. In this final chapter, we focus on the generalized formulation of the dynamical Schrödinger Bridge problem defined by (1.62). This problem aims at minimizing over generated trajectories both the standard quadratic transport cost and a time-dependent potential function V_t, while enforcing prescribed marginal constraints at the endpoints of the considered time interval. Extending the classical SB problem, which is primarily used for unpaired data translation tasks, the formulation (1.62) finds its main application in the inference of population dynamics, also known as trajectory inference. In this context, the potential V_t may encode either repulsive regions, such as physical obstacles, or attractive regions, for instance partially observed data that one seeks to align with. As this research direction is relatively recent, large-scale methodologies for solving (1.62) remain largely unexplored. In parallel, recent progress on solving the classical SB problem via iterative bridge matching methods, such as DSBM [Shi et al., 2023], which implements the Iterative Markovian Fitting (IMF) scheme, has demonstrated strong empirical performance and has motivated further investigation in this direction.

Within this context, Liu et al. [2024] recently proposed GSBM, a computational method designed to solve (1.62) by iteratively alternating between two steps analogous to those of DSBM: (a) learning a V_t-informed stochastic interpolant process using Gaussian parameterization and couplings inherited from the previous iteration, and (b) learning a stochastic dynamics with the same intermediate marginals as the interpolant obtained in step (a). Despite its strong empirical performance across a variety of tasks, GSBM presents two notable limitations. First, its theoretical update rules are formulated in continuous time, while the implementation operates in discrete time, potentially introducing bias through this mismatch. Second, although steps (a) and (b) resemble Reciprocal and Markovian projections inherited from the IMF scheme, their connection to the exact IMF updates is not rigorously established, again potentially inducing bias. In this chapter, we address these limitations by proposing a novel methodology designed as a faithful translation of the IMF scheme, applicable in both continuous and discrete time settings.

As a first main contribution, we introduce an alternative bridge matching formulation to GSBM for solving (1.62), keeping the original continuous-time setting. To this end, we rigorously reformulate this problem into the variational formulation (1.63), which we coin *Twisted Schrödinger Bridge* (TSB) problem, and propose an iterative scheme to solve it that is more closely aligned with the theoretical IMF procedure than GSBM. In particular, we establish a direct analogue of the Markovian projection objective, whose resulting loss introduces a novel component compared to GSBM, namely the explicit appearance of the gradient of the penalty ∇V. This feature enables direct exploitation of the potential V_t during optimization, which is not possible in Liu et al. [2024]. Moreover, when $V_t \equiv 0$, the DSBM objective is recovered exactly as a special case of our formulation, without any approximation, a property that does not hold for the GSBM objective. As already observed by Liu et al. [2024], the interpolant process associated with the Reciprocal projection is not tractable, in contrast to the classical DSBM setting. To address this issue, we also rely on Gaussian parameterization, but we propose a modified loss compared to Liu et al. [2024]. Specifically, we derive a variational and tractable objective whose minimizer is exactly the target interpolant process, a connection that is not made explicit in Liu et al. [2024]. We refer to the resulting algorithm as TSBM (Twisted Schrödinger Bridge Matching).

As a second main contribution, we develop a rigorous formulation of the TSB problem in discrete time. This formulation arises as a translation of the marginally constrained variational problem (1.63) from path space to the space of joint distributions, *i.e.*, stochastic processes defined over a finite number N of time steps. Our approach relies on a *twisted* Feynman–Kac reference measure [Heng et al., 2020]. To date, such a discrete-time formulation had only been considered by Gushchin et al. [2024] for the classical SB problem (1.58), corresponding to the case $V_t \equiv 0$. Extending the results of Gushchin et al. [2024], we derive an ideal iterative scheme to solve the resulting discrete-time TSB problem. In the spirit of the IMF scheme, this algorithm alternates between Markovian and Reciprocal projections, which can be rigorously defined in discrete time as KL projections on joint distributions. As in the continuous-time IMF setting, we show that the Reciprocal projection admits a tractable closed-form expression, while the Markovian projection is characterized as the minimizer of an explicit and tractable objective.

Finally, as a third main contribution, we translate this ideal discrete-time scheme into a fully practical algorithm, namely the discrete-time version of TSBM. For the Markovian projection, we restrict the projection to a class of joint distributions parameterized by neural networks, for which we derive a discrete-time bridge matching objective that can be efficiently implemented and optimized. In particular, this can be seen as the twisted version of the DDPM objective [Ho et al., 2020], applicable for any value of N, explicitly incorporating the discrete-time counterpart of the potential function V_t. For the reciprocal projection, the main difficulty lies in computing the discrete-time bridge measure required to construct the interpolant process, mirroring the challenges encountered in continuous time. We address this issue by reformulating the problem as a sampling task, whose goal is to generate samples from the bridge distribution rather than to approximate it via parameterization. To this end, we propose a learning-free approach, based solely on Monte Carlo techniques, that follows an original two-step procedure: first, samples are drawn from the bridge restricted to a subset of *control* time steps using a Gibbs-within-Langevin scheme; second, samples at the remaining time steps are generated conditionally on the control points, exploiting the Markovian structure of the bridge. In contrast to Liu et al. [2024], our approach introduces no approximation error at this stage, thereby avoiding additional bias.

Bibliography

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J. Ballard, Joshua Bambrick, Sebastian W. Bodenstein, David A. Evans, Chia-Chun Hung, Michael O'Neill, David Reiman, Kathryn Tunyasuvunakool, Zachary Wu, Akvilė Žemgulytė, Eirini Arvaniti, Charles Beattie, Ottavia Bertolli, Alex Bridgland, Alexey Cherepanov, Miles Congreve, Alexander I. Cowen-Rivers, Andrew Cowie, Michael Figurnov, Fabian B. Fuchs, Hannah Gladman, Rishub Jain, Yousuf A. Khan, Caroline M. R. Low, Kuba Perlin, Anna Potapenko, Pascal Savy, Sukhdeep Singh, Adrian Stecula, Ashok Thillaisundaram, Catherine Tong, Sergei Yakneen, Ellen D. Zhong, Michal Zielinski, Augustin Žídek, Victor Bapst, Pushmeet Kohli, Max Jaderberg, Demis Hassabis, and John M. Jumper. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016):493–500, 2024. 16, 115

Beatrice Acciaio, Julio Backhoff-Veraguas, and René Carmona. Extended mean field control problems: stochastic maximum principle and transport perspective. *SIAM journal on Control and Optimization*, 57(6):3666–3693, 2019. 137

S. Agapiou, O. Papaspiliopoulos, D. Sanz-Alonso, and A. M. Stuart. Importance sampling: Intrinsic dimension and computational cost. *Statistical Science*, 32(3):405–431, 2017. 7, 84, 115, 119

Abhinav Agrawal, Daniel R Sheldon, and Justin Domke. Advances in black-box vi: Normalizing flows, importance weighting, and optimization. In *Advances in Neural Information Processing Systems*, pages 17358–17369, 2020. 8

Martial Agueh and Guillaume Carlier. Barycenters in the Wasserstein space. *SIAM Journal on Mathematical Analysis*, 43(2):904–924, 2011. 27, 137, 144

Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition workshops*, pages 126–135, 2017. 368

Tara Akhound-Sadegh, Jarrid Rector-Brooks, Joey Bose, Sarthak Mittal, Pablo Lemos, Cheng-Hao Liu, Marcin Sendera, Siamak Ravanbakhsh, Gauthier Gidel, Yoshua Bengio, Nikolay Malkin, and Alexander Tong. Iterated denoising energy matching for sampling from boltzmann densities. In *International Conference on Machine Learning*, pages 760–786. PMLR, 2024. 24, 94, 104, 110, 134, 257

Tara Akhound-Sadegh, Jungyoon Lee, Joey Bose, Valentin De Bortoli, Arnaud Doucet, Michael M. Bronstein, Dominique Beaini, Siamak Ravanbakhsh, Kirill Neklyudov, and Alexander Tong. Progressive inference-time annealing of diffusion models for sampling from boltzmann densities. In *Advances in Neural Information Processing Systems*, 2025. 25, 132

Ahmed El Alaoui, Andrea Montanari, and Mark Sellke. Sampling from the sherrington-kirkpatrick gibbs measure via algorithmic stochastic localization. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 323–334, 2022. 91, 92

M. S. Albergo, G. Kanwar, and P. E. Shanahan. Flow-based generative models for markov chain monte carlo in lattice field theory. *Physical Review D*, 100(3):034515, 2019. 115

Michael Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. *arXiv preprint arXiv:2303.08797*, 2023. 16, 19, 86, 95, 147, 150, 152, 349, 350, 360

Michael Samuel Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *International Conference on Learning Representations*, 2023. 18

Michael Samuel Albergo and Eric Vanden-Eijnden. NETS: A non-equilibrium transport sampler. In *International Conference on Machine Learning*. PMLR, 2025. 23

Michael Samuel Albergo, Mark Goldstein, Nicholas Matthew Boffi, Rajesh Ranganath, and Eric Vanden-Eijnden. Stochastic interpolants with data-dependent couplings. In *International Conference on Machine Learning*, pages 921–937. PMLR, 2024. 20

Pedro C Álvarez-Esteban, E Del Barrio, JA Cuesta-Albertos, and C Matrán. A fixed-point approach to barycenters in Wasserstein space. *Journal of Mathematical Analysis and Applications*, 441(2):744–762, 2016. 137, 144

Brian DO Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12 (3):313–326, 1982. 17, 117

Christophe Andrieu and Gareth O. Roberts. The pseudo-marginal approach for efficient Monte Carlo computations. *The Annals of Statistics*, 37(2):697 – 725, 2009. 126

Michael Arbel, Alex Matthews, and Arnaud Doucet. Annealed flow transport monte carlo. In *International Conference on Machine Learning*, pages 318–330. PMLR, 2021. 104, 128

Oleg Arenz, Philipp Dahlinger, Zihan Ye, Michael Volpp, and Gerhard Neumann. A unified perspective on natural gradient variational inference with gaussian mixture models. *Transactions on Machine Learning Research*, 2023. 104

Yves F. Atchadé, Gareth O. Roberts, and Jeffrey S. Rosenthal. Towards optimal scaling of metropolis-coupled markov chain monte carlo. *Statistics and Computing*, 21(4):555–568, 2011. 86

Haim Avron and Sivan Toledo. Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix. *J. ACM*, 58(2), 2011. 127

Fan Bao, Chongxuan Li, Jiacheng Sun, Jun Zhu, and Bo Zhang. Estimating the optimal covariance with imperfect mean in diffusion probabilistic models. In *International Conference on Machine Learning*, pages 1555–1584. PMLR, 2022a. 119

Fan Bao, Chongxuan Li, Jun Zhu, and Bo Zhang. Analytic-DPM: an analytic estimate of the optimal reverse variance in diffusion probabilistic models. In *International Conference on Learning Representations*, 2022b. 119

Erhan Bayraktar, Alexander MG Cox, and Yavor Stoev. Martingale optimal transport with stopping. *SIAM Journal on Control and Optimization*, 56(1):417–433, 2018. 137

Jens Behrmann, Will Grathwohl, Ricky T. Q. Chen, David Duvenaud, and Joern-Henrik Jacobsen. Invertible residual networks. In *International Conference on Machine Learning*, pages 573–582. PMLR, 2019. 127, 270

Jean-David Benamou and Yann Brenier. A computational fluid mechanics solution to the monge-kantorovich mass transfer problem. *Numerische Mathematik*, 84(3):375–393, 2000. 30, 147, 150

Jean-David Benamou, Guillaume Carlier, Marco Cuturi, Luca Nenna, and Gabriel Peyré. Iterative Bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, 37(2):A1111–A1138, 2015. 28, 29, 137, 144

Julius Berner, Lorenz Richter, and Karen Ullrich. An optimal control perspective on diffusion-based generative modeling. *Transactions on Machine Learning Research*, 2023. 22, 86, 91, 99, 104, 110, 202, 236, 241

Julius Berner, Lorenz Richter, Marcin Sendera, Jarrid Rector-Brooks, and Nikolay Malkin. From discrete-time policies to continuous-time diffusion samplers: Asymptotic equivalences and faster training. *Transactions on Machine Learning Research*, 2026. 22

Espen Bernton, Jeremy Heng, Arnaud Doucet, and Pierre E. Jacob. Schrödinger bridge samplers. *arXiv preprint arXiv:1912.13170*, 2019a. 25

Espen Bernton, Pierre E Jacob, Mathieu Gerber, and Christian P Robert. On parameter estimation with the wasserstein distance. *Information and Inference: A Journal of the IMA*, 8(4):657–676, 2019b. 31

Michael Betancourt. A general metric for Riemannian manifold Hamiltonian Monte Carlo. In *International Conference on Geometric Science of Information*, pages 327–334. Springer, 2013. 74

Michael Betancourt. A conceptual introduction to Hamiltonian Monte Carlo. *arXiv preprint arXiv:1701.02434*, 2017. 71

David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe and. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017. 7

Denis Blessing, Xiaogang Jia, Johannes Esslinger, Francisco Vargas, and Gerhard Neumann. Beyond ELBOs: A large-scale evaluation of variational methods for sampling. In *International Conference on Machine Learning*, pages 4205–4229. PMLR, 2024. 81, 85, 104, 110, 229, 251, 252, 253, 256, 257

Denis Blessing, Julius Berner, Lorenz Richter, and Gerhard Neumann. Underdamped diffusion bridges with applications to sampling. In *International Conference on Learning Representations*, 2025a. 22

Denis Blessing, Xiaogang Jia, and Gerhard Neumann. End-to-end learning of gaussian mixture priors for diffusion sampler. In *International Conference on Learning Representations*, 2025b. 22

Nicholas Matthew Boffi, Michael Samuel Albergo, and Eric Vanden-Eijnden. How to build a consistency model: Learning flow maps via self-distillation. In *Advances in Neural Information Processing Systems*, 2025a. 358, 359, 360, 361, 362, 365, 366, 375

Nicholas Matthew Boffi, Michael Samuel Albergo, and Eric Vanden-Eijnden. Flow map matching with stochastic interpolants: A mathematical framework for consistency models. *Transactions on Machine Learning Research*, 2025b. 358, 360, 361

Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. Sliced and radon wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51:22–45, 2015. 116, 212, 225

Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. *Advances in Neural Information Processing Systems*, pages 17695–17709, 2021. 31, 41, 91, 137, 139, 141, 144, 146, 147, 148, 150, 333, 335, 336

Valentin De Bortoli, Guan-Horng Liu, Tianrong Chen, Evangelos A. Theodorou, and Weilie Nie. Augmented bridge matching. *arXiv preprint arXiv:2311.06978*, 2023. 20

Valentin De Bortoli, Michael Hutchinson, Peter Wirnsberger, and Arnaud Doucet. Target score matching. *arXiv preprint arXiv:2402.08667*, 2024a. 24, 94, 110, 119, 130, 271, 272

Valentin De Bortoli, Iryna Korshunova, Andriy Mnih, and Arnaud Doucet. Schrodinger bridge flow for unpaired data translation. In *Advances in Neural Information Processing Systems*, 2024b. 32, 148

Nawaf Bou-Rabee, Andreas Eberle, and Raphael Zimmer. Coupling and convergence for Hamiltonian Monte Carlo. *The Annals of Applied Probability*, 30(3):1209 – 1250, 2020. 82

Lev M Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR computational mathematics and mathematical physics*, 7(3):200–217, 1967. 29, 149

Yann Brenier. Décomposition polaire et réarrangement monotone des champs de vecteurs. *CR Acad. Sci. Paris Sér. I Math.*, 305:805–808, 1987. 27

James Brofos and Roy R Lederman. Evaluating the implicit midpoint integrator for Riemannian Hamiltonian Monte Carlo. In *International Conference on Machine Learning*, pages 1072–1081. PMLR, 2021a. 74, 75, 174

James Brofos, Marylou Gabrie, Marcus A. Brubaker, and Roy R. Lederman. Adaptation of the independent metropolis-hastings sampler with normalizing flow proposals. In *International Conference on Artificial Intelligence and Statistics*, pages 5949–5986. PMLR, 2022. 115

James A Brofos and Roy R Lederman. On numerical considerations for Riemannian Manifold Hamiltonian Monte Carlo. *arXiv preprint arXiv:2111.09995*, 2021b. 75, 174

Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng. *Handbook of Markov Chain Monte Carlo*. Chapman and Hall/CRC, May 2011. 11, 82, 259

Marcus Brubaker, Mathieu Salzmann, and Raquel Urtasun. A family of MCMC methods on implicitly defined manifolds. In *Artificial Intelligence and Statistics*, pages 161–172. PMLR, 2012. 71, 78, 79

Gerard Brunick and Steven Shreve. Mimicking an itô process by a solution of a stochastic differential equation. *The Annals of Applied Probability*, 23(4):1584–1628, Aug 2013. 92, 208

S. Bubeck, R Eldan, and J. Lehec. Finite-time analysis of projected Langevin Monte Carlo. In *Advances in Neural Information Processing Systems*, pages 1243–1251, 2015. 71

Monica F. Bugallo, Victor Elvira, Luca Martino, David Luengo, Joaquin Miguez, and Petar M. Djuric. Adaptive importance sampling: The past, the present, and the future. *IEEE Signal Processing Magazine*, 34(4):60–79, 2017. 7

Charlotte Bunne, Laetitia Papaxanthos, Andreas Krause, and Marco Cuturi. Proximal optimal transport modeling of population dynamics. In *International Conference on Artificial Intelligence and Statistics*, pages 6511–6528. PMLR, 2022. 137

Rainer Burkard, Mauro Dell'Amico, and Silvano Martello. *Assignment problems: revised reprint*. SIAM, 2012. 28

Simon Byrne and Mark Girolami. Geodesic Monte Carlo on embedded manifolds. *Scandinavian Journal of Statistics*, 40(4):825–845, 2013. 78

Alberto Cabezas, Louis Sharrock, and Christopher Nemeth. Markovian flow matching: Accelerating mcmc with continuous normalizing flows. In *Advances in Neural Information Processing Systems*, pages 104383–104411, 2024. 115

Jianrui Cai, Hui Zeng, Hongwei Yong, Zisheng Cao, and Lei Zhang. Toward real-world single image super-resolution: A new benchmark and a new model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3086–3095, 2019. 368

Gabriel Cardoso, Yazid Janati el idrissi, Sylvain Le Corff, and Eric Moulines. Monte carlo guided denoising diffusion models for bayesian linear inverse problems. In *International Conference on Learning Representations*, 2024. 128

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proceedings of the European conference on computer vision (ECCV) workshops*, pages 213–229. Springer, 2020. 137

Guillaume Carlier. On the linear convergence of the multimarginal Sinkhorn algorithm. *SIAM Journal on Optimization*, 32(2):786–794, 2022. 40, 142, 143

Guillaume Carlier and Ivar Ekeland. Matching for teams. *Economic theory*, 42(2):397–418, 2010. 27

Patrick Cattiaux, Giovanni Conforti, Ivan Gentil, and Christian Léonard. Time reversal of diffusion processes under a finite entropy condition. *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, 59(4):1844 – 1881, 2023. 105, 139, 221, 237, 258

Michele Ceriotti, Guy A. R. Brain, Oliver Riordan, and David E. Manolopoulos. The inefficiency of re-weighted sampling and the curse of system size in high-order path integration. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 468(2137):2–17, 2012. 96

Clément Chadebec, Onur Tasar, Eyal Benaroche, and Benjamin Aubin. Flash diffusion: Accelerating any conditional diffusion model for few steps image generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 15686–15695, 2025a. 358, 368

Clément Chadebec, Onur Tasar, Sanjeev Sreetharan, and Benjamin Aubin. Lbm: Latent bridge matching for fast image-to-image translation. In *International Conference on Computer Vision (ICCV)*, pages 29086–29098, 2025b. 368, 377, 378

Sourav Chatterjee and Persi Diaconis. The sample size required in importance sampling. *The Annals of Applied Probability*, 28(2):1099–1135, 2018. 7, 96

Jannis Chemseddine, Christian Wald, Richard Duong, and Gabriele Steidl. Neural sampling from boltzmann densities: Fisher-rao curves in the wasserstein geometry. In *International Conference on Learning Representations*, 2025. 23

Junhua Chen, Lorenz Richter, Julius Berner, Denis Blessing, Gerhard Neumann, and Anima Anandkumar. Sequential controlled langevin diffusions. In *International Conference on Learning Representations*, 2025. 22, 86, 102

Nanxin Chen, Yu Zhang, Heiga Zen, Ron J Weiss, Mohammad Norouzi, and William Chan. WaveGrad: Estimating gradients for waveform generation. In *International Conference on Learning Representations*, 2021a. 91

Ricky T. Q. Chen, Jens Behrmann, David K Duvenaud, and Joern-Henrik Jacobsen. Residual flows for invertible generative modeling. In *Advances in Neural Information Processing Systems*, 2019. 127

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in Neural Information Processing Systems*, 2018a. 115

Sitan Chen, Sinho Chewi, Jerry Li, Yuanzhi Li, Adil Salim, and Anru Zhang. Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions. In *International Conference on Learning Representations*, 2023a. 91

Tianrong Chen, Guan-Horng Liu, and Evangelos A Theodorou. Likelihood training of Schrödinger bridge using forward-backward SDEs theory. *International Conference on Learning Representations*, 2022. 137, 144

Tianrong Chen, Guan-Horng Liu, Molei Tao, and Evangelos Theodorou. Deep momentum multi-marginal schrödinger bridge. *Advances in Neural Information Processing Systems*, pages 57058–57086, 2023b. 144, 148

Wenlin Chen, Mingtian Zhang, Brooks Paige, José Miguel Hernández-Lobato, and David Barber. Diffusive Gibbs sampling. In *International Conference on Machine Learning*, pages 7731–7747. PMLR, 2024. 25

Yifan Chen and Eric Vanden-Eijnden. Variational optimality of föllmer processes in generative diffusions. *arXiv preprint arXiv:2602.10989*, 2026. 99

Yongxin Chen, Tryphon Georgiou, Michele Pavon, and Allen Tannenbaum. Robust transport over networks. *IEEE Transactions on Automatic Control*, 62(9):4675–4682, 2016. 139

Yongxin Chen, Giovanni Conforti, and Tryphon T Georgiou. Measure-valued spline curves: An optimal transport viewpoint. *SIAM Journal on Mathematical Analysis*, 50(6):5947–5968, 2018b. 144

Yongxin Chen, Tryphon T Georgiou, and Michele Pavon. Optimal transport in systems and control. *Annual Review of Control, Robotics, and Autonomous Systems*, 4, 2021b. 143

Yuansi Chen and Ronen Eldan. Localization schemes: A framework for proving mixing bounds for markov chains. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 110–122. IEEE, 2022. 91

Xiang Cheng, Niladri S Chatterji, Peter L Bartlett, and Michael I Jordan. Underdamped langevin mcmc: A non-asymptotic analysis. In *Conference on Learning Theory*, pages 300–323. PMLR, 2018. 11

Raphaël Chetrite and Hugo Touchette. Nonequilibrium markov processes conditioned on large deviations. In *Annales Henri Poincaré*, volume 16, pages 2005–2057. Springer, 2015. 151

Christophe Chipot. Free Energy Methods for the Description of Molecular Processes. *Annual Review of Biophysics*, 52:113–138, 2023. 82

Lénaïc Chizat. Doubly regularized entropic wasserstein barycenter. *Foundations of Computational Mathematics*, pages 1–38, 2025. 41, 143

Nicolas Chopin, Omiros Papaspiliopoulos, et al. *An introduction to Sequential Monte Carlo*, volume 4. Springer, 2020. 15, 198, 201, 284

Sebastian Claici, Edward Chien, and Justin Solomon. Stochastic Wasserstein barycenters. In *International Conference on Machine Learning*, pages 999–1008. PMLR, 2018. 144

Adam D Cobb, Atılım Güneş Baydin, Andrew Markham, and Stephen J Roberts. Introducing an explicit symplectic integration scheme for Riemannian Manifold Hamiltonian Monte Carlo. *arXiv preprint arXiv:1910.06243*, 2019. 74, 174

Samuel Cohen, Michael Arbel, and Marc Peter Deisenroth. Estimating barycenters of measures in high dimensions. *arXiv preprint arXiv:2007.07105*, 2020. 144

Giovanni Conforti, Alain Durmus, and Giacomo Greco. Quantitative contraction rates for sinkhorn algorithm: beyond bounded costs and compact marginals. *arXiv preprint arXiv:2304.04451*, 2023. 29

Codina Cotar, Gero Friesecke, and Claudia Klüppelberg. Density functional theory and optimal transportation with Coulomb cost. *Communications on Pure and Applied Mathematics*, 66(4):548–599, 2013. 137

Ben Cousins and Santosh Vempala. Gaussian cooling and o(n^3) algorithms for volume and gaussian volume. *SIAM Journal on Computing*, 47(3):1237–1273, 2018. 71

Imre Csiszár. I-divergence geometry of probability distributions and minimization problems. *The Annals of Probability*, pages 146–158, 1975. 322, 324, 328

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, 2013. 29, 137, 232, 332

Marco Cuturi and Arnaud Doucet. Fast computation of Wasserstein barycenters. In *International Conference on Machine Learning*, pages 685–693. PMLR, 2014. 31, 137, 139, 144, 145, 146, 336, 340

Yin Dai, Yuan Gao, Jian Huang, Yuling Jiao, Lican Kang, and Jin Liu. Lipschitz transport maps via the Föllmer flow. *arXiv preprint arXiv:2309.03490*, 2023. 234

Paolo Dai Pra. A stochastic control approach to reciprocal diffusion processes. *Applied Mathematics and Optimization*, 23(1):313–329, 1991. 137, 143

Arnak S Dalalyan. Theoretical guarantees for approximate sampling from smooth and log-concave densities. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 79(3):651–676, 2017. 11, 71, 82, 91, 98, 103

Kamélia Daudel, Joe Benton, Yuyang Shi, and Arnaud Doucet. Alpha-divergence variational inference meets importance weighted auto-encoders: Methodology and asymptotics. *Journal of Machine Learning Research*, 24(243):1–83, 2023. 9

Katherine Deck and Tobias Bischoff. Easing color shifts in score-based diffusion models. *arXiv preprint arXiv:2306.15832*, 2023. 375

Luigi Del Debbio, Joe Marsh Rossney, and Michael Wilson. Machine Learning Trivializing Maps: A First Step Towards Understanding How Flow-Based Samplers Scale Up. *PoS*, LATTICE2021:059, 2022. 115

Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. Sequential monte carlo samplers. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 68(3):411–436, 2006. 14, 85, 91, 103, 110, 121, 245

Mauricio Delbracio and Peyman Milanfar. Inversion by direct iteration: An alternative to denoising diffusion for image restoration. *Transactions on Machine Learning Research*, 2023. 358, 359

W Edwards Deming and Frederick F Stephan. On a least squares adjustment of a sampled frequency table when the expected marginal totals are known. *The Annals of Mathematical Statistics*, 11(4):427–444, 1940. 29

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society: series B (methodological)*, 39(1):1–22, 1977. 108

Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, 2021. 105

Keyan Ding, Kede Ma, Shiqi Wang, and Eero P Simoncelli. Image quality assessment: Unifying structure and texture similarity. *IEEE transactions on pattern analysis and machine intelligence*, 44(5):2567–2581, 2020. 368

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In *International Conference on Learning Representations*, 2017. 9, 200, 201

Carles Domingo-Enrich, Yuanqi Du, and Michael S. Albergo. A unified perspective on fine-tuning and sampling with diffusion and flow models. *arXiv preprint arXiv:2605.00229*, 2026. 110

Carles Domingo i Enrich, Jiequn Han, Brandon Amos, Joan Bruna, and Ricky TQ Chen. Stochastic optimal control matching. *Advances in Neural Information Processing Systems*, pages 112459–112504, 2024. 348

Zehao Dou and Yang Song. Diffusion posterior sampling for linear inverse problem solving: A filtering perspective. In *International Conference on Learning Representations*, 2024. 128

Randal Douc, Eric Moulines, Pierre Priouret, and Philippe Soulier. *Markov Chains*. Springer, 2018. 163

Arnaud Doucet, Nando De Freitas, Neil James Gordon, et al. *Sequential Monte Carlo methods in practice*, volume 1. Springer, 2001. 121

Arnaud Doucet, Will Grathwohl, Alexander G Matthews, and Heiko Strathmann. Score-based diffusion meets annealed importance sampling. *Advances in Neural Information Processing Systems*, pages 21482–21494, 2022. 22, 39, 99, 104

Yilun Du, Conor Durkan, Robin Strudel, Joshua B Tenenbaum, Sander Dieleman, Rob Fergus, Jascha Sohl-Dickstein, Arnaud Doucet, and Will Sussman Grathwohl. Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and mcmc. In *International Conference on Machine Learning*, pages 8489–8510. PMLR, 2023. 119

Yuanqi Du, Michael Plainer, Rob Brekelmans, Chenru Duan, Frank Noe, Carla P Gomes, Alan Aspuru-Guzik, and Kirill Neklyudov. Doob's lagrangian: A sample-efficient variational approach to transition path sampling. *Advances in Neural Information Processing Systems*, pages 65791–65822, 2024. 152, 351

Yuanqi Du, Jiajun He, Francisco Vargas, Yuanqing Wang, Carla P Gomes, José Miguel Hernández-Lobato, and Eric Vanden-Eijnden. FEAT: Free energy estimators with adaptive transport. In *Advances in Neural Information Processing Systems*, 2025. 130

Simon Duane, A.D. Kennedy, Brian J. Pendleton, and Duncan Roweth. Hybrid Monte Carlo. *Physics Letters B*, 195(2):216–222, 1987. 11, 71, 78, 82, 120, 259

Paul Dupuis and Richard S Ellis. *A weak convergence approach to the theory of large deviations*. John Wiley & Sons, 2011. 328

Alain Durmus and Éric Moulines. Quantitative bounds of convergence for geometrically ergodic markov chain in the wasserstein distance with application to the metropolis adjusted langevin algorithm. *Statistics and Computing*, 25:5–19, 2015. 11, 18, 106, 210, 212, 231

Alain Durmus and Eric Moulines. Nonasymptotic convergence analysis for the unadjusted Langevin algorithm. *The Annals of Applied Probability*, 27(3):1551–1587, 2017. 11, 71, 91, 98, 103

Alain Durmus and Éric Moulines. High-dimensional Bayesian inference via the unadjusted Langevin algorithm. *Bernoulli*, 25(4A):2854–2882, 2019. 82

Alain Durmus, Maxence Noble, and Thibaut Pellerin. Sampling from multi-modal distributions on riemannian manifolds with training-free stochastic interpolants. *arXiv preprint arXiv:2602.00641*, 2026. 33

Raaz Dwivedi, Yuansi Chen, Martin J Wainwright, and Bin Yu. Log-concave sampling: Metropolis-Hastings algorithms are fast! In *Conference On Learning Theory*, pages 793–797. PMLR, 2018. 82, 96

Martin Dyer and Alan Frieze. Computing the volume of convex bodies: a case where randomness provably helps. *Probabilistic combinatorics and its applications*, 44:123–170, 1991. 71

Marvin Eisenberger, Aysim Toker, Laura Leal-Taixé, and Daniel Cremers. Deep shells: Unsupervised shape correspondence with optimal transport. *Advances in Neural Information Processing Systems*, pages 10491–10502, 2020. 137

Ronen Eldan. Thin shell implies spectral gap up to polylog via a stochastic localization scheme. *Geometric and Functional Analysis*, 23(2):532–569, 2013. 91

Ronen Eldan. Taming correlations through entropy-efficient measure decompositions with applications to mean-field approximation. *Probability Theory and Related Fields*, 176(3-4):737–755, 2020. 91

Ronen Eldan. Analysis of high-dimensional distributions using pathwise methods. In *Proc. Int. Cong. Math*, volume 6, pages 4246–4270, 2022. 91

Víctor Elvira and Luca Martino. Advances in importance sampling. *arXiv preprint arXiv:2102.05407*, 2021. 7

Paula Cordero Encinar, Andrew B. Duncan, Sebastian Reich, and O. Deniz Akyildiz. Sampling by averaging: A multiscale approach to score estimation. In *Advances in Neural Information Processing Systems*, 2025. 25

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning*, pages 12606–12633. PMLR, 2024. 20, 358

Lawrence C. Evans. *Partial Differential Equations*. American Mathematical Society, 2 edition, 2010. 19

Jiaojiao Fan, Amirhossein Taghvaei, and Yongxin Chen. Scalable computations of wasserstein barycenter via input convex neural networks. In *International Conference on Machine Learning*, pages 1571–1581. PMLR, 2021. 144, 145, 333, 336, 341

Mingzhou Fan, Ruida Zhou, Chao Tian, and Xiaoning Qian. Path-guided particle-based sampling. In *International Conference on Machine Learning*, pages 12916–12934. PMLR, 2024. 23

Kilian Fatras, Younes Zine, Szymon Majewski, Rémi Flamary, Rémi Gribonval, and Nicolas Courty. Minibatch optimal transport distances; analysis and applications. *arXiv preprint arXiv:2101.01792*, 2021. 31

Jean Feydy, Benjamin Charlier, François-Xavier Vialard, and Gabriel Peyré. Optimal transport for diffeomorphic registration. In *Medical Image Computing and Computer Assisted Intervention- MICCAI 2017: 20th International Conference*, pages 291–299. Springer, 2017. 137

Jean Feydy, Thibault Séjourné, François-Xavier Vialard, Shun-ichi Amari, Alain Trouve, and Gabriel Peyré. Interpolating between optimal transport and mmd using sinkhorn divergences. In *International Conference on Artificial Intelligence and Statistics*, pages 2681–2690. PMLR, 2019. 375

Rémi Flamary, Nicolas Courty, Alexandre Gramfort, Mokhtar Z. Alaya, Aurélie Boisbunon, Stanislas Chambon, Laetitia Chapel, Adrien Corenflos, Kilian Fatras, Nemo Fournier, Léo Gautheron, Nathalie T. H. Gayraud, Hicham Janati, Alain Rakotomamonjy, Ievgen Redko, Antoine Rolet, Antony Schutz, Vivien Seguy, Danica J. Sutherland, Romain Tavenard, Alexander Tong, and Titouan Vayer. POT: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021. 225, 336, 340

R. Fletcher. *Practical Methods of Optimization*. Wiley, 2000. 81

Hans Föllmer. An entropy approach to the time reversal of diffusion processes. In *Stochastic Differential Systems Filtering and Control: Proceedings of the IFIP-WG 7/1 Working Conference Marseille-Luminy, France, March 12–17, 1984*, pages 156–163. Springer, 2005. 99

Hans Föllmer. Time reversal on wiener space. In *Stochastic Processes—Mathematics and Physics: Proceedings of the 1st BiBoS-Symposium held in Bielefeld, West Germany, September 10–15, 1984*, pages 119–129. Springer, 2006. 99

Robert Fortet. Résolution d'un système d'équations de M. Schrödinger. *Journal de Mathématiques Pures et Appliquées*, 19(1-4):83–105, 1940. 29, 31, 137, 148

Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. In *International Conference on Learning Representations*, 2025. 358, 359, 361, 362, 365, 376, 378

Daan Frenkel and Berend Smit. *Understanding Molecular Simulation: from Algorithms to Applications*. Elsevier, 2023. 115

Marylou Gabrié, Grant M. Rotskoff, and Eric Vanden-Eijnden. Adaptive monte carlo augmented with normalizing flows. *Proceedings of the National Academy of Sciences*, 119(10):e2109420119, 2022. 10, 101, 111, 115, 134, 226, 255

Ruiqi Gao, Yang Song, Ben Poole, Ying Nian Wu, and Diederik P Kingma. Learning energy-based models by diffusion recovery likelihood. In *International Conference on Learning Representations*, 2021. 109, 119, 249

Sicheng Gao, Xuhui Liu, Bohan Zeng, Sheng Xu, Yanjing Li, Xiaoyan Luo, Jianzhuang Liu, Xiantong Zhen, and Baochang Zhang. Implicit diffusion models for continuous super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10021–10030, 2023. 359, 364

Tomas Geffner and Justin Domke. MCMC variational inference via uncorrected hamiltonian annealing. *Advances in Neural Information Processing Systems*, pages 639–651, 2021. 22, 104

Tomas Geffner and Justin Domke. Langevin diffusion variational inference. In *International Conference on Artificial Intelligence and Statistics*, pages 576–593. PMLR, 2023. 22, 99, 104

A. E. Gelfand, A. F. Smith, and T.-M. Lee. Bayesian analysis of constrained parameter and truncated data problems using Gibbs sampling. *Journal of the American Statistical Association*, 87(418):523–532, 1992. 71

A. Gelman and X.-L. Meng. Simulating Normalizing Constants: From Importance Sampling to Bridge Sampling to Path Sampling. *Statistical Science*, 13:163–185, 1998. 121

Andrew Gelman, Walter R Gilks, and Gareth O Roberts. Weak convergence and optimal scaling of random walk metropolis algorithms. *The annals of applied probability*, 7(1):110–120, 1997. 10

Zhengyang Geng, Mingyang Deng, Xingjian Bai, J Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. In *Advances in Neural Information Processing Systems*, 2025. 358, 361, 362, 363

Charles J Geyer et al. Markov chain monte carlo maximum likelihood. In *Computing science and statistics: Proceedings of the 23rd Symposium on the Interface*, volume 156163. New York, 1991. 15, 85, 91, 103, 110, 121, 247

Davide Ghio, Yatin Dandi, Florent Krzakala, and Lenka Zdeborová. Sampling with flows, diffusion, and autoregressive neural networks from a spin-glass perspective. *Proceedings of the National Academy of Sciences*, 121(27):e2311810121, 2024. 92

Walter R Gilks, Sylvia Richardson, and David Spiegelhalter. *Markov chain Monte Carlo in practice*. CRC press, 1995. 10

Mark Girolami and Ben Calderhead. Riemann Manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2):123–214, 2011. 12, 34, 71, 74, 78, 174

Gerald Goertzel. Quota sampling and importance functions in stochastic solution of particle problems. Technical report, 1949. 6

Will Grathwohl, Ricky T. Q. Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. Ffjord: Free-form continuous dynamics for scalable reversible generative models. *International Conference on Learning Representations*, 2019. 115

Giacomo Greco, Maxence Noble, Giovanni Conforti, and Alain Durmus. Non-asymptotic convergence bounds for sinkhorn iterates and their gradients: a coupling approach. In *Conference on Learning Theory*, pages 716–746. PMLR, 2023. 29, 33

Louis Grenioux, Eric Moulines, and Marylou Gabrié. Balanced training of energy-based models with adaptive flow sampling. In *ICML 2023 Workshop on Structured Probabilistic Inference & Generative Modeling*, 2023a. 91

Louis Grenioux, Alain Oliviero Durmus, Eric Moulines, and Marylou Gabrié. On sampling with approximate transport maps. In *International Conference on Machine Learning*, pages 11698–11733. PMLR, 2023b. 38, 81, 87, 100, 103, 115, 120, 225, 232

Louis Grenioux, Maxence Noble, Marylou Gabrié, and Alain Oliviero Durmus. Stochastic localization via iterative posterior sampling. In *International Conference on Machine Learning*, pages 16337–16376. PMLR, 2024. 33, 86, 103, 111, 118, 199, 201, 270

Louis Grenioux, Maxence Noble, and Marylou Gabrié. Improving the evaluation of samplers on multi-modal targets. In *Frontiers in Probabilistic Inference: Learning meets Sampling*, 2025. 33, 116, 134, 282, 283

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773, 2012. 82, 232

Nikita Gushchin, Daniil Selikhanovych, Sergei Kholkin, Evgeny Burnaev, and Aleksandr Korotin. Adversarial schrödinger bridge matching. *Advances in Neural Information Processing Systems*, pages 89612–89651, 2024. 42, 148, 154, 155, 159

Florentin Guth, Zahra Kadkhodaie, and Eero P Simoncelli. Learning normalized image densities via dual score matching. In *Advances in Neural Information Processing Systems*, 2025. 119, 130, 272

Isabel Haasler, Axel Ringh, Yongxin Chen, and Johan Karlsson. Multimarginal Optimal Transport with a tree-structured cost and the Schrödinger bridge problem. *SIAM Journal on Control and Optimization*, 59(4): 2428–2453, 2021. 15, 40, 137, 138, 139, 140, 144, 317, 331, 332, 333

Ernst Hairer, Marlis Hochbruck, Arieh Iserles, and Christian Lubich. Geometric numerical integration. *Oberwolfach Reports*, 3(1):805–882, 2006. 11, 12, 34, 74, 76, 78, 174, 195, 196

Brian C. Hall. An elementary introduction to groups and representations. *arXiv preprint arXiv:0005032*, 2000. 269, 270

Ulrich G Haussmann and Etienne Pardoux. Time reversal of diffusions. *The Annals of Probability*, pages 1188–1205, 1986. 139, 141, 334

Aaron J Havens, Benjamin Kurt Miller, Bing Yan, Carles Domingo-Enrich, Anuroop Sriram, Daniel S. Levine, Brandon M Wood, Bin Hu, Brandon Amos, Brian Karrer, Xiang Fu, Guan-Horng Liu, and Ricky T. Q. Chen. Adjoint sampling: Highly scalable diffusion samplers via adjoint matching. In *International Conference on Machine Learning*, 2025. 24

Jiajun He, Wenlin Chen, Mingtian Zhang, David Barber, and José Miguel Hernández-Lobato. Training neural samplers with reverse diffusive kl divergence. In *International Conference on Artificial Intelligence and Statistics*, pages 5167–5175. PMLR, 2025a. 22

Jiajun He, Yuanqi Du, Francisco Vargas, Dinghuai Zhang, Shreyas Padhy, RuiKang OuYang, Carla P Gomes, and José Miguel Hernández-Lobato. No trick, no treat: Pursuits and challenges towards simulation-free training of neural samplers. In *Frontiers in Probabilistic Inference: Learning meets Sampling*, 2025b. 21, 38

Jiajun He, José Miguel Hernández-Lobato, Yuanqi Du, and Francisco Vargas. RNE: plug-and-play diffusion inference-time control and energy-based training. In *International Conference on Learning Representations*, 2026. 119, 130, 274

Xiao He, Huaao Tang, Zhijun Tu, Junchao Zhang, Kun Cheng, Hanting Chen, Yong Guo, Mingrui Zhu, Nannan Wang, Xinbo Gao, et al. One step diffusion-based super-resolution with time-aware distillation. *arXiv preprint arXiv:2408.07476*, 2024a. 358, 359

Ye He, Kevin Rojas, and Molei Tao. Zeroth-order sampling methods for non-log-concave distributions: Alleviating metastability by denoising diffusion. In *Advances in Neural Information Processing Systems*, 2024b. 25, 86

Jonathan Heek, Emiel Hoogeboom, and Tim Salimans. Multistep consistency models. *arXiv preprint arXiv:2403.06807*, 2024. 361

Laurent Heirendt, Sylvain Arreckx, Thomas Pfau, Sebastián N Mendoza, Anne Richelle, Almut Heinken, Hulda S Haraldsdóttir, Jacek Wachowiak, Sarah M Keating, Vanja Vlasov, et al. Creation and analysis of biochemical constraint-based models using the COBRA Toolbox v. 3.0. *Nature protocols*, 14(3):639–702, 2019. 79

Jeremy Heng, Adrian N Bishop, George Deligiannidis, and Arnaud Doucet. Controlled sequential monte carlo. *The Annals of Statistics*, 48(5):2904–2929, 2020. 42, 154, 157

Jeremy Heng, Valentin De Bortoli, and Arnaud Doucet. Diffusion schrödinger bridges for bayesian computation. *Statistical Science*, 39(1):90–99, 2024. 22

Jose Hernandez-Lobato, Yingzhen Li, Mark Rowland, Thang Bui, Daniel Hernandez-Lobato, and Richard Turner. Black-box alpha divergence minimization. In *International Conference on Machine Learning*, pages 1511–1520. PMLR, 2016. 9

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in Neural Information Processing Systems*, 2017. 368

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 21, 361

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, pages 6840–6851, 2020. 2, 4, 16, 17, 18, 42, 91, 103, 115, 119, 137, 139, 156, 358, 359

Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P. Kingma, Ben Poole, Mohammad Norouzi, David J. Fleet, and Tim Salimans. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022. 16, 115

Matthew D. Hoffman and Andrew Gelman. The No-U-Turn Sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15(47):1593–1623, 2014. 12, 82, 103, 259

Lars Holdijk, Yuanqi Du, Ferry Hooft, Priyank Jaini, Bernd Ensing, and Max Welling. PIPS: Path integral stochastic optimal control for path sampling in molecular dynamics. In *International Conference on Learning Representations*, 2023. 91, 104

Samuel Howard, Peter Potaptchik, and George Deligiannidis. Schrödinger bridge matching for tree-structured costs and entropic wasserstein barycentres. In *Advances in Neural Information Processing Systems*, 2025. 32

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022. 359, 368

Zheyuan Hu, Chieh-Hsin Lai, Yuki Mitsufuji, and Stefano Ermon. Cmt: Mid-training for efficient learning of consistency, mean flow, and flow map models. *arXiv preprint arXiv:2509.24526*, 2025. 358

Jerry Y. Huang, Justin Lin, Sheel Shah, Kartik Nair, and Nicholas M. Boffi. How to guide your flow: Few-step alignment via flow map reward guidance. *arXiv preprint arXiv:2604.27147*, 2026. 358

Jian Huang, Yuling Jiao, Lican Kang, Xu Liao, Jin Liu, and Yanyan Liu. Schrödinger-Föllmer sampler: sampling without ergodicity. *arXiv preprint arXiv:2106.10880*, 2021. 25, 99

Xunpeng Huang, Hanze Dong, Yifan HAO, Yian Ma, and Tong Zhang. Reverse diffusion monte carlo. In *International Conference on Learning Representations*, 2024a. 14, 25, 36, 86, 91, 93, 96, 99, 100, 101, 102, 103, 203, 221, 224

Xunpeng Huang, Difan Zou, Hanze Dong, Yi-An Ma, and Tong Zhang. Faster sampling without isoperimetry via diffusion-based Monte Carlo. In *Conference on Learning Theory*, pages 2438–2493. PMLR, 2024b. 25, 86, 103

Xunpeng Huang, Difan Zou, Hanze Dong, Yian Ma, and Tong Zhang. Faster sampling via stochastic gradient proximal sampler. In *International Conference on Machine Learning*, pages 20559–20596. PMLR, 2024c. 25

Koji Hukushima and Koji Nemoto. Exchange monte carlo method and application to spin glass simulations. *Journal of the Physical Society of Japan*, 65(6):1604–1608, 1996. 85, 121

M.F. Hutchinson. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics - Simulation and Computation*, 18(3):1059–1076, 1989. 127

Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(24):695–709, 2005. 91, 105, 108, 119, 139, 141

Michele Invernizzi, Andreas Krämer, Cecilia Clementi, and Frank Noé. Skipping the replica exchange ladder with normalizing flows. *The Journal of Physical Chemistry Letters*, 13(50):11643–11649, 2022. 128

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1125–1134, 2017. 377

Tommi S. Jaakkola and Michael I. Jordan. Improving the mean field approximation via the use of mixture distributions. In *Learning in Graphical Models*, pages 163–173. Springer, 1998. 8

Yazid Janati, Badr Moufad, Alain Durmus, Eric Moulines, and Jimmy Olsson. Divide-and-conquer posterior sampling for denoising diffusion priors. In *Advances in Neural Information Processing Systems*, pages 97408–97444, 2024. 128

Yazid Janati, Eric Moulines, Jimmy Olsson, and Alain Oliviero-Durmus. Bridging diffusion posterior sampling and monte carlo methods: a survey. *Philosophical Transactions A*, 383(2299):20240331, 2025. 128, 358

Ghassen Jerfel, Serena Wang, Clara Wong-Fannjiang, Katherine A. Heller, Yian Ma, and Michael I. Jordan. Variational refinement for importance sampling using the forward kullback-leibler divergence. In *Uncertainty in Artificial Intelligence*, pages 1819–1829. PMLR, 2021. 9, 85, 104

A Jolicoeur-Martineau. The relativistic discriminator: A key element missing from standard gan. *arXiv preprint arXiv:1807.00734*, 2018. 367

Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Saul. *An Introduction to Variational Methods for Graphical Models*, pages 105–161. Springer Netherlands, Dordrecht, 1998. 7

William L. Jorgensen. The Many Roles of Computation in Drug Discovery. *Science*, 303(5665):1813–1818, 2004. 82

Ravi Kannan and Hariharan Narayanan. Random walks on polytopes and an affine interior point method for linear programming. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 561–570, 2009. 78

Leonid Kantorovitch. On the translocation of masses. *Management science*, 5(1):1–4, 1958. 27

Hermann Karcher. Riemannian center of mass and so called Karcher mean. *arXiv preprint arXiv:1407.2087*, 2014. 138

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, pages 26565–26577, 2022. 17, 18, 93, 108, 117, 130, 258, 271, 272, 279

Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24174–24184, 2024. 18, 115, 270, 365

Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. Musiq: Multi-scale image quality transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5148–5157, 2021. 368

Beomsu Kim and Jong Chul Ye. Denoising MCMC for accelerating diffusion-based generative models. In *International Conference on Machine Learning*, pages 16955–16977. PMLR, 2023. 250

Beomsu Kim, Yu-Guan Hsieh, Michal Klein, Marco Cuturi, Jong Chul Ye, Bahjat Kawar, and James Thornton. Simple reflow: Improved techniques for fast flow models. In *International Conference on Learning Representations*, 2025a. 358, 361

Beomsu Kim, Jaemin Kim, Jeongsol Kim, and Jong Chul Ye. Generalized consistency trajectory models for image manipulation. In *International Conference on Learning Representations*, 2025b. 361

Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ODE trajectory of diffusion. In *International Conference on Learning Representations*, 2024. 361

Minkyu Kim, Kiyoung Seong, Dongyeop Woo, Sungsoo Ahn, and Minsu Kim. On scalable and efficient training of diffusion samplers. In *Advances in Neural Information Processing Systems*, 2025c. 22

Sunwoo Kim, Minkyu Kim, and Dongmin Park. Test-time alignment of diffusion models without reward overoptimization. In *International Conference on Learning Representations*, 2025d. 128

Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Advances in Neural Information Processing Systems*, pages 21696–21707, 2021. 93, 95

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 336

Diederik P Kingma and Ruiqi Gao. Understanding diffusion objectives as the ELBO with simple data augmentation. In *Advances in Neural Information Processing Systems*, 2023. 93, 95

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *International Conference on Learning Representations*, 2014. 8, 91

Leon Klein and Frank Noé. Transferable boltzmann generators. In *Advances in Neural Information Processing Systems*, pages 45281–45314, 2024. 115

Leon Klein, Andreas Krämer, and Frank Noé. Equivariant flow matching. *Advances in Neural Information Processing Systems*, 2023. 115

Philip A Knight. The Sinkhorn–Knopp algorithm: convergence and applications. *SIAM Journal on Matrix Analysis and Applications*, 30(1):261–275, 2008. 137, 139, 332

H Kober. A theorem on Banach spaces. *Compositio Mathematica*, 7:135–140, 1940. 326

Jonas Köhler, Leon Klein, and Frank Noé. Equivariant flows: exact likelihood generative learning for symmetric densities. *International Conference on Machine Learning*, 2020. 115

Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT press, 2009. 316

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations*, 2021. 115

Yunbum Kook, Yin-Tat Lee, Ruoqi Shen, and Santosh Vempala. Sampling with Riemannian Hamiltonian Monte Carlo in a constrained space. *Advances in Neural Information Processing Systems*, pages 31684–31696, 2022. 13, 12, 34, 71, 76, 78, 79, 80, 163, 194, 195, 196

Yunbum Kook, Yin Tat Lee, Ruoqi Shen, and Santosh Vempala. Condition-number-independent convergence rate of riemannian hamiltonian monte carlo with numerical integrators. In *Conference on Learning Theory*, pages 4504–4569. PMLR, 2023. 79

Nikita Kornilov, Petr Mokrov, Alexander Gasnikov, and Aleksandr Korotin. Optimal flow matching: Learning straight trajectories in just one step. *Advances in Neural Information Processing Systems*, pages 104180–104204, 2024. 361

Alexander Korotin, Lingxiao Li, Justin Solomon, and Evgeny Burnaev. Continuous wasserstein-2 barycenter estimation without minimax optimization. In *International Conference on Learning Representations*, 2021. 144, 145, 333, 338, 341

Alexander Korotin, Vage Egiazarian, Lingxiao Li, and Evgeny Burnaev. Wasserstein iterative networks for barycenter estimation. In *Advances in Neural Information Processing Systems*, 2022. 144, 146, 333

Werner Krauth. Statistical mechanics: algorithms and computations. *OUP Oxford*, 13, 2006. 6, 91, 103, 115

Dirk P. Kroese, Thomas Taimre, and Zdravko I. Botev. *Handbook of Monte Carlo methods*. Number 706 in Wiley series in probability and statistics. Wiley, Hoboken, 2011. 5, 91, 103

Solomon Kullback. Probability densities with given marginals. *The Annals of Mathematical Statistics*, 39(4): 1236–1243, 1968. 29, 137

Chieh-Hsin Lai, Yuhta Takida, Naoki Murata, Toshimitsu Uesaka, Yuki Mitsufuji, and Stefano Ermon. Fp-diffusion: Improving score-based diffusion models by enforcing the underlying score fokker-planck equation. In *International Conference on Machine Learning*, pages 18365–18398. PMLR, 2023. 273

Shiwei Lan and Babak Shahbaba. Sampling constrained probability distributions using spherical augmentation. In *Algorithmic Advances in Riemannian Geometry and Applications: For Machine Learning, Computer Vision, Statistics, and Optimization*, pages 25–71. Springer, 2016. 71

Yann LeCun. The MNIST database of handwritten digits. 1998. 145

John M Lee. *Riemannian Manifolds: an introduction to curvature*, volume 176. Springer Science & Business Media, 2006. 72, 164

Sangyun Lee, Zinan Lin, and Giulia Fanti. Improving the training of rectified flows. *Advances in Neural Information Processing Systems*, pages 63082–63109, 2024. 20, 361

Yin Tat Lee and Santosh S Vempala. Geodesic walks in polytopes. In *Proceedings of the 49th Annual ACM SIGACT Symposium on theory of Computing*, pages 927–940, 2017a. 78

Yin Tat Lee and Santosh S Vempala. Convergence rate of Riemannian Hamiltonian Monte Carlo and faster polytope volume computation. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1115–1121, 2018. 78

Yin Tat Lee and Santosh Srinivas Vempala. Eldan's stochastic localization and the KLS hyperplane conjecture: an improved lower bound for expansion. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 998–1007. IEEE, 2017b. 78

Benedict J Leimkuhler and Robert D Skeel. Symplectic numerical integrators in constrained Hamiltonian systems. *Journal of Computational Physics*, 112(1):117–125, 1994. 78

Tony Lelièvre, Mathias Rousset, and Gabriel Stoltz. Hybrid Monte Carlo methods for sampling probability measures on submanifolds. *Numerische Mathematik*, 143(2):379–421, 2019. 71, 75, 79

Tony Lelièvre, Gabriel Stoltz, and Wei Zhang. Multiple projection Markov Chain Monte Carlo algorithms on submanifolds. *IMA Journal of Numerical Analysis*, 2022. 71, 75, 77, 79

Tony Lelièvre, Gabriel Stoltz, and Mathias Rousset. *Free energy computations: a mathematical perspective*. Imperial College Press, London ; Hackensack, N.J, 2010. 82

Christian Léonard. Stochastic derivatives and generalized h-transforms of markov processes. *arXiv preprint arXiv:1102.3172*, 2011. 151, 347, 348

Christian Léonard. Some properties of path measures. *Séminaire de Probabilités XLVI*, pages 207–230, 2014a. 258

Christian Léonard. A survey of the Schrödinger problem and some of its connections with optimal transport. *Discrete & Continuous Dynamical Systems-A*, 34(4):1533–1574, 2014b. 30, 31, 139, 143, 148, 149, 150, 154

Nathan E Lewis, Harish Nagarajan, and Bernhard O Palsson. Constraining the metabolic genotype–phenotype relationship using a phylogeny of in silico methods. *Nature Reviews Microbiology*, 10(4):291–305, 2012. 5, 71

Haoying Li, Yifan Yang, Meng Chang, Shiqi Chen, Huajun Feng, Zhihai Xu, Qi Li, and Yueting Chen. Srdiff: Single image super-resolution with diffusion probabilistic models. *Neurocomputing*, 479:47–59, 2022. 358, 359

Liangchen Li and Jiajun He. Bidirectional consistency models. In *ICML 2024 Workshop on Structured Probabilistic Inference & Generative Modeling*, 2024. 361

Lingxiao Li, Aude Genevay, Mikhail Yurochkin, and Justin M Solomon. Continuous regularized Wasserstein barycenters. *Advances in Neural Information Processing Systems*, pages 17755–17765, 2020. 144, 145, 146, 333, 336, 341

Tiancheng Li, Shudong Sun, Tariq Pervez Sattar, and Juan Manuel Corchado. Fight sample degeneracy and impoverishment in particle filters: A review of intelligent approaches. *Expert Systems with applications*, 41 (8):3944–3954, 2014. 85

Xinqi Lin, Jingwen He, Ziyan Chen, Zhaoyang Lyu, Bo Dai, Fanghua Yu, Yu Qiao, Wanli Ouyang, and Chao Dong. Diffbir: Toward blind image restoration with generative diffusion prior. In *European conference on computer vision*, pages 430–448. Springer, 2024. 358, 359

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *International Conference on Learning Representations*, 2023. 16, 18, 146, 147, 358, 360

Robert Shevilevich Liptser and Albert Nikolaevich Shiryaev. *Statistics of random processes: General theory*, volume 394. Springer, 1977. 92, 208, 209

Guan-Horng Liu, Yaron Lipman, Maximilian Nickel, Brian Karrer, Evangelos Theodorou, and Ricky T. Q. Chen. Generalized schrödinger bridge matching. In *International Conference on Learning Representations*, 2024. 32, 33, 42, 148, 150, 151, 153, 157, 158, 159, 350

Guan-Horng Liu, Jaemoo Choi, Yongxin Chen, Benjamin Kurt Miller, and Ricky T. Q. Chen. Adjoint schrödinger bridge sampler. In *Advances in Neural Information Processing Systems*, 2025a. 24

Jun S Liu. Metropolized independent sampling with comparisons to rejection sampling and importance sampling. *Statistics and computing*, 6:113–119, 1996. 79, 196

Jun S Liu and Jun S Liu. *Monte Carlo strategies in scientific computing*, volume 10. Springer, 2001. 5, 71, 103, 115

Qiang Liu. Rectified flow: A marginal preserving approach to optimal transport. *arXiv preprint arXiv:2209.14577*, 2022. 20, 31, 361

Xingchao Liu, Chengyue Gong, and Qiang liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023. 16, 20, 31, 358

Xinyang Liu, Hengrong Du, Wei Deng, and Ruqi Zhang. Optimal stochastic trace estimation in generative modeling. In *International Conference on Artificial Intelligence and Statistics*, pages 4600–4608. PMLR, 2025b. 134

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 283

L. Lovász and R. Kannan. Faster mixing via average conductance. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pages 282–287. ACM, 1999. 71

L. Lovász and M. Simonovits. Random walks in a convex body and an improved volume algorithm. *Random structures & algorithms*, 4(4):359–412, 1993. 71

László Lovász and Santosh Vempala. Hit-and-run from a corner. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 310–314, 2004. 78

László Lovász and Santosh Vempala. The geometry of logconcave functions and sampling algorithms. *Random Struct. Algorithms*, 30(3):307–358, 2007. 71

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *Machine Intelligence Research*, pages 1–22, 2025. 358

Zhisheng Lu, Juncheng Li, Hong Liu, Chaoyan Huang, Linlin Zhang, and Tieyong Zeng. Transformer for single image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 457–466, 2022. 358

Zhi-Quan Luo and Paul Tseng. On the convergence of the coordinate descent method for convex differentiable minimization. *Journal of Optimization Theory and Applications*, 72(1):7–35, 1992. 332

Ziwei Luo, Fredrik K. Gustafsson, Zheng Zhao, Jens Sjölund, and Thomas B. Schön. Image restoration with mean-reverting stochastic differential equations. In *International Conference on Machine Learning*, pages 23045–23066. PMLR, 2023. 358, 359

David JC MacKay. Local minima, symmetry-breaking, and model pruning in variational free energy minimization. *Inference Group, Cavendish Laboratory, Cambridge, UK*, 2001. 85

Ashok Makkuva, Amirhossein Taghvaei, Sewoong Oh, and Jason Lee. Optimal transport mapping via input convex neural networks. In *International Conference on Machine Learning*, pages 6672–6681. PMLR, 2020. 144

Simone Di Marino and Augusto Gerolin. An optimal transport approach for the Schrödinger bridge problem and convergence of Sinkhorn algorithm. *Journal of Scientific Computing*, 85(2):27, 2020. 40, 137, 140, 142, 143

Bálint Máté and François Fleuret. Learning interpolations between boltzmann densities. *Transactions on Machine Learning Research*, 2023. 20, 23, 39, 85, 86, 121, 122, 201

Alex Matthews, Michael Arbel, Danilo Jimenez Rezende, and Arnaud Doucet. Continual repeated annealed flow transport monte carlo. In *International Conference on Machine Learning*, pages 15196–15219. PMLR, 2022. 104, 128

Curtis James McDonald and Andrew R. Barron. Proposal of a score based approach to sampling using monte carlo estimation of score and oracle access to target density. In *NeurIPS 2022 Workshop on Score-Based Methods*, 2022. 25

Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953. 9, 10, 82, 120, 259

Liang Mi, Tianshu Yu, Jose Bento, Wen Zhang, Baoxin Li, and Yalin Wang. Variational Wasserstein barycenters for geometric clustering. *arXiv preprint arXiv:2002.10543*, 2020. 144

L. I. Midgley, V. Stimper, G. N. C. Simm, and J. M. Hernández-Lobato. Bootstrap your flow. In *1st ELLIS Machine Learning for Molecule Discovery Workshop*, 2021. 128

Laurence Midgley, Vincent Stimper, Javier Antorán, Emile Mathieu, Bernhard Schölkopf, and José Miguel Hernández-Lobato. Se(3) equivariant augmented coupling flows. In *Advances in Neural Information Processing Systems*, pages 79200–79225, 2023a. 128

Laurence Illing Midgley, Vincent Stimper, Gregor N. C. Simm, Bernhard Schölkopf, and José Miguel Hernández-Lobato. Flow annealed importance sampling bootstrap. In *International Conference on Learning Representations*, 2023b. 105, 128

Stanislav Minsker, Sanvesh Srivastava, Lizhen Lin, and David Dunson. Scalable and robust Bayesian inference via the median posterior. In *International Conference on Machine Learning*, pages 1656–1664. PMLR, 2014. 338

Kam-Ping Mok. Metrics and connections on the cotangent bundle. In *Kodai Mathematical Seminar Reports*, volume 28, pages 226–238. Department of Mathematics, Tokyo Institute of Technology, 1977. 72, 165

Gaspard Monge. Mémoire sur la théorie des déblais et des remblais. *Mem. Math. Phys. Acad. Royale Sci.*, pages 666–704, 1781. 26

Andrea Montanari. Sampling, diffusions, and stochastic localization. *arXiv preprint arXiv:2305.10690*, 2023. 36, 92, 93, 199, 205, 207

Andrea Montanari and Yuchen Wu. Posterior sampling from the spiked models via diffusion processes. *arXiv preprint arXiv:2304.11449*, 2023. 91, 92, 93

Ben J Morris. Improved bounds for sampling contingency tables. *Random Structures & Algorithms*, 21(2): 135–146, 2002. 71

Badr Moufad, Yazid Janati, Lisa Bedin, Alain Oliviero Durmus, Eric Moulines, and Jimmy Olsson. Variational diffusion posterior sampling with midpoint guidance. In *International Conference on Learning Representations*, 2025. 158

Thomas Müller, Brian Mcwilliams, Fabrice Rousselle, Markus Gross, and Jan Novák. Neural importance sampling. *ACM Trans. Graph.*, 38(5), 2019. 84, 115

Kimia Nadjahi, Alain Durmus, Umut Simsekli, and Roland Badeau. Asymptotic guarantees for learning generative models with the sliced-wasserstein distance. *Advances in Neural Information Processing Systems*, 2019. 212, 225

Radford M Neal. Annealed importance sampling. *Statistics and computing*, 11:125–139, 2001. 13, 14, 91, 103, 120, 121, 245

Radford M. Neal. Slice sampling. *The Annals of Statistics*, 31(3):705 – 767, 2003. 225

Radford M Neal et al. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2(11):2, 2011. 12, 71, 103, 120

Yu Nesterov and Arkadi Nemirovski. Multi-parameter surfaces of analytic centers and long-step surface-following interior point methods. *Mathematics of Operations Research*, 23(1):1–38, 1998. 12, 34, 73, 166

Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2003. 166

Yurii Nesterov and Arkadii Nemirovskii. *Interior-point polynomial algorithms in convex programming*. SIAM, 1994. 34, 71, 73, 78, 165, 167

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021. 17, 93, 95, 119, 234, 276, 336

Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen. GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. In *International Conference on Machine Learning*, pages 16784–16804. PMLR, 2022. 91

Maxence Noble and Louis Grenioux. Diffusion-based annealed boltzmann generators : benefits, pitfalls and hopes. *arXiv preprint arXiv:2601.21026*, 2026. 33

Maxence Noble, Valentin De Bortoli, Arnaud Doucet, and Alain Durmus. Tree-based diffusion schrödinger bridge with applications to wasserstein barycenters. *Advances in Neural Information Processing Systems*, pages 55193–55236, 2023a. 33

Maxence Noble, Valentin De Bortoli, and Alain Durmus. Unbiased constrained sampling with self-concordant barrier hamiltonian monte carlo. *Advances in Neural Information Processing Systems*, pages 32672–32719, 2023b. 33, 126

Maxence Noble, Louis Grenioux, Marylou Gabrié, and Alain Oliviero Durmus. Learned reference-based diffusion sampling for multi-modal distributions. In *International Conference on Learning Representations*, 2025. 33, 81, 86, 116, 119, 133, 202, 283

Maxence Noble, Gonzalo Iñaki Quintana, Benjamin Aubin, and Clément Chadebec. Fast, faithful and photorealistic diffusion-based image super-resolution with enhanced flow map models. *arXiv preprint arXiv:2601.16660*, 2026. 33

Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer New York, 2006. 81

Frank Noé, Simon Olsson, Jonas Köhler, and Hao Wu. Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science*, 365(6457):eaaw1147, 2019. 9, 24, 25, 84, 103, 115

Nikolas Nüsken and Lorenz Richter. Solving high-dimensional hamilton–jacobi–bellman PDEs using neural networks: perspectives from the theory of controlled diffusions and measures on path space. *Partial differential equations and applications*, 2(4):48, 2021. 8, 105, 151, 348

Marcel Nutz. *Introduction to Entropic Optimal Transport*. Lecture notes, Columbia University, 2021. 28, 41, 142, 320, 322, 323

Kaoru Ohno, Keivan Esfarjani, and Yoshiyuki Kawazoe. *Computational Materials Science: From Ab Initio to Monte Carlo Methods*. Springer, 2018. 115

Tsuneyasu Okabe, Masaaki Kawata, Yuko Okamoto, and Masuhiro Mikami. Replica-exchange monte carlo method for the isobaric–isothermal ensemble. *Chemical Physics Letters*, 335(5):435–439, 2001. 15, 121

Bernt Øksendal. Stochastic differential equations. In *Stochastic differential equations: an introduction with applications*, pages 38–50. Springer, 2003. 273

Frank WJ Olver. *NIST handbook of mathematical functions*. Cambridge university press, 2010. 213

Zijing Ou, Mingtian Zhang, Andi Zhang, Tim Z. Xiao, Yingzhen Li, and David Barber. Improving probabilistic diffusion models with optimal diagonal covariance matching. In *International Conference on Learning Representations*, 2025. 39, 118, 119, 124

RuiKang OuYang, Bo Qiang, and José Miguel Hernández-Lobato. BNEM: A boltzmann sampler based on bootstrapped noised energy matching. *arXiv preprint arXiv:2409.09787*, 2024. 24, 25

Ari Pakman and Liam Paninski. Exact Hamiltonian Monte Carlo for truncated multivariate Gaussians. *Journal of Computational and Graphical Statistics*, 23(2):518–542, 2014. 71

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021. 9, 104, 115

P. M. Pardalos and G. Schnitger. Checking local optimality in constrained quadratic programming is NP-hard. *Operations Research Letters*, 7(1):33–35, 1988. 81

Brendan Pass. Multi-marginal optimal transport: theory and applications. *ESAIM: Mathematical Modelling and Numerical Analysis*, 49(6):1771–1790, 2015. 27

Grigorios A Pavliotis. Stochastic processes and applications. *Texts in applied mathematics*, 60, 2014. 11

Michele Pavon. On local entropy, stochastic control, and deep neural networks. *IEEE Control Systems Letters*, 7:437–441, 2022. 91, 104

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011. 258

Ofir Pele and Michael Werman. Fast and robust Earth mover's distances. In *2009 IEEE 12th International Conference on Computer Vision*, pages 460–467. IEEE, 2009. 137

Stefano Peluchetti. Diffusion bridge mixture transports, schrödinger bridge problems and generative modeling. *Journal of Machine Learning Research*, 24(374):1–51, 2023. 31, 146, 150

Stefano Peluchetti. BM^2: Coupled schrödinger bridge matching. *Transactions on Machine Learning Research*, 2025. 32, 148

Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019. 26, 29, 82, 137, 138, 139, 143, 203, 232, 324

Angus Phillips, Hai-Dang Dau, Michael John Hutchinson, Valentin De Bortoli, George Deligiannidis, and Arnaud Doucet. Particle denoising diffusion sampler. In *International Conference on Machine Learning*, pages 40688–40724. PMLR, 2024. 24, 25, 39, 103, 104, 110, 119, 123, 128, 130, 132, 134, 246, 247, 257

Michael Plainer, Hao Wu, Leon Klein, Stephan Günnemann, and Frank Noe. Consistent sampling and simulation: Molecular dynamics with energy-based diffusion models. In *Advances in Neural Information Processing Systems*, 2025. 23, 119, 130, 132, 273

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *International Conference on Learning Representations*, 2024. 368, 377

Emilia Pompe, Chris Holmes, and Krzysztof Łatuszyński. A framework for adaptive MCMC targeting multimodal distributions. *The Annals of Statistics*, 48(5):2930–2952, 10 2020. 103

Aram-Alexandre Pooladian and Jonathan Niles-Weed. Entropic estimation of optimal transport maps. *arXiv preprint arXiv:2109.12004*, 2021. 29

Aram-Alexandre Pooladian, Heli Ben-Hamu, Carles Domingo-Enrich, Brandon Amos, Yaron Lipman, and Ricky T. Q. Chen. Multisample flow matching: Straightening flows with minibatch couplings. In *International Conference on Machine Learning*, pages 28100–28127. PMLR, 2023. 31, 361

Peter Potaptchik, Adhi Saravanan, Abbas Mammadov, Alvaro Prat, Michael Samuel Albergo, and Yee Whye Teh. Meta flow maps enable scalable reward alignment. In *International Conference on Machine Learning*. PMLR, 2026. 358

Julien Rabin, Gabriel Peyré, Julie Delon, and Marc Bernot. Wasserstein barycenter and its application to texture mixing. In *Scale Space and Variational Methods in Computer Vision: Third International Conference*, pages 435–446. Springer, 2012. 144

Jean-Baptiste Regli and Ricardo Silva. Alpha-beta divergence for variational inference. *arXiv preprint arXiv:1805.01045*, 2018. 9

Alfréd Rényi. On measures of entropy and information. In *Proceedings of the fourth Berkeley symposium on mathematical statistics and probability, volume 1: contributions to the theory of statistics*, volume 4, pages 547–562. University of California Press, 1961. 7

Teodora Reu, Francisco Vargas, Anna Kerekes, and Michael M. Bronstein. To smooth a cloud or to pin it down: Expressiveness guarantees and insights on score matching in denoising diffusion models. In *Uncertainty in Artificial Intelligence*, pages 3094–3120. PMLR, 2024. 22, 234

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on Machine Learning*, pages 1530–1538. PMLR, 2015. 8, 85, 91, 115

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, pages 1278–1286. PMLR, 2014. 8, 91, 104

Lorenz Richter, Julius Berner, and Guan-Horng Liu. Improved sampling via learned diffusions. In *ICML Workshop on New Frontiers in Learning, Control, and Dynamical Systems*, 2023. 22, 37, 38, 91, 99, 104, 105, 106, 110, 200, 233, 234, 236, 241, 257, 283

H. Risken. *The Fokker–Planck Equation: Methods of Solution and Applications*. Springer, 1989. 19

Severi Rissanen, RuiKang OuYang, Jiajun He, Wenlin Chen, Markus Heinonen, Arno Solin, and José Miguel Hernández-Lobato. Progressive tempering sampler with diffusion. In *International Conference on Machine Learning*, 2025. 25

Herbert E Robbins. An empirical bayes approach to statistics. In *Breakthroughs in Statistics: Foundations and basic theory*, pages 388–394. Springer, 1992. 17, 92, 204

Christian P. Robert and George Casella. *Monte Carlo Statistical Methods*. Springer, 2nd edition, 2010. 4

Gareth O Roberts and Jeffrey S Rosenthal. Optimal scaling of discrete approximations to langevin diffusions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60(1):255–268, 1998. 10

Gareth O Roberts and Jeffrey S Rosenthal. Optimal scaling for various Metropolis-Hastings algorithms. *Statistical science*, 16(4):351–367, 2001. 79

Gareth O Roberts and Osnat Stramer. Langevin diffusions and Metropolis-Hastings algorithms. *Methodology and computing in applied probability*, 4:337–357, 2002. 79, 196

Gareth O. Roberts and Richard L. Tweedie. Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli*, 2(4):341–363, 1996. 10, 82, 96, 97, 103, 107, 120, 158, 197, 244, 259, 284

L Chris G Rogers and David Williams. *Diffusions, Markov processes, and martingales: Itô calculus*, volume 2. Cambridge university press, 2000. 150, 347

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 2, 4, 16, 91, 359

Bernard Roynette and Marc Yor. *Penalising brownian paths*, volume 1969. Springer Science & Business Media, 2009. 150

Ludger Ruschendorf. Convergence of the Iterative Proportional Fitting procedure. *The Annals of Statistics*, pages 1160–1174, 1995. 29, 41, 137, 138, 143, 324, 325, 326, 327, 328

Amirmojtaba Sabour, Sanja Fidler, and Karsten Kreis. Align your flow: Scaling continuous-time flow map distillation. In *Advances in Neural Information Processing Systems*, 2025. 358, 361, 362, 366, 367, 374

Matthias Sachs, Benedict Leimkuhler, and Vincent Danos. Langevin dynamics with variable coefficients and nonconservative forces: from stationary states to numerical methods. *Entropy*, 19(12):647, 2017. 222

Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE transactions on pattern analysis and machine intelligence*, 45 (4):4713–4726, 2022. 358, 359

Tim Salimans and Jonathan Ho. Should EBMs model the energy or the score? In *Energy Based Models Workshop-ICLR 2021*, 2021. 119, 258

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022. 17, 358

Sergey Samsonov, Evgeny Lagutin, Marylou Gabrié, Alain Durmus, Alexey Naumov, and Eric Moulines. Local-global MCMC kernels: the best of both worlds. In *Advances in Neural Information Processing Systems*, 2022. 115

Saeed Saremi and Aapo Hyvärinen. Neural empirical bayes. *Journal of Machine Learning Research*, 20(181): 1–23, 2019. 92, 93

Saeed Saremi and Rupesh Kumar Srivastava. Multimeasurement generative models. In *International Conference on Learning Representations*, 2022. 14, 100, 222

Saeed Saremi, Rupesh Kumar Srivastava, and Francis Bach. Universal smoothed score functions for generative modeling. *arXiv preprint arXiv:2303.11669*, 2023. 14, 222

Saeed Saremi, Ji Won Park, and Francis Bach. Chain of log-concave markov chains. In *International Conference on Learning Representations*, 2024. 14, 26, 96, 100, 101, 203, 207, 216, 222, 223, 224

Shigeo Sasaki. On the differential geometry of tangent bundles of Riemannian manifolds. *Tohoku Mathematical Journal, Second Series*, 10(3):338–354, 1958. 165

Axel Sauer, Frederic Boesel, Tim Dockhorn, Andreas Blattmann, Patrick Esser, and Robin Rombach. Fast high-resolution image synthesis with latent adversarial diffusion distillation. In *SIGGRAPH Asia 2024 Conference Papers*, pages 1–11, 2024a. 358

Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *European Conference on Computer Vision*, pages 87–103. Springer, 2024b. 358

Geoffrey Schiebinger, Jian Shu, Marcin Tabaka, Brian Cleary, Vidya Subramanian, Aryeh Solomon, Joshua Gould, Siyan Liu, Stacie Lin, Peter Berube, et al. Optimal-transport analysis of single-cell gene expression identifies developmental trajectories in reprogramming. *Cell*, 176(4):928–943, 2019. 2, 4, 137

Morgan A Schmitz, Matthieu Heitz, Nicolas Bonneel, Fred Ngole, David Coeurjolly, Marco Cuturi, Gabriel Peyré, and Jean-Luc Starck. Wasserstein dictionary learning: Optimal transport-based unsupervised nonlinear dictionary learning. *SIAM Journal on Imaging Sciences*, 11(1):643–678, 2018. 137

Erwin Schrödinger. Sur la théorie relativiste de l'électron et l'interprétation de la mécanique quantique. *Annales de l'Institut Henri Poincaré*, 2(4):269–310, 1932. 30, 137, 143, 147, 150

Babak Shahbaba, Shiwei Lan, Wesley O Johnson, and Radford M Neal. Split Hamiltonian Monte Carlo. *Statistics and Computing*, 24(3):339–349, 2014. 74

David E Shaw, Paul Maragakis, Kresten Lindorff-Larsen, Stefano Piana, Ron O Dror, Michael P Eastwood, Joseph A Bank, John M Jumper, John K Salmon, Yibing Shan, et al. Atomic-level characterization of the structural dynamics of proteins. *Science*, 330(6002):341–346, 2010. 2, 3

Dazhong Shen, Guanglu Song, Yi Zhang, Bingqi Ma, Lujundong Li, Dongzhi Jiang, Zhuofan Zong, and Yu Liu. Adt: Tuning diffusion models with adversarial supervision. *arXiv preprint arXiv:2504.11423*, 2025. 367

Yuyang Shi, Valentin De Bortoli, Andrew Campbell, and Arnaud Doucet. Diffusion schrödinger bridge matching. *Advances in Neural Information Processing Systems*, pages 62183–62223, 2023. 31, 32, 41, 146, 148, 149, 150, 159, 351

Zhekun Shi, Longlin Yu, Tianyu Xie, and Cheng Zhang. Diffusion-pinn sampler. *arXiv preprint arXiv:2410.15336*, 2024. 23, 81, 119, 130, 132, 273

Marta Gentiloni Silveri, Giovanni Conforti, and Alain Durmus. Exponential convergence guarantees for iterative markovian fitting. *arXiv preprint arXiv:2510.20871*, 2025. 32, 149

Raghav Singhal, Zachary Horvitz, Ryan Teehan, Mengye Ren, Zhou Yu, Kathleen McKeown, and Rajesh Ranganath. A general framework for inference-time scaling and steering of diffusion models. In *International Conference on Machine Learning*, 2025. 128

Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348, 1967. 29, 137, 139, 332

Jeffrey Mark Siskind. Automatic differentiation: Inverse accumulation mode. In *Program Transformations for ML Workshop at NeurIPS 2019*, 2019. 128

Marta Skreta, Tara Akhound-Sadegh, Viktor Ohanesian, Roberto Bondesan, Alan Aspuru-Guzik, Arnaud Doucet, Rob Brekelmans, Alexander Tong, and Kirill Neklyudov. Feynman-kac correctors in diffusion: Annealing, guidance, and product of experts. In *International Conference on Machine Learning*, 2025. 128

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015. 16, 91, 103, 115, 358

Roman Soletskyi, Marylou Gabrié, and Bruno Loureiro. A theoretical perspective on mode collapse in variational inference. *arXiv preprint arXiv:2410.13300*, 2024. 85

Justin Solomon, Raif Rustamov, Leonidas Guibas, and Adrian Butscher. Wasserstein propagation for semi-supervised learning. In *International Conference on Machine Learning*, pages 306–314. PMLR, 2014. 139, 143, 329, 330

Justin Solomon, Fernando De Goes, Gabriel Peyré, Marco Cuturi, Adrian Butscher, Andy Nguyen, Tao Du, and Leonidas Guibas. Convolutional Wasserstein distances: Efficient optimal transportation on geometric domains. *ACM Transactions on Graphics (ToG)*, 34(4):1–11, 2015. 139, 143, 144, 329, 330, 340

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a. 18, 358

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, 2019. 245

Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. In *Advances in Neural Information Processing Systems*, pages 12438–12448, 2020. 245

Yang Song and Diederik P. Kingma. How to train your energy-based models. *arXiv preprint arXiv:2101.03288*, 2021. 119

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021b. 2, 4, 16, 17, 18, 86, 91, 93, 103, 105, 108, 110, 115, 117, 118, 119, 130, 137, 139, 141, 147, 199, 205, 221, 233, 234, 271, 276, 278, 336, 358

Sanvesh Srivastava, Cheng Li, and David B Dunson. Scalable Bayes via barycenter in Wasserstein space. *The Journal of Machine Learning Research*, 19(1):312–346, 2018. 137, 146, 338

Matthew Staib, Sebastian Claici, Justin M Solomon, and Stefanie Jegelka. Parallel streaming Wasserstein barycenters. *Advances in Neural Information Processing Systems*, 2017. 144

Vincent Stimper, David Liu, Andrew Campbell, Vincent Berenz, Lukas Ryll, Bernhard Schölkopf, and José Miguel Hernández-Lobato. normflows: A pytorch package for normalizing flows. *Journal of Open Source Software*, 8(86):5361, 2023. 200, 201

Gabriel Stoltz, Mathias Rousset, et al. *Free energy computations: A mathematical perspective*. World Scientific, 2010. 6, 21, 103, 115

Gilbert Strang. On the construction and comparison of difference schemes. *SIAM Journal on Numerical Analysis*, 5(3):506–517, 1968. 74, 174

Daniel W Stroock and SR Srinivasa Varadhan. *Multidimensional diffusion processes*, volume 233. Springer Science & Business Media, 1997. 317

Zhengyu Su, Yalin Wang, Rui Shi, Wei Zeng, Jian Sun, Feng Luo, and Xianfeng Gu. Optimal mass transport for shape matching and comparison. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37 (11):2246–2259, 2015. 137

Jingtong Sun, Julius Berner, Lorenz Richter, Marius Zeinhofer, Johannes Müller, Kamyar Azizzadenesheli, and Anima Anandkumar. Dynamical measure transport and neural PDE solvers for sampling. *arXiv preprint arXiv:2407.07873*, 2024. 23, 132, 273

Ruoyu Sun, Tiantian Fang, and Alexander Schwing. Towards a better global loss landscape of gans. *Advances in Neural Information Processing Systems*, pages 10186–10198, 2020. 367

Robert H. Swendsen and Jian-Sheng Wang. Replica monte carlo simulation of spin-glasses. *Phys. Rev. Lett.*, 57:2607–2609, 1986. 15, 85, 121

Saifuddin Syed, Vittorio Romaniello, Trevor Campbell, and Alexandre Bouchard-Cote. Parallel tempering on optimized paths. In *International Conference on Machine Learning*, pages 10033–10042. PMLR, 2021. 86, 102, 112, 283

Saifuddin Syed, Alexandre Bouchard-Côté, George Deligiannidis, and Arnaud Doucet. Non-reversible parallel tempering: A scalable highly parallel mcmc scheme. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 84(2):321–350, 2022. 15, 86, 102, 103, 112, 121, 122, 283

Saifuddin Syed, Alexandre Bouchard-Côté, Kevin Chern, and Arnaud Doucet. Optimised annealed sequential monte carlo samplers. *arXiv preprint arXiv:2408.12057*, 2024. 85, 102, 112, 122

Ying Tai, Rui Xie, Chen Zhao, Kai Zhang, Zhenyu Zhang, Jun Zhou, and Jian Yang. Addsr: Accelerating diffusion-based blind super-resolution with adversarial diffusion distillation. *Pattern Recognition*, page 113012, 2026. 358, 359

Ella Tamir, Martin Trapp, and Arno Solin. Transport with support: Data-conditional diffusion bridges. *Transactions on Machine Learning Research*, 2023. 148

Kirill Tamogashev and Nikolay Malkin. Data-to-energy stochastic dynamics. In *2nd edition of Frontiers in Probabilistic Inference: Learning meets Sampling*, 2025. 22

Molei Tao. Explicit symplectic approximation of nonseparable Hamiltonians: Algorithm and long time performance. *Physical Review E*, 94(4):043303, 2016. 174

Nicholas G Tawn, Gareth O Roberts, and Jeffrey S Rosenthal. Weight-preserving simulated tempering. *Statistics and Computing*, 30(1):27–41, 2020. 103

Alexis MH Teter, Wenqing Wang, and Abhishek Halder. Schrödinger bridge with quadratic state cost is exactly solvable. *arXiv preprint arXiv:2406.00503*, 2024. 151

Ines Thiele, Neil Swainston, Ronan MT Fleming, Andreas Hoppe, Swagatika Sahoo, Maike K Aurich, Hulda Haraldsdottir, Monica L Mo, Ottar Rolfsson, Miranda D Stobbe, et al. A community-driven global reconstruction of human metabolism. *Nature biotechnology*, 31(5):419–425, 2013. 5, 71

James Thornton, Louis Béthune, Ruixiang ZHANG, Arwen Bradley, Preetum Nakkiran, and Shuangfei Zhai. Controlled generation with distilled diffusion energy models and sequential monte carlo. In *International Conference on Artificial Intelligence and Statistics*, 2025. 119, 128, 130, 132, 283

Yifeng Tian, Nishant Panda, and Yen Ting Lin. Liouville flow importance sampler. In *International Conference on Machine Learning*, pages 48186–48210. PMLR, 2024. 23

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2023. 31, 147, 361

Alexander Y. Tong, Nikolay Malkin, Kilian Fatras, Lazar Atanackovic, Yanlei Zhang, Guillaume Huguet, Guy Wolf, and Yoshua Bengio. Simulation-free Schrödinger bridges via score and flow matching. In *International Conference on Artificial Intelligence and Statistics*, pages 1279–1287. PMLR, 2024. 31, 147

Ryan Turner, Jane Hung, Eric Frank, Yunus Saatchi, and Jason Yosinski. Metropolis-hastings generative adversarial networks. In *International Conference on Machine Learning*, pages 6345–6353. PMLR, 2019. 91

Belinda Tzen and Maxim Raginsky. Theoretical guarantees for sampling and inference in generative models with latent diffusions. In *Conference on Learning Theory*, pages 3084–3114. PMLR, 2019. 22, 91, 104, 105, 233

Masatoshi Uehara, Yulai Zhao, Tommaso Biancalani, and Sergey Levine. Understanding reinforcement learning-based fine-tuning of diffusion models: A tutorial and review. *arXiv preprint arXiv:2407.13734*, 2024. 128

Gabriel Valiente. *Algorithms on Trees and Graphs*, volume 112. Springer, 2002. 315, 316

Francisco Vargas, Pierre Thodoroff, Austen Lamacraft, and Neil Lawrence. Solving Schrödinger bridges via maximum likelihood. *Entropy*, 23(9):1134, 2021. 137, 144

Francisco Vargas, Will Sussman Grathwohl, and Arnaud Doucet. Denoising diffusion samplers. In *International Conference on Learning Representations*, 2023a. 22, 36, 37, 86, 91, 93, 99, 104, 105, 106, 110, 199, 200, 202, 233, 234, 236, 239

Francisco Vargas, Andrius Ovsianas, David Fernandes, Mark Girolami, Neil D Lawrence, and Nikolas Nüsken. Bayesian learning via neural schrödinger–Föllmer flows. *Statistics and Computing*, 33(1):3, 2023b. 22, 99, 104, 105, 233

Francisco Vargas, Teodora Reu, and Anna Kerekes. Expressiveness remarks for denoising diffusion based sampling. In *Fifth Symposium on Advances in Approximate Bayesian Inference*, 2023c. 91

Francisco Vargas, Shreyas Padhy, Denis Blessing, and Nikolas Nüsken. Transport meets Variational Inference: Controlled Monte Carlo Diffusions. In *International Conference on Learning Representations*, 2024. 22, 86, 91, 99, 104, 110, 236, 243, 257

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017. 336

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7): 1661–1674, 2011. 91, 105, 108, 119, 139, 141

Martin J. Wainwright and Michael I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1:1–305, 2008. 7, 91, 104

Fu-Yun Wang, Ling Yang, Zhaoyang Huang, Mengdi Wang, and Hongsheng Li. Rectified diffusion: Straightness is not your need in rectified flow. In *International Conference on Learning Representations*, 2025a. 20, 358

Jianyi Wang, Kelvin CK Chan, and Chen Change Loy. Exploring clip for assessing the look and feel of images. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 2555–2563, 2023. 368

Jianyi Wang, Zongsheng Yue, Shangchen Zhou, Kelvin CK Chan, and Chen Change Loy. Exploiting diffusion prior for real-world image super-resolution. *International Journal of Computer Vision*, 132(12):5929–5949, 2024a. 358, 359

Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European conference on computer vision (ECCV) workshops*, pages 63–079, 2018. 358, 359

Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *International Conference on Computer Vision Workshops (ICCVW)*, 2021. 358, 359, 364, 368, 377

Yan Wang, Ling Guo, Hao Wu, and Tao Zhou. Energy-based diffusion generator for efficient sampling of boltzmann distributions. *Neural Networks*, page 108126, 2025b. 22

Yufei Wang, Wenhan Yang, Xinyuan Chen, Yaohui Wang, Lanqing Guo, Lap-Pui Chau, Ziwei Liu, Yu Qiao, Alex C Kot, and Bihan Wen. Sinsr: diffusion-based image super-resolution in a single step. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 25796–25805, 2024b. 358, 359

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 368

Pengxu Wei, Ziwei Xie, Hannan Lu, Zongyuan Zhan, Qixiang Ye, Wangmeng Zuo, and Liang Lin. Component divide-and-conquer for real-world image super-resolution. In *European conference on computer vision*, pages 101–117. Springer, 2020. 368

Li K. Wenliang and Heishiro Kanagawa. Blindness of score-based methods to isolated components and mixing proportions. In *NeurIPS 2021 Workshop Your Model Is Wrong: Robustness and Misspecification in Probabilistic Modeling*, 2021. 132

Christopher Williams, Andrew Campbell, Arnaud Doucet, and Saifuddin Syed. Score-optimal diffusion schedules. In *Advances in Neural Information Processing Systems*, pages 107960–107983, 2024. 102

Ulli Wolff. Critical slowing down. *Nuclear Physics B - Proceedings Supplements*, 17:93–102, September 1990. 81

Dongyeop Woo and Sungsoo Ahn. Iterated energy-based flow matching for sampling from boltzmann densities. *arXiv preprint arXiv:2408.16249*, 2024. 24

Dawn Woodard, Scott Schmidler, and Mark Huber. Sufficient Conditions for Torpid Mixing of Parallel and Simulated Tempering. *Electronic Journal of Probability*, 14:780 – 804, 2009. 103

Hao Wu, Jonas Köhler, and Frank Noé. Stochastic normalizing flows. *Advances in Neural Information Processing Systems*, pages 5933–5944, 2020. 104

Luhuan Wu, Brian L. Trippe, Christian A Naesseth, John Patrick Cunningham, and David Blei. Practical and asymptotically exact conditional sampling in diffusion models. In *Advances in Neural Information Processing Systems*, 2023. 128

Luhuan Wu, Yi Han, Christian A. Naesseth, and John Patrick Cunningham. Reverse diffusion sequential monte carlo samplers. In *Advances in Neural Information Processing Systems*, 2025a. 25

Rongyuan Wu, Lingchen Sun, Zhiyuan Ma, and Lei Zhang. One-step effective diffusion network for real-world image super-resolution. *Advances in Neural Information Processing Systems*, pages 92529–92553, 2024a. 358, 359, 364, 368

Rongyuan Wu, Tao Yang, Lingchen Sun, Zhengqiang Zhang, Shuai Li, and Lei Zhang. Seesr: Towards semantics-aware real-world image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 25456–25467, 2024b. 358, 359

Xue Wu, Jingwei Xin, Zhijun Tu, Jie Hu, Jie Li, Nannan Wang, and Xinbo Gao. One-step diffusion-based real-world image super-resolution with visual perception distillation. *arXiv preprint arXiv:2506.02605*, 2025b. 358, 359, 378

Shahar Yadin, Noam Elata, and Tomer Michaeli. Classification diffusion models: Revitalizing density ratio estimation. In *Advances in Neural Information Processing Systems*, 2024. 119

Sidi Yang, Tianhe Wu, Shuwei Shi, Shanshan Lao, Yuan Gong, Mingdeng Cao, Jiahao Wang, and Yujiu Yang. Maniqa: Multi-dimension attention network for no-reference image quality assessment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1191–1200, 2022. 368

Tao Yang, Rongyuan Wu, Peiran Ren, Xuansong Xie, and Lei Zhang. Pixel-aware stable diffusion for realistic image super-resolution and personalized stylization. In *European conference on computer vision*, pages 74–91. Springer, 2024. 358

Yuling Yao, Aki Vehtari, and Andrew Gelman. Stacking for Non-mixing Bayesian Computations: The Curse and Blessing of Multimodal Posteriors. *Journal of Machine Learning Research*, 23(79):1–45, 2022. 81

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6613–6623, 2024. 358

Hanlin Yu, Arto Klami, Aapo Hyvarinen, Anna Korba, and Omar Chehab. Density ratio estimation with conditional probability paths. In *International Conference on Machine Learning*. PMLR, 2025. 119, 130, 272

Zongsheng Yue, Jianyi Wang, and Chen Change Loy. Resshift: Efficient diffusion model for image super-resolution by residual shifting. *Advances in Neural Information Processing Systems*, pages 13294–13307, 2023. 358, 359, 368

Zongsheng Yue, Jianyi Wang, and Chen Change Loy. Efficient diffusion model for image restoration by residual shifting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 358, 359

Emilio Zappa, Miranda Holmes-Cerfon, and Jonathan Goodman. Monte Carlo on manifolds: sampling densities and integrating functions. *Communications on Pure and Applied Mathematics*, 71(12):2609–2647, 2018. 71, 75, 79

Aiping Zhang, Zongsheng Yue, Renjing Pei, Wenqi Ren, and Xiaochun Cao. Degradation-guided one-step image super-resolution with diffusion priors. *arXiv preprint arXiv:2409.17058*, 2024a. 358, 359, 364, 365, 368

Fengzhe Zhang, Jiajun He, Laurence I Midgley, Javier Antorán, and José Miguel Hernández-Lobato. Efficient and unbiased sampling of boltzmann distributions via consistency models. *arXiv preprint arXiv:2409.07323*, 2024b. 123, 128

Fengzhe Zhang, Laurence Illing Midgley, and José Miguel Hernández-Lobato. Efficient and unbiased sampling from boltzmann distributions via variance-tuned diffusion models. *Transactions on Machine Learning Research*, 2025. J2C Certification. 123, 128

Leo Zhang, Peter Potaptchik, Jiajun He, Yuanqi Du, Arnaud Doucet, Francisco Vargas, Hai-Dang Dau, and Saifuddin Syed. Accelerated parallel tempering via neural transports. In *International Conference on Learning Representations*, 2026. 24, 25, 39, 123, 125, 128, 130, 132

Lin Zhang, Lei Zhang, and Alan C Bovik. A feature-enriched completely blind image quality evaluator. *IEEE Transactions on Image Processing*, 24(8):2579–2591, 2015. 368

Mingtian Zhang, Oscar Key, Peter Hayes, David Barber, Brooks Paige, and Francois-Xavier Briol. Towards healing the blindness of score matching. In *NeurIPS 2022 Workshop on Score-Based Methods*, 2022. 40, 132

Qinsheng Zhang and Yongxin Chen. Path Integral Sampler: a stochastic control approach for sampling. In *International Conference on Learning Representations*, 2022. 22, 37, 86, 91, 99, 104, 105, 106, 110, 233, 236, 257

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018. 359, 364, 368

Xinwei Zhang, Zhiqiang Tan, and Zhijian Ou. Persistently trained, diffusion-assisted energy-based models. *Stat*, 12(1):e625, 2023. 109, 119, 250

Jianbin Zheng, Minghui Hu, Zhongyi Fan, Chaoyue Wang, Changxing Ding, Dacheng Tao, and Tat-Jen Cham. Trajectory consistency distillation: Improved latent consistency distillation by semi-linear consistency function with trajectory mapping. *arXiv preprint arXiv:2402.19159*, 2024. 361

Kaiwen Zheng, Yuji Wang, Qianli Ma, Huayu Chen, Jintao Zhang, Yogesh Balaji, Jianfei Chen, Ming-Yu Liu, Jun Zhu, and Qinsheng Zhang. Large scale diffusion distillation via score-regularized continuous-time consistency. *arXiv preprint arXiv:2510.08431*, 2025. 358

Yan Zhou, Adam M. Johansen, and John A.D. Aston. Toward automatic model comparison: An adaptive sequential monte carlo approach. *Journal of Computational and Graphical Statistics*, 25(3):701–726, 2016. 85, 102, 112, 283

Yaxuan Zhu, Jianwen Xie, Ying Nian Wu, and Ruiqi Gao. Learning energy-based models by cooperative diffusion recovery likelihood. In *International Conference on Learning Representations*, 2024. 109, 119, 249

Part I

On Fundamental Challenges in Sampling

Chapter 2

Unbiased constrained sampling with Self-Concordant Barrier Hamiltonian Monte Carlo

2.1 Introduction

Markov Chain Monte Carlo (MCMC) methods is one of the primary algorithmic approaches to obtain approximate samples from a target distribution π. They have been successively applied over these past decades in a large panel of practical settings, [Liu and Liu, 2001]. In particular, gradient-based MCMC methods have shown their efficiency and robustness in high-dimensional settings, and come nowadays with strong theoretical guarantees, [Dalalyan, 2017, Durmus and Moulines, 2017]. However, they still struggle in facing the case where the target distribution is supported on a *constrained* subset M of $\mathbb{R}^d$, [Gelfand et al., 1992, Pakman and Paninski, 2014, Lan and Shahbaba, 2016]. Yet, this problem appears in various fields; see e.g., [Morris, 2002, Lewis et al., 2012, Thiele et al., 2013] with some important applications in computational statistics and biology.

Drawing samples from such distributions is indeed a challenging problem that has been intensively studied in the literature, [Dyer and Frieze, 1991, Lovász and Simonovits, 1993, Lovász and Kannan, 1999, Lovász and Vempala, 2007, Brubaker et al., 2012, Cousins and Vempala, 2018, Pakman and Paninski, 2014, Lan and Shahbaba, 2016, Bubeck et al., 2015]. In particular, some recent extensions of the popular Metropolis-Hastings (MH) algorithm to constrained spaces consist in designing proposals based on dynamics for which π is invariant, [Zappa et al., 2018, Lelièvre et al., 2019, 2022]. In practice, the corresponding solutions are numerically integrated based on implicit and symplectic schemes, which however come with additional difficulties. As first observed by Zappa et al. [2018], an additional "involution checking step" in the usual MH filter is necessary to ensure that the resulting Markov kernel admits π as a stationary distribution.

In this paper, we aim at generalizing this family of methods by taking into account the geometry of the constrained subspace, building on the Riemannian Manifold Hamiltonian Monte Carlo (RMHMC) algorithm introduced by Girolami and Calderhead [2011]. Similarly to HMC, [Duane et al., 1987, Neal et al., 2011, Betancourt, 2017], RMHMC aims to target a positive distribution π on $\mathbb{R}^d$ and relies on the integration of a canonical Hamiltonian equation. In contrast, RMHMC incorporates some *geometrical* information in the definition of the Hamiltonian via a Riemannian metric $\mathfrak{g}$ on $\mathbb{R}^d$. When considering applications of RMHMC to a convex, open and bounded subset M, a natural choice for this metric is the Hessian metric associated with a self-concordant barrier on M, [Nesterov and Nemirovskii, 1994], as suggested by Kook et al. [2022]. We adopt here the same approach and now focus on a constrained subspace M which is assumed to be equipped with an appropriately designed "self-concordant" metric, [Nesterov and Nemirovskii, 1994].

Given this setting, we propose the BHMC (Barrier HMC) algorithm. To the best of our knowledge, this is **the first version of RMHMC incorporating an "involution checking step"** to assess the issue of asymptotic bias arising from the use of implicit integration methods to solve the Hamiltonian dynamics. In this work, we focus on the *numerical* version of BHMC (n-BHMC), for which we provide a numerical implementation. We refer to Appendix A.4 for an introduction to the ideal version of BHMC, called *continuous* BHMC (c-BHMC), where it is assumed that we have access to the continuous Hamiltonian dynamics but which cannot be computed in practice. For each algorithm, we rigorously prove that the associated Markov chain preserves π.

The rest of the paper is organized as follows. In Section 2.2, we introduce our sampling framework and review some background on self-concordance and Riemannian geometry. In Section 2.3, we present n-BHMC, and derive theoretical results for this algorithm in Section 2.4. We review related works in Section 2.5 and provide numerical experiments for n-BHMC in Section 2.6.

Notation. For any $f \in \mathrm{C}^3(\mathbb{R}^d, \mathbb{R})$, we denote by Df (and $D^2 f$) the Jacobian (resp. the Hessian) of f. For any $\mathrm{A} \in \mathbb{R}^{d\times d}$ and any $(x, y) \in \mathbb{R}^d \times \mathbb{R}^d$, we denote by $\mathrm{A} : D^3 f(x)$ the vector $(\mathrm{Tr}(\mathrm{A}^\top \{D^3 f(x)\}_i))_{i\in[d]} \in \mathbb{R}^d$, and define $D^3 f(x)[x, y]$ as the vector $x^\top D^3 f(x) y \in \mathbb{R}^d$. For any positive-definite matrix $\mathrm{A} \in \mathbb{R}^{d\times d}$, $\langle \cdot, \cdot \rangle_{\mathrm{A}}$ stands for the scalar product induced by A on $\mathbb{R}^d$, defined by $\langle x, y \rangle_{\mathrm{A}} = \langle x, \mathrm{A}y \rangle$. The "momentum reversal" operator $s : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d \times \mathbb{R}^d$ is defined for any $(x, p) \in \mathbb{R}^d \times \mathbb{R}^d$ by $s(x, p) = (x, -p)$. Let E, F be two sets and $h : \mathsf{E} \to 2^{\mathsf{F}}$, where 2^{F} is the set of sets of F. We say that h is a set-valued map. Note that any map $g : \mathsf{E} \to \mathsf{F}$ can be extended to a set-valued map by identifying, for any $x \in \mathsf{E}$, $g(x)$ and $\{g(x)\}$. Let $f : \mathsf{F} \to 2^{\mathsf{G}}$. We define $f \circ h : \mathsf{E} \to 2^{\mathsf{G}}$ for any $x \in \mathsf{E}$ by $f \circ h(x) = \cup_{y\in h(x)} f(y)$, where by convention $\cup_\emptyset = \emptyset$. For any topological space X, we denote $\mathcal{B}(\mathsf{X})$ its Borel sets. For any probability measure $\mu \in \mathscr{P}(\mathsf{X})$ and measurable map $\varphi : \mathsf{X} \to \mathsf{Y}$, we denote $\varphi_{\#}\mu \in \mathscr{P}(\mathsf{Y})$ the pushforward of μ by φ. In general, we will equivalently denote by z or (x, p) a state of the Hamiltonian system.

2.2 Setting and Background

In this paper, we consider an open subset $\mathsf{M} \subset \mathbb{R}^d$, and we aim at sampling from a target distribution π given for any $x \in \mathsf{M}$ by

$$\mathrm{d}\pi(x)/\mathrm{d}x = \exp[-V(x)]/Z \,, \tag{2.1}$$

where $V \in \mathrm{C}^2(\mathsf{M}, \mathbb{R})$ and $Z = \int_{\mathsf{M}} \exp[-V(x)]\mathrm{d}x$. We view M as an embedded d-dimensional submanifold of $\mathbb{R}^d$, equipped with a metric $\mathfrak{g}$, satisfying the following assumptions.

A0. M *is an open convex bounded subset of* $\mathbb{R}^d$.

A1. *There exists* ϕ*, a* α*-regular and* ν*-self-concordant barrier on* M *such that* $\mathfrak{g} = D^2\phi$.

We provide in Section 2.2.1 basic Riemannian facts along with the definition of the Hamiltonian dynamics of RMHMC and introduce self-concordance and α-regularity in Section 2.2.2.

2.2.1 Riemannian Manifold Hamiltonian dynamics

Basics on Riemannian geometry. Let M be a d-dimensional smooth manifold, endowed with a metric $\mathfrak{g}$. Denoting by $\mathrm{d}x$ a dual coframe, [Lee, 2006, Lemma 3.2.], we recall that the *Riemannian volume element* corresponding to $(\mathsf{M}, \mathfrak{g})$ is given in local coordinates by

$$\mathrm{dvol}_{\mathsf{M}}(x) = \sqrt{\det(\mathfrak{g})}\mathrm{d}x \,. \tag{2.2}$$

We denote by $\mathrm{T}^\star_x\mathsf{M}$ the dual of the tangent space at $x \in \mathsf{M}$, *i.e.*, the cotangent space. For any $x \in \mathsf{M}$, $\mathrm{T}^\star_x\mathsf{M}$ is a vector space naturally endowed with the scalar product $\langle \cdot, \cdot \rangle_{\mathfrak{g}(x)^{-1}}$, [Mok, 1977]. We recall that the cotangent bundle $\mathrm{T}^\star\mathsf{M}$ is defined by $\mathrm{T}^\star\mathsf{M} = \sqcup_{x\in\mathsf{M}}\{x\} \cup \mathrm{T}^\star_x\mathsf{M}$. This set is a $2d$-dimensional manifold endowed with a Riemannian metric $\mathfrak{g}^\star$ given by Mok [1977], inherited from $\mathfrak{g}$. With this metric, the volume form on $\mathrm{T}^\star\mathsf{M}$ does not depend on $\mathfrak{g}$ anymore, and satisfies for any $(x, p) \in \mathrm{T}^\star\mathsf{M}$

$$\mathrm{dvol}_{\mathrm{T}^\star\mathsf{M}}(x, p) = \mathrm{d}x\mathrm{d}p \,,$$

where $\mathrm{d}x\mathrm{d}p$ is a dual coframe for $\mathrm{T}^\star\mathsf{M}$.

Therefore, under **A**0, identifying $\mathrm{T}^\star\mathsf{M}$ with $\mathsf{M} \times \mathbb{R}^d$, the volume form on $\mathrm{T}^\star\mathsf{M}$ induced by $\mathfrak{g}^\star$ is the Lebesgue measure of $\mathbb{R}^d \times \mathbb{R}^d$ restricted to $\mathrm{T}^\star\mathsf{M}$. Despite adopting at first a Riemannian perspective on M, we actually recover the Euclidean setting on $\mathrm{T}^\star\mathsf{M}$, which motivates us to consider $\mathrm{T}^\star\mathsf{M}$ as our sampling space. We refer to Appendix A.2 for details on the cotangent bundle and its metric $\mathfrak{g}^\star$.

Hamiltonian dynamics. Note that with the previously introduced notation, π can be expressed as

$$\mathrm{d}\pi/\mathrm{dvol}_{\mathsf{M}}(x) = \exp[-V(x) - \tfrac{1}{2}\log(\det(\mathfrak{g}(x)))]/Z \ . \tag{2.3}$$

This motivates the introduction of the following Hamiltonian on $\mathrm{T}^\star\mathsf{M}$

$$H(x,p) = V(x) + \tfrac{1}{2}\log\left(\det \mathfrak{g}(x)\right) + \tfrac{1}{2}\|p\|^2_{\mathfrak{g}(x)^{-1}} \ . \tag{2.4}$$

In this definition, we notably take into account the scalar product $\langle\cdot,\cdot\rangle_{\mathfrak{g}(x)^{-1}}$ on $\mathrm{T}^\star_x\mathsf{M}$. Finally, we consider the joint distribution $\bar{\pi}$ on $\mathrm{T}^\star\mathsf{M}$

$$\mathrm{d}\bar{\pi}(x,p) = (1/\bar{Z})\exp[-H(x,p)]\mathrm{dvol}_{\mathrm{T}^\star\mathsf{M}}(x,p) \ , \tag{2.5}$$

where $\bar{Z} = \int_{\mathrm{T}^\star\mathsf{M}} \exp[-H(x,p)]\mathrm{dvol}_{\mathrm{T}^\star\mathsf{M}}(x,p)$, for which the first marginal is π. Indeed, for any $\varphi \in \mathrm{C}(\mathsf{M},\mathbb{R})$, we have

$$\int_{\mathrm{T}^\star\mathsf{M}} \varphi(x)\mathrm{d}\bar{\pi}(x,p) = \int_{\mathrm{T}^\star\mathsf{M}} \varphi(x)\mathrm{d}\pi(x)\mathrm{N}_x(p;0,\mathrm{Id})\mathrm{d}p = \int_{\mathsf{M}} \varphi(x)\mathrm{d}\pi(x) \ , \tag{2.6}$$

where we denote by $\mathrm{N}_x(0,\mathrm{I}_d)$ the centered standard Gaussian distribution w.r.t. $\|\cdot\|_{\mathfrak{g}(x)^{-1}}$. The Hamiltonian dynamics for H is given by the coupled Ordinary Differential Equations (ODEs)

$$\dot{x} = \partial_p H(x,p) \ , \qquad \dot{p} = -\partial_x H(x,p) \ , \tag{2.7}$$

where the derivatives of H can be computed explicitly as

$$\partial_p H(x,p) = \mathfrak{g}(x)^{-1}p \ , \qquad \partial_x H(x,p) = -\tfrac{1}{2}D\mathfrak{g}(x)[\mathfrak{g}(x)^{-1}p, \mathfrak{g}(x)^{-1}p] + L(x) \ , \tag{2.8}$$

where $L(x) = \nabla V(x) + \frac{1}{2}\mathfrak{g}(x)^{-1} : D\mathfrak{g}(x)$.

2.2.2 Self-concordance and regularity

Until now, we have considered an arbitrary Riemannian metric $\mathfrak{g}$. In the rest of this work, we focus on metrics given by Hessian of self-concordant barriers.

Self-concordance. We first introduce self-concordant barriers, a family of smooth convex functions which are well-suited for minimization by the Newton method.

Definition 1 (Nesterov and Nemirovskii [1994]). *Let U be a non-empty open convex domain in $\mathbb{R}^d$. A function $\phi : \mathsf{U} \to \mathbb{R}$ is said to be a ν-self-concordant barrier (with $\nu \geq 1$) on U if it satisfies: (a) $\phi \in \mathrm{C}^3(\mathsf{U},\mathbb{R})$ and ϕ is convex, (b) $\phi(x) \to +\infty$ as $x \to \partial\mathsf{U}$, (c) $|D^3\phi(x)[h,h,h]| \leq 2\|h\|^3_{\mathfrak{g}(x)}$, (d) $|D\phi(x)[h]|^2 \leq \nu\|h\|^2_{\mathfrak{g}(x)}$, for any $x \in \mathsf{M}$, $h \in \mathbb{R}^d$, where $\mathfrak{g}(x) = D^2\phi(x)$.*

Balls for $\|\cdot\|_{\mathfrak{g}(x)}$, called Dikin ellipsoids, are key for the study of self-concordance, see Appendix A.3.

Regularity. The property of α-regularity for some $\alpha \geq 1$ is shared by many self-concordant barriers, including logarithmic and quadratic programming barriers. It ensures stability for interior point polynomial-time methods. Definition and properties of α-regularity are recalled in Appendix A.3.

Example of the polytope. Let us assume that M is the polytope $\mathsf{M} = \{x : \mathrm{A}x < b\}$, . We endow it with the Riemannian metric $\mathfrak{g}(x) = D^2\phi(x)$ where $\phi : \mathsf{M} \to \mathbb{R}$ is the logarithmic barrier given for any $x \in \mathsf{M}$ by $\phi(x) = -\sum_{i=1}^m \log\left(b_i - \mathrm{A}_i^\top x\right)$. The barrier ϕ is both a m-self-concordant barrier and a 2-regular function, [Nesterov and Nemirovski, 1998, page 3]. Moreover, we have for any $x \in \mathsf{M}$, $\mathfrak{g}(x) = \mathrm{A}^\top S(x)^{-2}\mathrm{A}$, where $S(x) = \mathrm{Diag}\left(b_i - \mathrm{A}_i^\top x\right)_{i\in[m]}$. We provide in Figure 2.1 an illustration of this barrier.

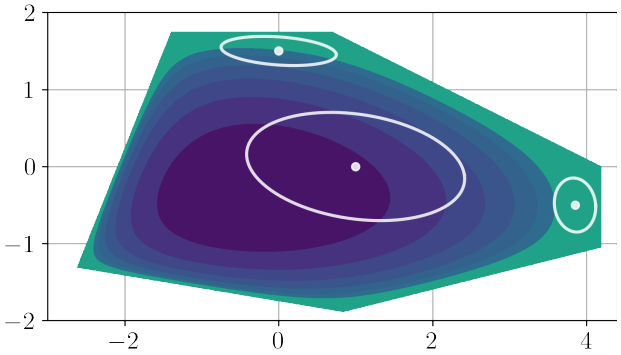


Figure 2.1: A polytope $\mathsf{M} \subset \mathbb{R}^2$ with three Dikin ellipsoids $\{y \in \mathbb{R}^d : (y-x)^\top\mathfrak{g}(x)(y-x) < 1\}$ (*white circles*) derived from its logarithmic barrier for three positions $x \in \mathsf{M}$ (*white dots*).

2.3 The n-BHMC algorithm

In practice, it is not possible to *exactly* compute the Riemannian Hamiltonian dynamics (2.7). We thus introduce a *numerical* version of BHMC (n-BHMC), in which we replace the continuous ODE integration by a symplectic numerical scheme. We first define the Hamiltonian integrators used in n-BHMC in Section 2.3.1 and provide details on the different steps of our algorithm in Section 2.3.2.

2.3.1 Hamiltonian integrators of n-BHMC

In the same spirit as Shahbaba et al. [2014], we first rewrite the Hamiltonian H given in (2.4) as $H = H_1 + H_2$, where we highlight the non separable aspect of H in H_2, *i.e.*, for any $(x, p) \in \mathrm{T}^\star\mathsf{M}$

$$H_1(x,p) = V(x) + \tfrac{1}{2}\log(\det \mathfrak{g}(x)) \;, \quad H_2(x,p) = \tfrac{1}{2}\|p\|^2_{\mathfrak{g}(x)^{-1}} \;. \tag{2.9}$$

Therefore, we have

$$\partial_x H_1(x,p) = \nabla V(x) + \tfrac{1}{2}\mathfrak{g}(x)^{-1} : D\mathfrak{g}(x) \;, \qquad \partial_p H_1(x,p) = 0 \;, \tag{2.10}$$

$$\partial_x H_2(x,p) = -\tfrac{1}{2} D\mathfrak{g}(x)[\mathfrak{g}(x)^{-1}p, \mathfrak{g}(x)^{-1}p] \;, \qquad \partial_p H_2(x,p) = \mathfrak{g}(x)^{-1}p \;. \tag{2.11}$$

Leveraging the separable structure of H_1, we now define an implicit scheme to discretize the Hamiltonian dynamics (2.7) by *splitting* it into the Hamiltonian dynamics (2.10) and (2.11). In the rest of this section, we consider some step-size $h \in \mathbb{R}$.

Explicit integrator of H_1. We first approximate the dynamics (2.10) on a step-size $h/2$ using a first-order Euler method [Hairer et al., 2006, Theorem VI.3.3.]. Since H_1 is separable, this integrator simply reduces to the map $\mathrm{S}_{h/2} : \mathrm{T}^\star\mathsf{M} \to \mathrm{T}^\star\mathsf{M}$ defined by

$$\mathrm{S}_{h/2}(x,p) = (x, p - \tfrac{h}{2}\partial_x H_1(x,p)) \;.$$

We have: (i) $\mathrm{S}_{h/2}(\mathrm{T}^\star\mathsf{M}) \subset \mathrm{T}^\star\mathsf{M}$, (ii) $\mathrm{S}_{h/2}$ is symplectic (we refer to Appendix A.6 for more details on symplecticity), (iii) $\mathrm{S}_{-h/2} \circ \mathrm{S}_{h/2} = \mathrm{Id}$ and (iv) $\mathrm{S}_{-h/2} = s \circ \mathrm{S}_{h/2} \circ s$. Note that $s \circ \mathrm{S}_{h/2}$ is an involution on $\mathrm{T}^\star\mathsf{M}$, which inherits from properties (i) and (ii) of $\mathrm{S}_{h/2}$.

Implicit integrator of H_2. Since H_2 is not separable, a common discretization scheme such as the Euler method is not symplectic anymore, which requires to use an implicit method instead. We thus approximate these dynamics on a step-size h with a symplectic second-order integrator denoted by G_h. For theoretical purposes, we focus on the Störmer-Verlet scheme, [Hairer et al., 2006, Theorem VI.3.4.], also known as the *generalized Leapfrog* integrator, which is widely used in geometric integration, [Girolami and Calderhead, 2011, Betancourt, 2013, Cobb et al., 2019, Brofos and Lederman, 2021a]. We refer to Appendix A.6 for a discussion on other numerical schemes. For any $z^{(0)} \in \mathrm{T}^\star\mathsf{M}$, $\mathrm{G}_h(z^{(0)}) \subset \mathrm{T}^\star\mathsf{M}$ consists of points $z^{(1)} = (x^{(1)}, p^{(1)})$ which are solution of

$$p^{(1/2)} = p^{(0)} - \tfrac{h}{2}\partial_x H_2(x^{(0)}, p^{(1/2)}) \;, \tag{2.12}$$

$$x^{(1)} = x^{(0)} + \tfrac{h}{2}[\partial_p H_2(x^{(0)}, p^{(1/2)}) + \partial_p H_2(x^{(1)}, p^{(1/2)})] \;, \tag{2.13}$$

$$p^{(1)} = p^{(1/2)} - \tfrac{h}{2}\partial_x H_2(x^{(1)}, p^{(1/2)}) \;. \tag{2.14}$$

As defined, there is no guarantee that (2.14) admits a (unique) solution. As a matter of fact, the integrator G_h can be seen as a set-valued map. Moreover, it is easy to check that: (i) $\mathrm{G}_h(\mathrm{T}^\star\mathsf{M}) \subset 2^{\mathrm{T}^\star\mathsf{M}}$, (ii) $\mathrm{G}_{-h} = s \circ \mathrm{G}_h \circ s$, since $\partial_p H_2(s(z)) = -\partial_p H_2(z)$ and $\partial_x H_2(s(z)) = \partial_x H_2(z)$, and (iii) if $|\mathrm{G}_h(z)| > 0$ then $z \in (\mathrm{G}_{-h} \circ \mathrm{G}_h)(z)$.

Implicit integrator of H. Relying on the integrators $\mathrm{S}_{h/2}$ and G_h previously defined, we now explain how we approximate the Hamiltonian dynamics (2.7) on a step-size h. We first define the set-valued maps $\mathrm{F}_h : \mathrm{T}^\star\mathsf{M} \to 2^{\mathrm{T}^\star\mathsf{M}}$ and $\mathrm{R}_h : \mathrm{T}^\star\mathsf{M} \to 2^{\mathrm{T}^\star\mathsf{M}}$ by

$$\mathrm{F}_h = \mathrm{G}_h \circ s \;, \quad \mathrm{R}_h = (s \circ \mathrm{S}_{h/2}) \circ \mathrm{F}_h \circ (s \circ \mathrm{S}_{h/2}) \;. \tag{2.15}$$

Using properties (ii) and (iii) of G_h, we have for any $z \in \mathrm{T}^\star\mathsf{M}$ such that $|\mathrm{F}_h(z)| > 0$, $z \in (\mathrm{F}_h \circ \mathrm{F}_h)(z)$, and for any $z' \in \mathrm{T}^\star\mathsf{M}$ such that $|(\mathrm{F}_h \circ s \circ \mathrm{S}_{h/2})(z')| > 0$, we have $z' \in (\mathrm{R}_h \circ \mathrm{R}_h)(z')$. Thus, F_h and R_h are *involutive* in the sense of set-valued maps. Note also that $s \circ \mathrm{R}_h = \mathrm{S}_{h/2} \circ \mathrm{G}_h \circ \mathrm{S}_{h/2}$ boils down to the *Strang splitting* [Strang, 1968] of the dynamics (2.7) and thus, is a symplectic scheme approximating the dynamics of Hamiltonian H on a step-size h.

Numerical integrators. In practice, we do not have access to F_h but approximate it with a *numerical map* Φ_h. For clarity's sake, we denote by $\mathrm{dom}_{\Phi_h} \subset \mathrm{T}^\star\mathsf{M}$ the domain of this integrator with $\Phi_h(\mathrm{dom}_{\Phi_h}) \subset \mathrm{T}^\star\mathsf{M}$. In practice, dom_{Φ_h} corresponds to the set of points for which the numerical integration of F_h outputs a solution. We also approximate R_h with the *numerical map* $\mathrm{R}_h^\Phi : (s \circ \mathrm{S}_{h/2})(\mathrm{dom}_{\Phi_h}) \subset \mathrm{T}^\star\mathsf{M} \to \mathrm{T}^\star\mathsf{M}$

$$\mathrm{R}_h^\Phi = (s \circ \mathrm{S}_{h/2}) \circ \Phi_h \circ (s \circ \mathrm{S}_{h/2}) \,. \tag{2.16}$$

Similarly to $s \circ \mathrm{R}_h$, $s \circ \mathrm{R}_h^\Phi$[1] approximates the dynamics of Hamiltonian H on a step-size h. In our experiments, we design Φ_h using a fixed-point solver with a given number of iterations following Brofos and Lederman [2021a,b]. We refer to Appendix A.11 for details on computations of Φ_h.

2.3.2 Steps of the algorithm

For any $z = (x, p) \in \mathrm{T}^\star\mathsf{M}$, we define on $\mathrm{T}^\star\mathsf{M}$ the norm $\|\cdot\|_z$ by

$$\|(x', p')\|_z = \|x'\|_{\mathfrak{g}(x)} + \|p'\|_{\mathfrak{g}(x)^{-1}} \,, \quad \forall (x', p') \in \mathrm{T}^\star\mathsf{M} \,, \tag{2.17}$$

where $\|\cdot\|_{\mathfrak{g}(x)^{-1}}$ is the canonical norm on $\mathrm{T}_x^\star\mathsf{M}$ induced by the Riemannian metric $\mathfrak{g}$ (see Section 2.2.1) and $\|\cdot\|_{\mathfrak{g}(x)}$ is the common norm used on M to study properties of self-concordance (see Section 2.2.2). As defined, this norm will be crucial to correct the bias of the numerical integration in n-BHMC, presented in Algorithm 1, for which we are now ready to detail the steps.

Steps 1 and 5: applying a momentum update. Assume that the current state at stage $n \geq 1$ is $(X_{n-1}, P_{n-1}) \in \mathrm{T}^\star\mathsf{M}$. Then, the momentum is first partially refreshed such that $\tilde{P}_n \leftarrow \sqrt{1-\beta} P_{n-1} + \sqrt{\beta} G_n$, where $G_n \sim \mathrm{N}_x(0, \mathrm{I}_d)$ is independent from P_{n-1}. This update is applied both at the beginning and the end of the iteration, similarly to Lelièvre et al. [2022].

Step 2: solving a discretized version of ODE (2.7). Starting from $(X'_n, P'_n) = (X_{n-1}, \tilde{P}_n)$, we now approximate the dynamics of H on a step-size h with the integrators presented in Section 2.3.1, while ensuring that the proposal state belongs to $\mathrm{T}^\star\mathsf{M}$. We proceed as follows:
1. We first run the *explicit* integrator $\mathrm{S}_{h/2}$ and compute $Z_n^{(0)} = (s \circ \mathrm{S}_{h/2})(X_{n-1}, \tilde{P}_n)$.
2. If $Z_n^{(0)} \notin \mathrm{dom}_{\Phi_h}$, then (X'_n, P'_n) is not updated and we directly go to Step 3. Otherwise, we define $Z_n^{(1)} = \Phi_h(Z_n^{(0)})$. To ensure the reversibility of this step, we then perform an "involution checking step", [Zappa et al., 2018, Lelièvre et al., 2019], highlighted in yellow in Algorithm 1, *i.e.*, we verify (a) $Z_n^{(1)} \in \mathrm{dom}_{\Phi_h}$ and (b) $Z_n^{(0)} = \Phi_h(Z_n^{(1)})$. In practice, we replace condition (b) by combining an explicit tolerance threshold η with the norm defined in (2.17) and accept the proposal if

$$\|Z_n^{(0)} - \Phi_h(Z_n^{(1)})\|_{Z_n^{(0)}} + \|Z_n^{(0)} - \Phi_h(Z_n^{(1)})\|_{\Phi_h(Z_n^{(1)})} \leq \eta \,. \tag{2.18}$$

We discuss the choice of this norm to perform the "involution checking step" in Appendix A.11. If both of these conditions are satisfied, we finally set $(X'_n, P'_n) = (s \circ \mathrm{S}_{h/2})(Z_n^{(1)})$, to ensure the reversibility of the update of (X'_n, P'_n). Otherwise, (X'_n, P'_n) is not updated and we go to Step 3.

Step 3: computing the acceptance filter. We denote by $a(x', p'|x, p)$ the acceptance probability to move from $(x, p) \in \mathrm{T}^\star\mathsf{M}$ to $(x', p') \in \mathrm{T}^\star\mathsf{M}$, which is given by

$$a(x', p'|x, p) = 1 \wedge \exp[-H(x', p') + H(x, p)] \,. \tag{2.19}$$

After Step 2, we perform a simple MH filter by accepting the proposal with probability $A_n = a(X'_n, P'_n | X_{n-1}, \tilde{P}_n)$ similarly to a classical HMC algorithm.

Step 4: applying momentum reversal. To ensure a move along the dynamics of H with step-size h, we reverse the momentum after the acceptance step. Indeed, if the acceptance filter is successful, then $(X_n, \hat{P}_n) = (s \circ \mathrm{R}_h^\Phi)(X_{n-1}, \tilde{P}_{n-1})$ approximates the Hamiltonian dynamics (2.7) on a step-size h starting at $(X_{n-1}, \tilde{P}_{n-1})$, as seen in Section 2.3.1. Otherwise, $(X_n, \hat{P}_n) = (X_{n-1}, -\tilde{P}_{n-1})$.

[1] Note that $s \circ \mathrm{R}_h^\Phi$ is a *function* and not a set-valued map.

Algorithm 1: n-BHMC with Momentum Refresh

Input: $(X_0, P_0) \in \mathrm{T}^\star\mathsf{M}$, $\beta \in (0,1]$, $N \in \mathbb{N}$, $h > 0$, $\eta > 0$, Φ_h with domain dom_{Φ_h}
Output: $(X_n, P_n)_{n\in[N]}$

1 **for** $n = 1, ..., N$ **do**
2 Step 1: $G_n \sim \mathrm{N}_x(0, \mathrm{I}_d)$, $\tilde{P}_n \leftarrow \sqrt{1-\beta}P_{n-1} + \sqrt{\beta}G_n$
3 Step 2: solving a discretized version of ODE (2.7)
4 $X'_n, P'_n \leftarrow X_{n-1}, \tilde{P}_n$; $X_n^{(0)}, P_n^{(0)} \leftarrow (s \circ \mathrm{S}_{h/2})(X_n, \tilde{P}_n)$
5 **if** $Z_n^{(0)} = (X_n^{(0)}, P_n^{(0)}) \in \mathrm{dom}_{\Phi_h}$ **then**
6 $Z_n^{(1)} = \Phi_h(Z_n^{(0)})$
7 $\mathrm{err}_0 = \|Z_n^{(0)} - \Phi_h(Z_n^{(1)})\|_{Z_n^{(0)}}$, $\mathrm{err}_1 = \|Z_n^{(0)} - \Phi_h(Z_n^{(1)})\|_{\Phi_h(Z_n^{(1)})}$ (if defined)
8 **if** $Z_n^{(1)} \in \mathrm{dom}_{\Phi_h}$ & $\mathrm{err}_0 + \mathrm{err}_1 \leq \eta$ **then** $X'_n, P'_n \leftarrow (s \circ \mathrm{S}_{h/2})(Z_n^{(1)})$
9 **end**
10 Step 3: $A_n \leftarrow \min(1, \exp[-H(X'_n, P'_n) + H(X_{n-1}, \tilde{P}_n)])$; $U_n \sim U[0,1]$
11 **if** $U_n \leq A_n$ **then** $\bar{X}_n, \bar{P}_n \leftarrow X'_n, P'_n$
12 **else** $\bar{X}_n, \bar{P}_n \leftarrow X_{n-1}, \tilde{P}_n$
13 Step 4: $X_n, \hat{P}_n \leftarrow s(\bar{X}_n, \bar{P}_n)$
14 Step 5: $G'_n \sim \mathrm{N}_x(0, \mathrm{I}_d)$, $P_n \leftarrow \sqrt{1-\beta}\hat{P}_n + \sqrt{\beta}G'_n$
15 **end**

2.3.3 About the usefulness of the "involution checking step"

In their paper, Kook et al. [2022] also incorporate self-concordance into RMHMC to sample from distributions supported on polytopes. To integrate the Hamiltonian dynamics, they propose to use a numerical version of the *Implicit Midpoint integrator* [Hairer et al., 2006, Theorem VI.3.5.], which shares the same theoretical properties as the Störmer-Verlet scheme. This results in CRHMC [Kook et al., 2022, Algorithm 3], whose main difference compared to n-BHMC is the lack of the "involution checking step" (Line 8 in Algorithm 1). Crucially, **Kook et al. [2022] assume that their implicit integrator admits a unique solution** given any starting point and any step-size h, which is then approximated by their numerical scheme. However, this statement may not be true in the self-concordant setting, as we explain below with a simple example. This explains why we always refer to the implicit integrator as a *set-valued map*. In their setting, the numerical integrator is not guaranteed to be involutive, which results in an asymptotic bias that the MH filter cannot solve.

In our paper, we do not make such an assumption on the Störmer-Verlet integrator, and consider a numerical solver which outputs *one* approximate solution to this implicit scheme. By doing so, our analysis is meant to be as close as possible to the practical implementation of RMHMC. Then, the "involution checking step" is critical to make the numerical integrator locally involutive, which enables us to derive rigorous reversibility results, see Theorem 3, and to solve the issue of asymptotic bias caused by implicit integration, see Section 2.6. We refer to Appendix A.10 for a detailed comparison between n-BHMC and CRHMC. We now present a simple setting where the assumption made by Kook et al. [2022] does not hold in the case of the Störmer-Verlet integrator, which is defined in (2.14).

Consider the 1-dimensional setting $\mathsf{M} = (-\infty, 0)$ with $\mathfrak{g} : x \mapsto 1/x^2$. Assume that $\beta = 1$. Let $h > 0$ and let $X_0 \in \mathsf{M}$ be the (position) state of the Markov chain at the beginning of the first iteration in n-BHMC. In this case, $\tilde{P}_0 \sim \mathrm{N}(0, 1/X_0^2)$ and we have $P_0^{(0)} = \tilde{P}_0 - (h/2)\partial_x H_1(X_0, \tilde{P}_0)$, where the function H_1 is defined in Section 2.3.1. Using (2.10), we have $P_0^{(0)} \sim \mathrm{N}(\mu_0, 1/X_0^2)$, where $\mu_0 = -(h/2)\partial_x H_1(X_0, \tilde{P}_0)$ only depends on X_0. Then, the implicit equation in $P_0^{(1/2)}$, given by the first equation of (2.14), reads as $P_0^{(1/2)} = P_0^{(0)} - (h/2)\partial_x H_2(X_0, P_0^{(1/2)})$. Using (2.11), it can be rewritten as a polynomial equation of degree 2 in $P_0^{(1/2)}$

$$\tfrac{h}{2} X_0 (P_0^{(1/2)})^2 + P_0^{(1/2)} - P_0^{(0)} = 0 \quad . \tag{2.20}$$

Denote $\Delta = 1 + 2hX_0P_0^{(0)}$. Then, (2.20) may admit 0,1 or 2 solutions depending on the sign of Δ. If $P_0^{(0)} \leq -1/(2hX_0)$, *i.e.*, $\Delta \geq 0$, then (2.20) admits 1 or 2 solutions. However, if h is small enough, only one solution is valid when considering the constraint on the position update in (2.14). Whenever $P_0^{(0)} > -1/(2hX_0)$, *i.e.*, $\Delta < 0$, (2.20) admits no solution. Recalling that $P_0^{(0)} \sim \mathrm{N}(\mu_0, 1/X_0^2)$, this event occurs with positive probability, thus violating the assumption made by Kook et al. [2022].

2.4 Theoretical results

We now study the reversibility of n-BHMC with respect to the target distribution. To do so, we first present theoretical results on the *exact* integrators of the discretized Hamiltonian dynamics in Section 2.4.1, from which we derive our assumption on the numerical integrator used in n-BHMC. Finally, we state our main result on n-BHMC in Section 2.4.2.

2.4.1 From implicit to numerical integrators

We show in Proposition 2 that even though F_h is a *set-valued* map, it can *locally* be identified with a C^1-diffeomorphism. In a manner akin to Lelièvre et al. [2022], this justifies our assumption **A**2 that the *numerical* map Φ_h used to approximate F_h in Step 2 of Algorithm 1 is *locally* a C^1-involution.

Proposition 2. *Assume* **A**0, **A**1. *Let* $z^{(0)} \in \mathrm{T}^\star\mathsf{M}$, *then there exists* $h^\star > 0$ *(explicit in Appendix A.7) such that for any* $h \in (0, h^\star)$, *there exist* $z_h^{(1)} \in \mathrm{F}_h(z^{(0)})$, *a neighborhood* $\mathsf{U} \subset \mathrm{T}^\star\mathsf{M}$ *of* $z^{(0)}$ *and a* C^1*-diffeomorphism* $\gamma_h : \mathsf{U} \to \gamma_h(\mathsf{U}) \subset \mathrm{T}^\star\mathsf{M}$ *with*
(a) $\gamma_h(z^{(0)}) = z_h^{(1)}$ *and* $|\det \mathrm{Jac}(\gamma_h)| = 1$.

(b) $\gamma_h(z)$ *is the only element of* $\mathrm{F}_h(z)$ *in* $\gamma_h(\mathsf{U})$ *for any* $z \in \mathsf{U}$.

Proposition 2 shows that while F_h can take multiple (or none) values on $\mathrm{T}^\star\mathsf{M}$, for any $z^{(0)} \in \mathrm{T}^\star\mathsf{M}$, there exists h small enough and a neighborhood U of $z^{(0)}$ such that the set-valued map F_h is locally symplectic on U. The proof of Proposition 2 is given in Appendix A.7. It first relies on considering the Störmer-Verlet scheme introduced in (2.14) as the composition of maps for which we derive the existence of solutions and secondly applying the implicit function theorem on these maps. Motivated by this result on the *implicit map* F_h and the fact that for any $z \in \mathrm{T}^\star\mathsf{M}$ with $|\mathrm{F}_h(z)| > 0$, $z \in \mathrm{F}_h \circ \mathrm{F}_h(z)$ (see Section 2.3.1), we make the following assumption on the *numerical map* Φ_h.

A2. *There exists* $\lambda \in (0,1)$ *s.t. for any* $z^{(0)} \in \mathrm{T}^\star\mathsf{M}$, *there exists* $h_\star > 0$ *and for any* $h \in (0, h_\star)$
(a) $\mathsf{B} = \mathsf{B}_{\|\cdot\|_{z^{(0)}}}(z^{(0)}, \lambda r^\star(x^{(0)})) \subset \mathrm{dom}_{\Phi_h}$,
(b) $\Phi_h \in \mathrm{C}^1(\mathsf{B}, \mathrm{T}^\star\mathsf{M})$ *and* $\Phi_h \circ \Phi_h = \mathrm{Id}$ *on* B,
with $r^\star : \mathsf{M} \to (0, +\infty)$ *depending only on* $\mathfrak{g}$ *and defined in Appendix A.9.*

Assumption **A**2 can be thought as a strengthening of Proposition 2-(a), where (i) $h_\star$ refers to $h^\star$, and (ii) B and Φ_h correspond to *explicit* versions of U and γ_h. We conjecture that the involution condition in **A**2-(b) could in fact be replaced by the condition that for any $z \in \mathsf{B}$, $\Phi_h(z) \in \mathrm{F}_h(z)$, in a manner akin to Lelièvre et al. [2022]. We leave this study for future work.

2.4.2 Reversibility results

Beyond n-BHMC. While Algorithm 1 can be implemented practically, it cannot be easily analysed under **A**0, **A**1 and **A**2. The main reason for this limitation is that the self-concordance properties are defined locally, whereas deriving reversibility results require global control. In particular, we cannot ensure that Φ_h is locally an involution around $z^{(0)}$ if $h > h_\star$. To circumvent this issue, we enforce a condition on h to be small enough in n-BHMC. Using the notation from **A**2, we define the set

$$\mathsf{A}_h = \{z \in \mathrm{T}^\star\mathsf{M} \,:\, h < \min_{\tilde{z} \in \mathsf{B}_{\|\cdot\|_z}(z,1)} h_\star(\tilde{z})\} \subset \mathrm{dom}_{\Phi_h} \,. \tag{2.21}$$

Let $z^{(0)} \in \mathsf{A}_h$. By **A**2, we know that Φ_h is an involution on a neighbourhood of $z^{(0)}$; in particular, it comes that $(\Phi_h \circ \Phi_h)(z^{(0)}) = z^{(0)}$. Hence, the condition (b) of the "involution checking step" in Algorithm 1 is *de facto* satisfied. This naturally leads to replace "$Z_n^{(1)} \in \mathrm{dom}_{\Phi_h}$", *i.e.*, the condition (a) of the "involution checking step", by the more restrictive condition "$Z_n^{(1)} \in \mathsf{A}_h$", as presented in Algorithm 3 (see Appendix A.8), for which we are able to derive a reversibility result.

We denote by $\mathrm{Q} : \mathrm{T}^\star\mathsf{M} \times \mathcal{B}(\mathrm{T}^\star\mathsf{M}) \to [0,1]$, the transition kernel of the (homogeneous) Markov chain $(X_n, P_n)_{n\in\mathbb{N}}$ obtained with Algorithm 3. We now state our main result on n-BHMC.

Theorem 3. *Assume* **A**0, **A**1, **A**2. *Then,* Q *is reversible up to momentum reversal, i.e., we have for any* $f \in \mathrm{C}(\mathrm{T}^\star\mathsf{M} \times \mathrm{T}^\star\mathsf{M}, \mathbb{R})$ *with compact support*

$$\int_{\mathrm{T}^\star\mathsf{M}\times\mathrm{T}^\star\mathsf{M}} f(z,z')\mathrm{d}\bar{\pi}(z)\mathrm{Q}(z,\mathrm{d}z') = \int_{\mathrm{T}^\star\mathsf{M}\times\mathrm{T}^\star\mathsf{M}} f(s(z'),s(z))\mathrm{d}\bar{\pi}(z)\mathrm{Q}(z,\mathrm{d}z') \,. \tag{2.22}$$

In particular, $\bar{\pi}$ *is an invariant measure for* Q.

Proof. We provide here a sketch of the proof, technical details being postponed to Appendix A.9. The reversibility (up to momentum reversal) of the momentum update is straightforward. To establish the reversibility up to momentum reversal of the *numerical* Hamiltonian integration step, we cover the compact support of f with a finite family of open balls with respect to the metric $\mathfrak{g}$. Combining **A**1 and **A**2, we show that Φ_h is a volume-preserving C^1-diffeomorphism on each one of these sets. We then conclude upon combining this result with the fact that $R_h^{\Phi} \circ s \circ S_{h/2} = s \circ S_{h/2} \circ \Phi_h$. □

Although Algorithm 3 may be implemented, this would come with a huge (and unrealistic) computational cost since verifying that $z \in \mathsf{A}_h$ implies to find the minimal critical step-size $h_\star$ on a neighborhood of z. The question of the existence of a global step-size h such that Algorithm 1 can be properly analyzed is not the topic of this paper and is left for future work.

Comparison with Theorem 8 in Kook et al. [2022]. Although Kook et al. [2022] present a theoretical result similar to Theorem 3, we claim that *their statement is not correct*. Indeed, the authors make a critical confusion between the *ideal* integrator as considered in Hairer et al. [2006] and the *numerical* version they implement. Then, in the proof of reversibility for their scheme, they act as if the two algorithms were the same while it is not true (see Appendix A.10 for further details). On the other hand, (i) we make a clear distinction between the ideal integrator and its *numerical* implementation (see Section 2.3.1), and (ii) implement the "involution checking step" to enforce reversibility (Line 8 in Algorithm 1).

2.5 Related work

Sampling on manifolds. Traditional *constrained* sampling methods in Euclidean spaces include the Hit-and-Run algorithm [Lovász and Vempala, 2004], the Random Walk Metropolis-Hastings (RWMH) algorithm, also referred to as Ball Walk [Lee and Vempala, 2017b], and HMC, [Duane et al., 1987]. However, it has been empirically demonstrated that RWMH and HMC require small step-sizes in order to correctly sample from a target distribution π over a *submanifold* $\mathsf{M} \subset \mathbb{R}^d$, thus resulting in poor mixing time [Girolami and Calderhead, 2011, Figures 1 and 3]. In the specific case where $\mathsf{M} = \{x \in \mathbb{R}^d \,:\, c(x) = 0\}$ for some $c : \mathbb{R}^d \to \mathbb{R}^m$, Brubaker et al. [2012] combine HMC with the RATTLE integrator [Leimkuhler and Skeel, 1994], incorporating the constraints of M in the Hamiltonian dynamics via Lagrange multipliers. Girolami and Calderhead [2011] adopt an original approach by endowing M with a Riemannian metric $\mathfrak{g}$ and propose RMHMC (Riemannian Manifold HMC), a version of HMC where the Hamiltonian depends on $\mathfrak{g}$ as in (2.4). It consists of integrating the Hamiltonian dynamics of (2.7) on short-time steps using the generalized Leapfrog integrator (2.14) and the acceptance filter defined in (2.19). This method does not include an "involution checking step", and thus the reversibility of the algorithm cannot be ensured in practice and in theory. Similarly, Byrne and Girolami [2013] propose to design a numerical integrator for RMHMC using geodesics.

Choice of the metric in RMHMC. There are various ways to design a *meaningful* metric $\mathfrak{g}$ given a submanifold M. In a Bayesian perspective, Girolami and Calderhead [2011] aim at computing the *a posteriori* distribution of a statistical model of interest and thus choose $\mathfrak{g}$ to be the Fisher-Rao metric. In this paper, we consider another approach which exploits the *geometrical* structure of M. For instance, one can always define a self-concordant barrier ϕ when M is a convex body, [Nesterov and Nemirovskii, 1994], and choose $\mathfrak{g} = D^2\phi$, as done by Kook et al. [2022].

Self-concordance in RMHMC. Elaborating on Riemannian geodesics, Lee and Vempala [2018] provide theoretical guarantees of fast mixing time for RMHMC when (i) M is a polytope, and (ii) $\mathfrak{g}$ is the Hessian of a self-concordant function ϕ (as in our setting). Their result notably improves the complexity of uniform polytope sampling from algorithms relying on self-concordance such as the Dikin Walk [Kannan and Narayanan, 2009] and the Geodesic Walk [Lee and Vempala, 2017a]. However, they only consider the case where the *exact* continuous Hamiltonian dynamics is used, as in c-BHMC (see Appendix A.4). On the other hand, Kook et al. [2022] integrate the Hamiltonian dynamics via implicit schemes without any "involution checking step" in a similar self-concordant setting. In particular, they consider a convex body K equipped with a self-concordant barrier ϕ, combined with a linear equality constraint $\mathrm{A}x = b$. This is a special case of our framework (see **A**0 and **A**1) by rewriting this whole set as $\mathsf{K}' = \{\mathrm{A}^\dagger b + u \in \mathsf{K} \,:\, u \in \mathrm{Ker}(\mathrm{A})\}$[2], equipped with the self-concordant barrier $u \mapsto \phi(\mathrm{A}^\dagger b + u)$. Besides this, Kook et al. [2022] provide an efficient implementation of their algorithm (CRHMC) in the case of a convex bounded manifold of the form $\mathsf{M} = \{x \in \mathbb{R}^d \,:\, \mathrm{A}x = b, \ell < x < u\}$. Although

[2] $\mathrm{A}^\dagger$ is the pseudo-inverse of A.

CRHMC demonstrates empirical fast mixing time, we show in Section 2.6 that it suffers from an asymptotic bias. More recently, Kook et al. [2023] built upon the empirical results of Kook et al. [2022] to derive theoretical results of fast mixing time. However, they do not question the issue of asymptotic bias in CRHMC.

Enforcing reversibility. Zappa et al. [2018] propose a version of RWMH including projection steps after each proposal. To our knowledge, they are the first to practice "involution checking" of the proposal, and thus *enforce the reversibility* of the Markov chain with respect to π. Lelièvre et al. [2019] notably combine this method with the discretization suggested by Brubaker et al. [2012] to also *enforce the constraints* of the manifold. Lelièvre et al. [2022] elaborate on this framework, by designing a symplectic numerical integrator with multiple possible outputs, and provide a rigorous proof of reversibility. Note that it only applies when $\mathfrak{g}$ is induced by the flat metric of $\mathbb{R}^d$ and therefore cannot be combined with our approach as such.

2.6 Numerical experiments

In our experiments[3], we illustrate the performance of n-BHMC (Algorithm 1) to sample from target distributions which are supported on polytopes. We compare our method with the numerical implementation of CRHMC[4] provided by Kook et al. [2022]. In all of our settings, we compute $\mathfrak{g}$ as the Hessian of the logarithmic barrier, see Section 2.2.2. The algorithms are always initialized at the center of mass of the considered polytope. At each iteration of n-BHMC, we perform one step of numerical integration, using the Störmer-Verlet scheme with $K = 30$ fixed-point steps and keep the refresh parameter β equal to 1. We refer to Appendix A.11 for more details on the setting of our experiments and additional results.

Synthetic data. We first consider the problem of sampling from the truncated Gaussian distribution

$$(\mathrm{d}\pi/\mathrm{dLeb})(x) = \exp[-\|x-\mu\|^2/2]\mathbb{1}_{\mathsf{M}}(x)/\int_{\mathsf{M}} \exp[-\|\tilde{x}-\mu\|^2/2]\mathrm{d}\tilde{x}, \quad \forall x \in \mathbb{R}^d\,, \tag{2.23}$$

where M is the hypercube or the simplex, and $\mu \in \mathbb{R}^d$ is defined by

$$\mu = (10/\sqrt{d-1}) \times \{\mathbf{1} - e_1 + (\sqrt{d-1}-1)e_2\}\,,$$

where e_i stands for the i-th canonical vector of $\mathbb{R}^d$ and $\mathbf{1} = \sum_{i=1}^d e_i$. In particular, $\mu_1 = 0$, $\mu_2 = 10$ and $\mu_j = \mu_2/\sqrt{d-1}$ for any $j \in \{3,\dots,d\}$. Therefore, the mass of π is not evenly distributed on the boundary of M, which is key to observe the impact of the reversibility condition. For this experiment, we consider the low-dimensional setting $d \in \{5, 10\}$. To sample from π, we run 10 times the algorithms CRHMC and n-BHMC for $N = 10^5$ iterations, and update the step-size h such that the average acceptance rate in the MH filter is roundly equal to 0.5, following Roberts and Rosenthal [2001]. We discuss the setting of the tolerance parameter η for n-BHMC in Appendix A.11. In order to correctly assess the bias of each numerical method, we aim at accurately computing the target quantity $Q = \int_{\mathsf{M}} \langle x, \mu\rangle_2 \mathrm{d}\pi(x)$. Although our choice of Q is arbitrary, it highlights well the bias in CRHMC, which is corrected when using n-BHMC.

We evaluate their performance by taking as ground truth the estimate given by (i) the Metropolis-Adjusted Langevin Algorithm (MALA) [Roberts and Stramer, 2002] for the hypercube and (ii) the Independent MH (IMH) algorithm [Liu, 1996] for the simplex, which are also run 10 times, but 10 times longer than n-BHMC and CRHMC, *i.e.*, for $N = 10^6$ iterations. To compute the target quantity Q, we keep the whole trajectories and report the confidence intervals in Figure 2.2. For both polytopes, we notably observe that our method is less biased than CRHMC.

Real-world data. We then consider 10 polytopes given in the COBRA Toolbox v3.0 [Heirendt et al., 2019], which model molecular systems, and follow the method provided in [Kook et al., 2022, Appendix A] to pre-process them. Here, we aim at uniformly sampling from these polytopes. However, we do not have access to a realistic baseline that yields an unbiased estimator, since the sampling dimension is too high and running MALA or IMH would be too costly. Hence, instead of assessing the bias of the algorithms, we rather want to highlight that the "involution checking step" does not hurt the convergence properties of BHMC compared to CRHMC. We evaluate the efficiency of the algorithms by computing *the sampling time per effective sample* (in seconds), defined as the total sampling time until termination divided by the Effective Sample Size. We set the initial step-size to 0.01 for both algorithms and $\eta = 10$ in n-BHMC. We then follow the exact same procedure as in [Kook et al., 2022, Table 1] by drawing 1000 uniform samples with limit on running time set to 1 day.

[3] Our code: https://github.com/maxencenoble/barrier-hamiltonian-monte-carlo.
[4] https://github.com/ConstrainedSampler/PolytopeSamplerMatlab

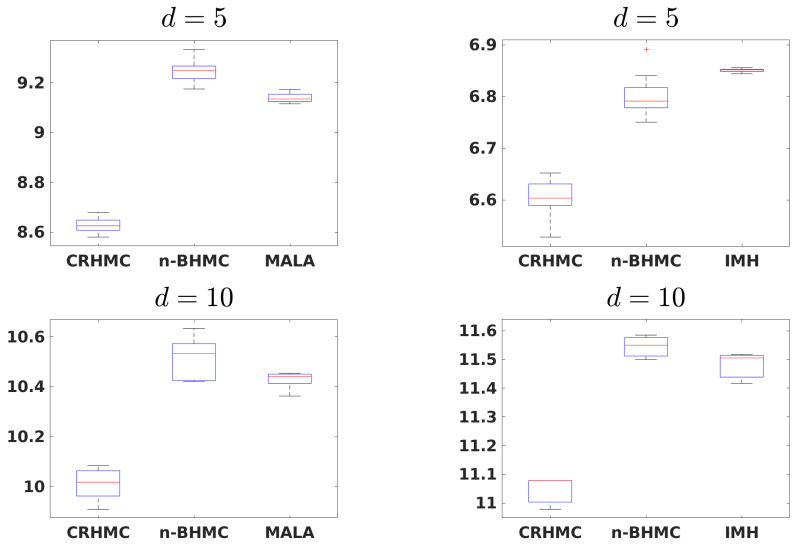


Figure 2.2: Comparison between n-BHMC and CRHMC on the hypercube (left) and the simplex (right).

Model	d	NNZ	n-BHMC	CRHMC
ecoli	95	291	**0.08822**	5.719
cardiac-mit	220	228	**0.1905**	5.304
Aci-D21	851	1758	141.9	**46.17**
Aci-MR95	994	2859	325.1	**61.34**
Abi-49176	1069	2951	179.4	**74.22**
Aci-20731	1090	2946	**0.7124**	88.82
Aci-PHEA	1561	4640	**2.343**	100.8
iAF1260	2382	6368	537.9	**170.5**
iJO1366	2583	7284	537.9	**169.4**
Recon1	3742	8717	**3.740**	3280

Table 2.1: Real-world data setting: comparison with Kook et al. [2022].

Results are given in Table 2.1. For each molecular model, we specify by NNZ the number of non-zero entries in the matrix A defining the corresponding pre-processed polytope, which thus reflects its complexity. In particular, the sampling dimension d corresponds to the number of columns of A. Note that the sampling time values are not of interest in themselves, but are only meant to compare CRHMC and n-BHMC. While it is clear that adding an "involution checking step" implies a trade-off between accuracy and complexity of the method, our results demonstrate that it does not penalize BHMC in the considered settings, and may even make it more efficient in some cases.

2.7 Discussion

In this paper, we introduced a novel version of RMHMC, Barrier HMC (BHMC), which addresses the problem of sampling from a distribution π over a constrained convex subset $\mathsf{M} \subset \mathbb{R}^d$ equipped with a self-concordant barrier ϕ. Our contribution highlights that the use of well known implicit schemes for ODE integration combined with these space constraints leads to asymptotic bias when it comes to their numerical implementation. We propose to solve it in a straightforward manner via our algorithm, n-BHMC, which relies on an additional "involution checking step". Under reasonable assumptions, our theory shows that this critical step removes the asymptotic bias. This result is supported by numerical experiments where we highlight this lack of bias compared to state-of-the-art methods. In future work, we would like to investigate the "coupled" behaviour of the hyperparameters h and η in practice, the study of irreductiblity of n-BHMC and the influence of η on the bias. Moreover, we are interested in the application of n-BHMC for efficient polytope volume computation. More generally, we are convinced that our study is a first step for future work on designing implicit integration schemes for unbiased constrained sampling methods.

Acknowledgments

AD acknowledges support from the Lagrange Mathematics and Computing Research Center. AD and MN would like to thank the Isaac Newton Institute for Mathematical Sciences for support and hospitality during the programme *The mathematical and statistical foundation of future data-driven engineering* when work on this paper was undertaken. MN acknowledges funding from the grant SCAI (ANR-19-CHIA-0002).

Chapter 3

Improving the evaluation of samplers on multi-modal targets

3.1 Introduction

Consider the problem of sampling from a probability density known up to a normalizing constant. More precisely, let π be a target distribution with probability density given by $x \mapsto \gamma(x)/\mathcal{Z}$, where $\gamma : \mathbb{R}^d \to \mathbb{R}_+$ can be pointwise evaluated and the normalizing constant $\mathcal{Z} = \int_{\mathbb{R}^d} \gamma(x)\mathrm{d}x$ is intractable. This reflection paper is concerned with cases where the distribution π is multi-modal, i.e., γ admits several local maxima. Such high-dimensional multi-modal distributions appear in various fields. In physics, they describe the thermodynamics of metastable systems, or the action of quantum field theories [Wolff, 1990]. Bayesian modeling can also face multi-modal posterior distributions in cases where the observations fail to lift ambiguities between several distinctive sets of parameters [Yao et al., 2022]. Hence, continuing efforts have been made to design sampling strategies that handle multi-modality. However, sampling from multi-modal distributions is notoriously difficult, especially in large dimension d.

There are two major sources of difficulties to overcome: (i) locating the modes, that is the different high-probability regions of π induced by its local maxima, and (ii) estimating their relative importance. Performing an exhaustive search of modes is equivalent to the problem of finding the local maxima of the possibly high-dimensional function γ, which is NP-hard Pardalos and Schnitger [1988]. Although a flurry of heuristics have been developed to explore probability landscapes and find modes [Fletcher, 2000, Nocedal and Wright, 2006], this fact sets the problem of sampling from multi-modal distributions as arbitrarily hard in general.

In fact, even when the location of the modes is known, the second difficulty of estimating their relative proportion remains a significant challenge to this day. The ability of samplers to represent the different modes with correct proportions is, however, seldom reported, with exceptions including Grenioux et al. [2023b], Blessing et al. [2024], Shi et al. [2024], Noble et al. [2025].

In this reflection paper, we wish to advocate for:

- Running systematic experiments on toy mixture distributions with increasing dimension and mode separation, as these two parameters are commonly mediating the hardness of the task for common sampling technics.
- Evaluation based on the criterion of accurate estimation of the statistical weight of each mode, possibly assuming knowledge of mode locations. The reasons are twofold: first because it is an interpretable metric, incurring almost no additional computational cost, and second precisely because it is a challenging test.

Running these experiments and evaluation would enable diagnosing strengths and weaknesses of newly proposed algorithms, which is crucial to foster the development of robust algorithms in the multi-modal setting, and therefore drive progress in the field.

In the following, we start with some reminders in Section 3.2 on the difficulty of sampling from multi-modal distributions, and in particular in estimating relative importance of modes. In Section 3.3, we examine the strengths and weaknesses of common sampling strategies in multi-modal scenarios and illustrate the importance of reporting performance within the proposed experimental framework.

3.2 What is hard in sampling from multi-modal distributions?

As mentioned in the introduction, the first challenge in sampling from a multi-modal distribution is identifying all its modes. We do not discuss this issue further, as exhaustive searches can be arbitrarily complex. Moreover, even when mode locations are known, accurate sampling remains difficult and practically relevant. In some cases, heuristics or expert knowledge can help identify modes. For instance, in physics and drug design [Jorgensen, 2004], determining molecular binding events involves two modes: the bonded complex and the unbound molecules.

Benefits and pitfalls of local samplers. If the target's mode locations are known, it may seem that the sampling problem is solved by using this prior information to initialize a Markov Chain Monte Carlo (MCMC) sampler. MCMCs based on local updates are among the most widely used sampling approaches, including Random Walk Metropolis-Hastings (RWMH) [Metropolis et al., 1953], Metropolis-Adjusted Langevin Algorithm (MALA) [Roberts and Tweedie, 1996], Hamiltonian Monte Carlo (HMC) [Duane et al., 1987, Brooks et al., 2011], and No U-Turn Sampler (NUTS) [Hoffman and Gelman, 2014]. These algorithms have demonstrated both theoretical and practical efficiency when the target density satisfies log-concavity assumptions [Dalalyan, 2017, Durmus and Moulines, 2019, Dwivedi et al., 2018, Bou-Rabee et al., 2020]. Paradoxically, the very properties that make these samplers effective for log-concave distributions hinder their performance in the multi-modal case, where log-concavity does not hold. While MCMC transition kernels are designed to guide the chain toward target modes, they also, with high probability, trap it within a mode, preventing broader exploration. Consequently, initializing an MCMC sampler at the modes of π—where γ is locally log-concave—may yield locally accurate samples while failing to capture the global mode weights of π.

Mode weight estimation. To define the mode weights of the distribution of interest π, one can leverage a partition of its support (taken equal to $\mathbb{R}^d$ here for simplicity) that draws a one-to-one correspondence between the target modes and space subsets. Then, the mode weights of π can be defined as the relative statistical weights of the constituents of this partition with respect to π. Formally, assume that π has K modes and that we have access to a partition $\{S_1, \dots, S_K\}$ of $\mathbb{R}^d$, where $S_k \subset \mathbb{R}^d$ represents a region including the k-th mode of π. The (un-normalized) k-th mode weight of π is simply defined by

$$w_k = \int_{\mathbb{R}^d} \mathbb{1}_{S_k}(x)\gamma(x)\mathrm{d}x = \mathcal{Z}\pi(S_k) \, . \tag{3.1}$$

The important quantities are the weight ratios $w_k/w_{k'}$ for $k' \neq k$ for which the normalizing constant $\mathcal{Z}$ simplifies. Therefore, the weights w_k may be defined up to a common multiplicative constant. Fixing the multiplicative constant, w_k can be interpreted as the normalization constant of the target distribution π restricted to S_k, with un-normalized density given by $x \mapsto \mathbb{1}_{S_k}(x)\gamma(x)$. As such, it becomes clear that mode weight estimation is related to the estimation of a normalization constant.

The need to go beyond local MCMC samplers for multi-modal distributions is well established, with annealing-based methods playing a key role (detailed in the next section). However, accurately estimating mode weights, even when mode locations are known, remains an open challenge in high dimensions. This issue parallels free energy computations in statistical mechanics, an active area of research [Lelièvre et al., 2010, Chipot, 2023].

Despite this, new samplers rarely evaluate their accuracy in estimating mode weights. Yet, this criterion complements common choices like integral probability metrics (IPMs) – such as the Wasserstein distances [Peyré et al., 2019] or the maximum mean discrepancy [Gretton et al., 2012] – the estimation of which is complex and typically comes without guarantees like unbiasedness. Mode weights, in contrast, can be easily estimated from the proportion of samples in each mode, offering a more interpretable measure that directly highlights the core challenge of multi-modal sampling.

3.3 Towards a systematic experiment for evaluating samplers

We propose a simple but challenging experimental setting to assess the ability of a given sampler at handling multi-modality by probing systematically two sources of difficulty: dimension and separation of the modes. We illustrate this setting by applying it to a set of common samplers, revealing their strengths and weaknesses in the task of mode weight estimation.

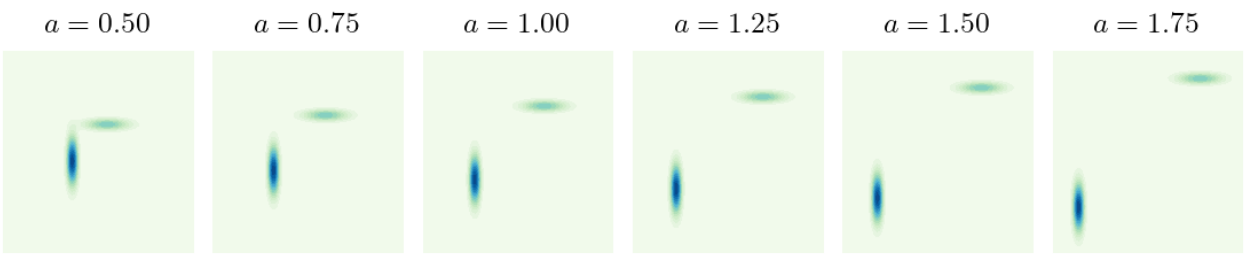


Figure 3.1: Density of the bi-modal Gaussian mixture used in our experiments ($d = 2$) with increasing values of a. As expected, a larger value of a makes the modes further from each other, which reinforces the sampling difficulty.

3.3.1 Description of the experimental setting

Multi-modal target distribution. We consider π defined as a d-dimensional bi-modal Gaussian mixture with unbalanced weights and covariances with strong components in different directions. More precisely, the mixture components are $q_1(x) = \mathrm{N}(x; -a\mathbf{1}_d, \Sigma_1)$, with weight $w = 2/3$, and $q_2(x) = \mathrm{N}(x; a\mathbf{1}_d, \Sigma_2)$, with weight $1 - w = 1/3$, where $a \in (0, \infty)$ controls the distance between the local maxima of π as showed in Figure 3.1, $\mathbf{1}_d$ is the d-dimensional vector with all components equal to 1, Σ_1, Σ_2 are fixed diagonal positive matrices with conditioning number equal to 20. To make this target even more challenging, we set those covariance matrices such that q_1 has marginally large variance in directions where q_2 has small variance (and inversely), see Appendix B.2 for more details. In this setting, the mode partition of π is taken as $S_1 = \{x \in \mathbb{R}^d : q_1(x) > q_2(x)\}$ and $S_2 = S_1^c$, such that the induced w_1, defined in (3.1), verifies $w_1 \approx w$.

Mode weight metric. Below, we report the bias and variance for the Monte Carlo estimations of w_1, while varying two parameters: the distance between the mode centroids, controlled by the parameter a, and the dimension d. The results are displayed on Figures 3.2 and 3.3 for a selection of sampling methods. We provide, in Appendix B.2, a brief description each method with its hyperparameters and, below, comment on its performance. Small bias and variance indicate that the considered sampler is able to recover the mode proportions of our challenging test target both accurately and robustly, and is therefore a promising approach.

General remarks on the methodology. We emphasize that our numerical results should not be interpreted as a precise or definitive quantitative comparison between samplers for several reasons. First, their computational budgets were only roughly matched, see Figure B.2 in Appendix B.2.

Secondly, to showcase their best-case performance, their hyperparameters were individually tuned based on access to ground-truth samples, which would not be available in real-world scenarios, thus making their setting unrealistic for practical deployment.

Additionally, while the experimental target is admittedly a simplified, toy example compared to the complex multi-modal distributions discussed above, we note that most of the considered samplers already fail to perform well on this basic benchmark. This is a key observation: we argue that if a method cannot reliably sample from this controlled bi-modal setting, it is unlikely to succeed in more challenging, high-dimensional applications. Rather than offering fine-grained comparisons, these results are therefore intended to provide insight into the typical behavior and limitations of popular sampling algorithms.

3.3.2 Evaluation on various samplers

Local Markov Chain Monte Carlo. MCMC methods produce Markov chains that admit π as their stationary measure. Standard designs of transition kernels for log-concave distributions are based on discretizations of either Langevin or Hamiltonian dynamics. However, as already discussed in Section 3.2, the resulting sampling algorithms are inefficient when facing multi-modal targets, as moving between distinct modes becomes an unlikely event.

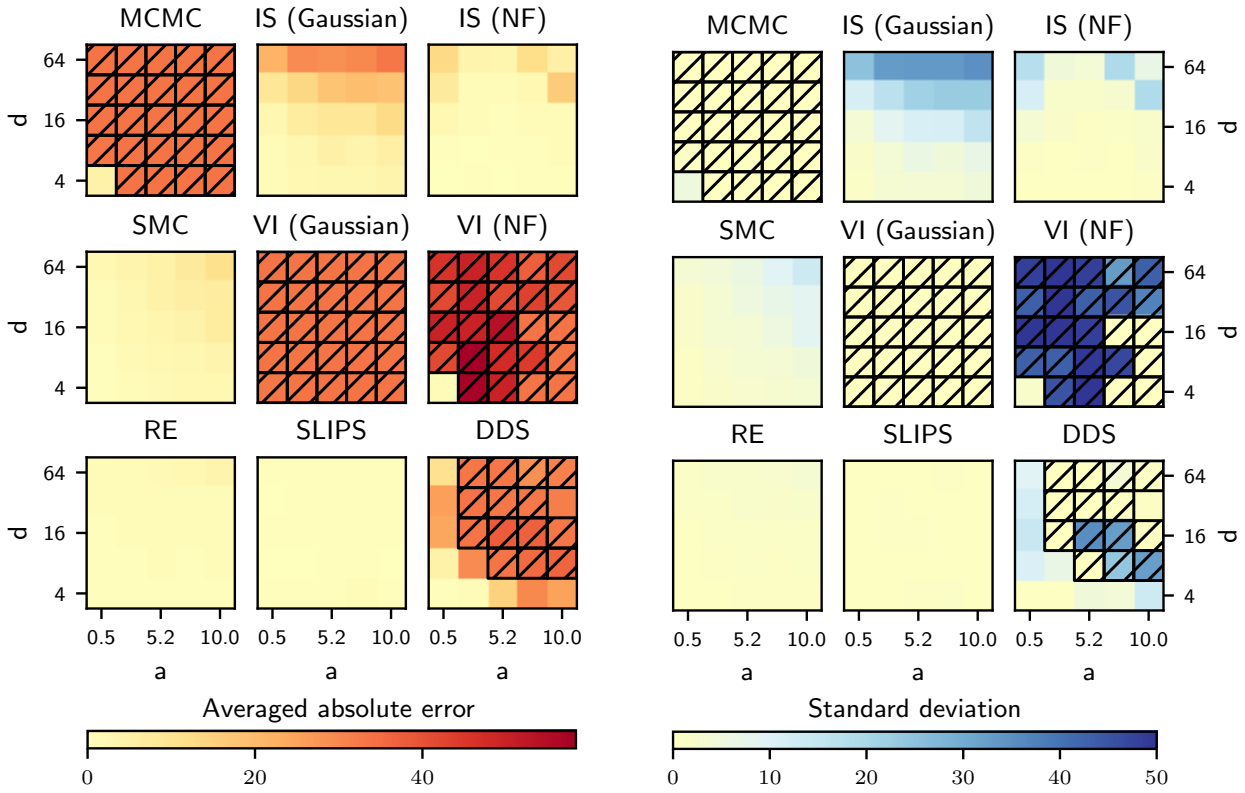


Figure 3.2: Results on mode weight estimation for the bi-modal Gaussian mixture defined in Section 3.3.1, when varying hyperparameters a and d. For each setting, we aggregate 48 Monte Carlo estimations, each one being computed over 8192 samples. Hashed areas indicate settings with systematic mode collapse in the sampling process. **(Left):** Averaged absolute error of the estimation with respect to the true mode weight $w \approx 66.7\%$. **(Right):** Standard deviation of the estimation.

Testing an implementation of MALA, this behavior is easily evidenced in our experiment. The chains get trapped in one of the modes as soon as there exists a significant log-probability "barrier" between the modes, i.e., when a or d increases. The zero variance observed in these cases is not to be mistaken as a positive result. As the chains stay trapped in the mode they are initialized in, a setting we refer to as *mode collapsed*, the estimation of w_1 is consistently biased in the same manner.

Importance Sampling. Importance Sampling (IS) builds Monte Carlo estimators as re-weighted averages of samples from a tractable proposal distribution. The variance of IS estimations, related to the degeneracy of the so-called "importance weights", depends on the agreement between the target and the proposal distributions and is known to increase exponentially with d [Agapiou et al., 2017].

In Figure 3.2, the bias and variance of the estimator of w_1 are reported for two versions of IS. First, using a unimodal Gaussian proposal, the accuracy decreases rapidly along both the mode separation and dimension axes. As either of these parameters increases, the intersection of the typical sets of the target and proposal distribution decreases, leading to more degenerate weights. Using a normalizing flow has been proposed to adapt the proposal to the target and mitigate the issue [Müller et al., 2019, Noé et al., 2019]. Here, the normalizing flow was trained on target samples to serve as a proposal, and indeed mitigates the degradation of performance, in particular with respect to the mode separation. The normalizing flow can learn to focus mass on different modes even if distant. However, the lack of perfect agreements between the target and the proposal starts showing as the dimension increases.

Variational Inference. Variational Inference (VI) reformulates the sampling problem as an optimization problem over the space of probability distributions. It seeks the closest distribution to the target within a family of tractable parametric distributions, known as the variational family. This problem requires a notion of discrepancy between probability distributions, which is chosen as the reverse Kullback-Leibler (KL) divergence in most cases.

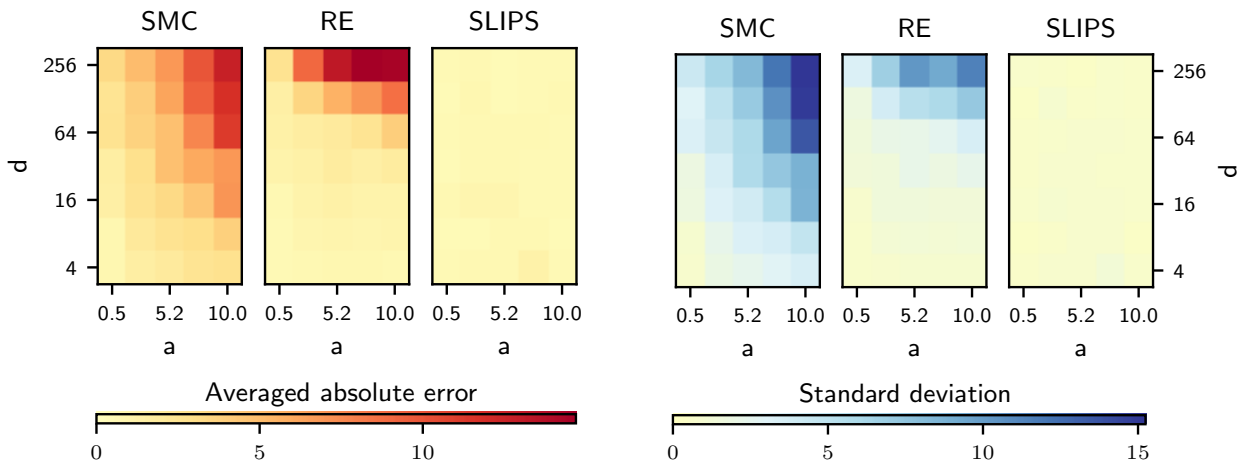


Figure 3.3: Same as Figure 3.2 for the best performing SMC, RE and SLIPS reaching higher-dimensions. For sake of pedagogy, the color scale is different from above (a consistent color scale can be found in Figure B.1 of Appendix B.2). **(Left):** Averaged absolute error of the estimation with respect to the true mode weight $w \approx 66.7\%$. **(Right):** Standard deviation of the estimation.

For multi-modal target distributions, many works have showed that even for variational family able to represent multiple modes, the VI procedure often leads to mode collapse, due to the mode-seeking behavior of the reverse KL divergence [MacKay, 2001, Jerfel et al., 2021, Blessing et al., 2024, Soletskyi et al., 2024].

In our experiment, we consider two families of variational distributions, unimodal Gaussian distributions and normalizing flows Rezende and Mohamed [2015]. In the Gaussian case, we systematically observe a mode collapsed estimation, with the same consistent bias as in the MCMC setting, resulting in a zero-variance outcome. While using normalizing flows remediates the issue when the modes are almost connected and the dimension is low (lowest values of a and d), mode collapse is systematic beyond this setting.

Annealed sampling methods. Annealed samplers were developed to address the challenges posed by multi-modal distributions. The core idea is to introduce a sequence of distributions that gradually transition from an easy-to-sample distribution to the target distribution. These samplers leverage this sequence by breaking the original sampling problem into multiple, more manageable subproblems. In practice, this is often achieved through a geometric interpolation of densities.

Sequential Annealed Sampling. A naive annealing approach consists in performing recursive MCMC sampling of the bridging distributions. Beginning with the simplest distribution and sequentially moving towards the target, each intermediate distributions is targeted by a standard MCMC sampler, that uses samples obtained for the preceding distribution as a warm-start. The Sequential Monte Carlo (SMC) algorithm [Del Moral et al., 2006] improves upon this approach by reweighting warm-start samples at certain steps using importance weights updated throughout the process. While SMC significantly outperforms the earlier samplers in handling multi-modality, it has been shown to degrade in high dimensions. Similar to Importance Sampling (IS), its resampling weights suffer from degeneracy [Li et al., 2014], a problem that can be exacerbated by the default geometric interpolation path [Máté and Fleuret, 2023]. Increasing the number of intermediate steps can help mitigate this issue in higher dimensions but comes at a significant computational cost.

For our experiment, we leverage an optimal sequence of intermediate distributions based on the recommendations of Zhou et al. [2016] and Syed et al. [2024], placing SMC in its most favorable setting. In Figure 3.2, we observe that SMC is perfectly able to recover the true mode proportion with high accuracy and robustness, confirming the superiority of annealing schemes over standard MCMC methods for multi-modal distributions. However, as expected, SMC performance deteriorates as soon as both of the mode separation and the dimension increase together as shown in Figure 3.3.

Parallel annealed sampling. A second approach amounts to perform MCMC sampling of each intermediate distribution in parallel, and to exchange states from the generated adjacent chains based on a Metropolis-Hastings acceptance step. This algorithm is often referred to as Replica Exchange (RE) [Swendsen and Wang, 1986] or Parallel Tempering (PT) [Geyer et al., 1991, Hukushima and Nemoto, 1996]. Similarly to sequential

annealing, this approach is known to suffer from high dimension, due to the degeneracy of the acceptance probabilities.

Here, we test an implementation of RE with a number of intermediate distributions that is fixed over the choices of a and d [1]. Although RE accurately recovers the true mode proportion in the settings from Figure 3.2, its performance declines with increasing mode separation and dimension, as showed in Figure 3.3. This degradation appears slower than SMC, but note the adaptive SMC converged in a computational budget 20 times faster than the one considered for RE.

While adaptive methods have been proposed to optimize the annealing schedule in RE (similarly to those developed for SMC) [Syed et al., 2021, 2022], we did not include them in this study, as they substantially increase algorithmic complexity and computational cost. Moreover, RE performance is known to deteriorate with dimensionality, as swap acceptance rates typically decay exponentially [Atchadé et al., 2011, Syed et al., 2022]; this limitation can be alleviated by increasing the number of intermediate chains, as was pointed out to us following this work. Both directions constitute promising avenues for future research aimed at enabling fairer and more informative benchmarking in high-dimensional settings.

Diffusion-based sampling methods. Building on classical sequential annealing, the recently introduced diffusion-based sampling framework generates samples from the target distribution by following a Stochastic Differential Equation (SDE) that transitions from an easy-to-sample distribution to the target via a predefined stochastic interpolant process. This approach is inspired by the success of score-based diffusion models in generative modeling [Song et al., 2021b, Albergo et al., 2023], where the goal is to generate samples without access to an explicit density. In practice, the drift term of this generative SDE is intractable and must be estimated, typically using Monte Carlo methods or variational approximations.

Monte-Carlo diffusion-based sampling. Leveraging the fact that the intractable drift can be expressed as a conditional expectation, several recent works have proposed estimating it on-the-fly—i.e., while running the generative SDE—using unparameterized Monte Carlo methods [Huang et al., 2024a,b, He et al., 2024b]. A related approach, Stochastic Localization via Iterative Posterior Sampling (SLIPS) [Grenioux et al., 2024], estimates the drift using standard MCMC techniques. While these methods have demonstrated strong empirical performance, they require careful hyperparameter tuning which becomes particularly challenging when no ground truth samples are available for their calibration, which is often the case in practice. This represents a critical limitation for their real-world applicability.

With this informed tuning, SLIPS performs very well at the mode weight estimation task as presented in Figure 3.2 and even outperforms SMC and RE in higher dimension, as displayed in Figure 3.3. This result may be explained by the fact that the sequence of intermediate densities induced by the diffusion path (which are also bi-modal) preserve the mode weights of the target distribution. This is contrary to the geometric interpolation paths [Máté and Fleuret, 2023] and avoids the use of advanced resampling schemes that notoriously fail when a or d increases.

Variational diffusion-based sampling. Another line of research has proposed to learn the drift of the generative SDE with a neural network, by solving a Variational Inference problem, this time defined on the space of path measures [Zhang and Chen, 2022, Berner et al., 2023, Vargas et al., 2024, Chen et al., 2025]. In this case, one aims to minimize a discrepancy between the generative SDE and a variational family of neural SDEs, i.e., SDEs whose drift is a neural network. In our experiment, we considered the Denoising Diffusion Sampler (DDS) [Vargas et al., 2023a]. The interest of such parameterized methods lies in their ability to amortize inference; once the drift is learned, samples can be obtained by solving the corresponding neural SDE. Their performance were found to severely depends on the values of their hyperaparameters Noble et al. [2025].

In Figure 3.2, we display results of DDS, where its hyperparameters were tuned favorably with respect to π, see Appendix B.2. Despite relative good performance in the most simple multi-modal settings, we still observe that it suffers from the joint increase of a and d. The difference with the Monte-Carlo diffusion-based sampler (SLIPS) may be explained by a manifestation of mode collapse in this extended VI problem.

[1] We explored various types of discretization schedules for the intermediate distributions and observed that a logarithmic spacing was the best suited for this target.

3.4 Perspectives

We have demonstrated that evaluating a sampler's ability to recover mode weights accurately provides an interpretable benchmark for designing samplers for multi-modal distributions. Gradually increasing both dimensionality and mode separation in a controlled experiment reveals the breaking points of different samplers and helps identify the most promising approaches. Designed to handle multi-modal distributions, traditional annealing-based samplers, such as SMC and RE, are generally more robust than direct sampling methods. Recently introduced diffusion-based samplers also show promise, but their precise tuning and scalability remain open challenges.

While we focused on dimension and mode separation, the number of modes also adds to the challenge of sampling. The proposed experimental setting can be straightforwardly extended in this third axis of difficulty. In this case, the mode weight criterion translates into a comparison of the empirical histogram of visited modes with the ground-truth components' weights (see e.g., Grenioux et al. [2023b]). We also note that the proposed framework does not include heavy-tails and complex local geometries, and it would be an interesting direction to define a systematic benchmark including also these features.

It is worth emphasizing that this benchmark represents a deliberately simplified form of multimodality. As discussed earlier, real-world problems often involve far more complex landscapes, featuring exponentially many modes or severe ill-conditioning, which could alter algorithmic rankings and expose additional limitations. Still, controlled studies such as ours provide a necessary foundation: if a sampler fails on these basic multimodal cases, it is unlikely to succeed on more realistic, high-complexity targets. Extending this framework to encompass richer scenarios remains a promising direction for future work.

Acknowledgments

We would like to thank Alain Durmus for many stimulating discussions and for reviewing the manuscript prior to submission. LG acknowledge funding from Hi! Paris. This work was partially performed using HPC resources from GENCI–IDRIS (AD011015234).

Part II

Sampling methodologies based on diffusion models

Chapter 4

Stochastic Localization via Iterative Posterior Sampling

4.1 Introduction

We consider in this paper the problem of sampling from a probability density known up to a normalization constant. This problem finds its origin in various tasks, ranging from Bayesian statistics Kroese et al. [2011] to statistical mechanics [Krauth, 2006], including now generative modeling [Turner et al., 2019, Grenioux et al., 2023a].

Markov chain Monte Carlo (MCMC) samplers are among the most common approaches for this task with a wide span of applicability. In addition, under appropriate conditions on the target, theoretical guarantees can be derived [Dalalyan, 2017, Durmus and Moulines, 2017]. However, for complex distributions, simple MCMC algorithms have some limitations. As a workaround, it has been suggested to solve the problem progressively, targeting intermediate smoothed distributions. The resulting algorithms include Replica Exchange [Geyer et al., 1991], Annealed Importance Sampling [Neal, 2001] and Sequential Monte Carlo [Del Moral et al., 2006]. Nevertheless, these methods can still largely struggle in high-dimensional settings. To address this issue, approximate inference methods such as Variational Inference (VI) [Wainwright and Jordan, 2008] have emerged, in connection with deep generative models such as Variational Auto-Encoders [Rezende et al., 2014, Kingma and Welling, 2014] or Normalizing Flows [Rezende and Mohamed, 2015].

In contrast to "density-based sampling", which is the setting considered in the present work, generative modeling assumes the availability of training samples and aims to generate similar realizations. Henceforth, we refer to generative modeling as "data-based sampling" as opposed to "density-based sampling". Diffusion-based generative models [Sohl-Dickstein et al., 2015, Ho et al., 2020, Song et al., 2021b] now constitute the state-of-the-art of data-based sampling. In these models, noise is progressively added to the training samples via a forward stochastic process. The derivatives of the logarithm of the forward marginal densities, called *scores*, are learned via score matching techniques [Hyvärinen, 2005, Vincent, 2011], and new samples are obtained by simulating the backward process using the learned scores. This approach to generative modeling has been found to scale well with dimension in many of applications [Rombach et al., 2022, Chen et al., 2021a, Nichol et al., 2022] and comes with theoretical guarantees [Chen et al., 2023a].

Extending diffusion-based model techniques to density-based sampling proves challenging as it requires an efficient estimator of the score without samples of the target. Relying on the link made with stochastic optimal control Tzen and Raginsky [2019], Bortoli et al. [2021], Holdijk et al. [2023], Pavon [2022], several VI methods using deep neural networks have recently been proposed for this task [Zhang and Chen, 2022, Berner et al., 2023, Vargas et al., 2023a,c, 2024, Richter et al., 2023]. On the other hand, Huang et al. [2024a] take a non-parametric approach and proposed a scheme based on a MCMC estimation of the scores and therefore can potentially mitigate the numerical bias intrinsic in using neural networks or more broadly any parametric estimation method. This is also our approach.

Closely related to denoising diffusion models, Stochastic Localization (SL) techniques have been first employed as a tool to establish results in geometric measure theory [Eldan, 2013, 2020, 2022, Chen and Eldan, 2022], and more recently have been proposed as a density-based sampling approach. Alaoui et al. [2022], Montanari and

Wu [2023] demonstrate the potential of SL for specific challenging distributions and Ghio et al. [2024] provide a conjecture on the type of distributions which can be efficiently sampled with these techniques. Montanari [2023] provides a blueprint on using SL for density-based sampling but does not discuss a practical strategy for an arbitrary target distribution.

Contributions. Building on these prior works, we bring the following contributions.

- We investigate a general framework of SL that allows to reflect on the role of the signal-to-noise scheduling when using SL for sampling.
- We identify the challenges in its practical implementation and propose a learning-free sampling methodology using MCMC estimation, which requires few tuning. We elucidate our algorithmic design with theoretical and numerical considerations applicable to a certain class of non log-concave distributions.
- We provide numerical evidence of the robustness of the proposed approach in large dimension, beyond the class of distributions amenable to theoretical guarantees. Those results show that the proposed algorithm is on par or superior to modern sampling methods in a wide variety of settings.

The code to reproduce our experiments is available on Github.

Notation. For any distribution μ with finite first moment, its expectation is denoted by $\mathbf{m}_\mu = \int_{\mathbb{R}^d} x \mathrm{d}\mu(x)$. The density of the Gaussian distribution with mean $\mathbf{m} \in \mathbb{R}^d$ and covariance $\Sigma \in \mathbb{R}^{d\times d}$ is denoted by $x \mapsto \mathrm{N}(x; \mathbf{m}, \Sigma)$.

4.2 Background on SL for sampling

Consider a target distribution π defined on $(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$, with $\mathcal{B}(\mathbb{R}^d)$ Borel sets of $\mathbb{R}^d$ endowed with the Euclidean norm. Using SL to sample from π consists in identifying and simulating a stochastic process that converges almost surely to a random variable distributed as π [Alaoui et al., 2022, Montanari and Wu, 2023, Montanari, 2023]. A specific example is to consider, given $X \sim \pi$, the stochastic process $(Y_t)_{t\geq 0}$, called the *observation process*, defined by

$$Y_t = tX + \sigma W_t \,, \tag{4.1}$$

where $(W_t)_{t\geq 0}$ is a standard Brownian motion on $\mathbb{R}^d$, independent from X, and $\sigma > 0$[1]. The time-rescaled Y_t/t converges almost surely to X as $t \to \infty$. Moreover, if π admits a finite second order moment, we can show that the 2-Wasserstein distance between the distribution of Y_t/t and π is bounded by $\sigma\sqrt{d/t}$ [2], see Appendix C.2.2. Thus, for T large, Y_T/T is approximately distributed according to π.

Sampling from $(Y_t)_{t\geq 0}$ using directly (4.1) cannot be done in practice, precisely since it requires to first sample from π. Nevertheless, this issue can be overcome under the assumption that π admits a finite first moment. Indeed, in this case, $(Y_t)_{t\geq 0}$ solves the Stochastic Differential Equation (SDE),

$$\mathrm{d}Y_t = u_t(Y_t)\mathrm{d}t + \sigma \mathrm{d}B_t \,,\; Y_0 = 0 \,, \tag{4.2}$$

where $(B_t)_{t\geq 0}$ is a standard Brownian motion on $\mathbb{R}^d$ and $u_t(y) = \int_{\mathbb{R}^d} x q_t(x|y)\mathrm{d}x$ for any $y \in \mathbb{R}^d$, see [Brunick and Shreve, 2013, Corollary 3.7.]. The drift function u_t involves the conditional density of X given $Y_t = y \in \mathbb{R}^d$ defined for $x \in \mathbb{R}^d$ as

$$q_0(x|y) \propto \pi(x) \,, \tag{4.3}$$
$$q_t(x|y) \propto \pi(x)\mathrm{N}(x; y/t, \sigma^2/t\,\mathrm{I}_d) \,, \quad t > 0 \,. \tag{4.4}$$

In the literature, the *random* conditional expectation $u_t(Y_t) = \mathbb{E}[X|Y_t]$ is often referred to as the *Bayes estimator* of X given Y_t [Robbins, 1992, Saremi and Hyvärinen, 2019] or the *optimal denoiser* of Y_t [Montanari, 2023, Ghio et al., 2024]. Moreover, if π has finite second moment, one can show that the SDE (4.2) admits unique weak solutions [Liptser and Shiryaev, 1977, Theorem 7.6. & Remark 7.2.7.], see Appendix C.2.2. As a result, one can simulate $(Y_t)_{t\in[0,T]}$ for any $T > 0$ by integrating the SDE (4.2), while avoiding to first sample from π, if the drift function $(u_t)_{t\in[0,T]}$ (or an approximation of it) is known.

[1] Note that the authors originally considered $\sigma = 1$.
[2] This bound will be referred to as the *localization rate*.

Yet, the challenge of this sampling approach lies in the estimation of $(u_t)_{t\in[0,T]}$ without samples from π available *a priori*. At time $t = 0$, we have $u_0(y) = \mathbf{m}_\pi$. For any $t > 0$, u_t is linked to the *score* of the marginal distribution of Y_t, given by $p_t(y) = \int_{\mathbb{R}^d} \mathrm{N}(y; tx, \sigma^2 t\, \mathrm{I}_d)\mathrm{d}\pi(x)$, following Tweedie's formula given in Appendix C.1,

$$u_t(y) = y/t + \sigma^2 \nabla \log p_t(y) \ . \tag{4.5}$$

Therefore, sampling via SL can be reduced to a score estimation task, as noted by Montanari [2023], who also establishes a direct connection between the SL induced by the process (4.1) and variance-preserving diffusions [Song et al., 2021b], see Appendix C.2. In data-based sampling, *i.e.*, when samples $\{X^i\}_{i=1}^N$ from π are at the disposal of the user, u_t can be estimated using a parameterized model through a least-squares regression [Saremi and Hyvärinen, 2019, Montanari, 2023] following score matching techniques.

A few recent works started to discuss the density-based sampling task using SL or diffusions Montanari and Wu [2023], Huang et al. [2024a], Vargas et al. [2023a]. Here, we focus on the SL formalism and propose in Section 4.4 a sampling algorithm for arbitrary distributions based on a learning-free estimator of the denoiser. Before, we provide in Section 4.3 a working definition of SL with flexible denoising schedules.

4.3 SL with flexible denoising scheduling

Inspired by recent developments studying optimal noise scheduling in diffusion models [Nichol and Dhariwal, 2021, Kingma et al., 2021, Karras et al., 2022, Kingma and Gao, 2023], we consider a new class of explicit observation processes associated with flexible denoising schedules.

4.3.1 A more general observation process

For a possibly finite time horizon $T_{\mathrm{gen}} \in (0, \infty]$, we now consider the observation process $(Y_t^\alpha)_{t\in[0,T_{\mathrm{gen}})}$ defined by

$$Y_t^\alpha = \alpha(t)X + \sigma W_t \ , \tag{4.6}$$

where $(W_t)_{t\geq 0}$ is a standard Brownian motion on $\mathbb{R}^d$, independent from $X \sim \pi$, and $\alpha(t) = t^{1/2}g(t)$ with the following assumptions on g:

(a) $g \in \mathrm{C}^0([0, T_{\mathrm{gen}}), \mathbb{R}_+) \cap \mathrm{C}^1((0, T_{\mathrm{gen}}), \mathbb{R}_+)$;

(b) $g(t) \sim Ct^{\beta/2}$ as $t \to 0$, for some $\beta \geq 1$, $C > 0$;

(c) g is strictly increasing and $\lim_{t\to T_{\mathrm{gen}}} g(t) = \infty$.

Under these assumptions, the observation process (4.6) verifies $Y_0^\alpha = 0$ and $Y_t^\alpha/\alpha(t) \to X$ almost surely as $t \to T_{\mathrm{gen}}$. Moreover, by denoting π_t^α the distribution of $Y_t^\alpha/\alpha(t)$, one can show that for any $t \in (0, T_{\mathrm{gen}})$, the localization rate is now function of $g(t)$:

$$W_2(\pi, \pi_t^\alpha) \leq \sigma\sqrt{d}/g(t) \ , \tag{4.7}$$

where W_2 denotes the 2-Wasserstein distance. Proofs and refinements in the case where π is a Gaussian distribution are postponed to Appendix C.2.2. We recover the process (4.1) discussed in Section 4.2 with $g(t) = t^{1/2}$ and $T_{\mathrm{gen}} = \infty$.

As in the previous section, sampling from the observation process via (4.6) requires to first sample from π. We again bypass this issue by introducing an equivalent SDE. In Appendix C.2.2, we indeed show that $(Y_t^\alpha)_{t\in[0,T_{\mathrm{gen}})}$ solves

$$\mathrm{d}Y_t^\alpha = \dot\alpha(t)u_t^\alpha(Y_t^\alpha)\mathrm{d}t + \sigma \mathrm{d}B_t \ , \ Y_0^\alpha = 0 \ , \tag{4.8}$$

where $(B_t)_{t\geq 0}$ is a standard Brownian motion on $\mathbb{R}^d$ and the drift function u_t^α is defined for any $y \in \mathbb{R}^d$ by

$$u_t^\alpha(y) = \int_{\mathbb{R}^d} x q_t^\alpha(x|y)\mathrm{d}x \ . \tag{4.9}$$

Here, the conditional density of X given $Y_t = y$ is

$$q_0^\alpha(x|y) \propto \pi(x) \ , \tag{4.10}$$

$$q_t^\alpha(x|y) \propto \pi(x)\mathrm{N}(x; y/\alpha(t), \sigma^2/g(t)^2\, \mathrm{I}_d) \ , \ t \in (0, T_{\mathrm{gen}}) \ . \tag{4.11}$$

SL scheme	Hyper-parameters	$g(t)$
Geom-$\infty(\alpha_1)$	$\alpha_1 \geq 1$	$t^{\alpha_1/2}$
Geom(α_1, α_2)	$\alpha_1 \geq 1$, $\alpha_2 > 0$	$t^{\alpha_1/2}(1-t)^{-\alpha_2/2}$

Table 4.1: **Stochastic Localization schemes.**

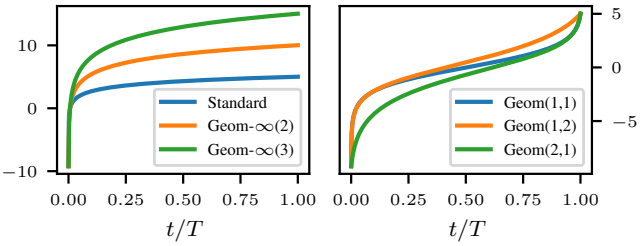


Figure 4.1: **Display of** $\log \mathrm{SNR}$ **for the localization schemes** Geom-∞ (**Left**) and Geom (**Right**).

Assuming further that π has finite second moment and that g satisfies technical conditions, (4.8) admits unique weak solutions, see Appendix C.2.2. Then again, if u_t^α is known, integrating (4.8) up to time $T \in (0, T_{\mathrm{gen}})$ is equivalent to sampling from $(Y_t^\alpha)_{t\in[0,T]}$. Then, $Y_T^\alpha/\alpha(T)$ is approximately distributed according to π for any T close enough to T_{gen}.

The challenge still lies in the estimation of the drift function u_t^α which verifies $u_0^\alpha(y) = \mathbf{m}_\pi$ at time $t = 0$, and which can be expressed, for any $t \in (0, T_{\mathrm{gen}})$, using the score of the marginal distribution $p_t^\alpha(y)$ given by $p_t^\alpha(y) = \int_{\mathbb{R}^d} \mathrm{N}(y; \alpha(t)x, \sigma^2 t\, \mathrm{I}_d)\mathrm{d}\pi(x)$. In Appendix C.2.2, we indeed show that

$$u_t^\alpha(y) = y/\alpha(t) + \{\sigma^2 t/\alpha(t)\}\nabla \log p_t^\alpha(y)\ , \tag{4.12}$$

recovering the previously mentioned link between score estimation and denoiser estimation.

Moreover, the score of p_t^α can also be written as an expectation over the same posterior distribution q_t^α but with a different observable

$$\nabla \log p_t^\alpha(y) = \tfrac{1}{\alpha(t)} \int_{\mathbb{R}^d} \nabla \log \pi(x) q_t^\alpha(x|y)\mathrm{d}x\ . \tag{4.13}$$

Hence, this suggests to consider an alternative expression of the SDE defined in (4.8), involving the expectation given in (4.13) instead of the denoiser function u_t^α [3]. We refer to Appendix C.2.3 for more details. However, we experimentally observed this approach was less stable than our original method, which is why we decided to consider the SDE (4.8).

4.3.2 Hyper-parameter selection and SNR profile

At first sight, determining appropriate choices for g, σ and T_{gen}, without introducing further complexity, can be challenging. Below, we propose interpretable choices, relying on the analysis of the *signal-to-noise ratio* of the observation process, which determines the speed at which the denoising process localizes on the target distribution.

An intuitive definition of the Signal-to-Noise Ratio (SNR) of the observation process (4.6) is, for any $t \in [0, T_{\mathrm{gen}})$,

$$\mathrm{SNR}(t) = \frac{\mathbb{E}\left[\|\alpha(t)(X - \mathbb{E}[X])\|^2\right]}{\mathbb{E}\left[\|\sigma W_t\|^2\right]} = \frac{R_\pi^2 g(t)^2}{\sigma^2 d}\ , \tag{4.14}$$

where $R_\pi^2 = \int_{\mathbb{R}^d} \|x - \mathbf{m}_\pi\|^2 \mathrm{d}\pi(x)$ is the scalar variance of π. The monotonicity of g ensures that the SNR is strictly increasing on $[0, T_{\mathrm{gen}})$, with $\lim_{t\to 0} \mathrm{SNR}(t) = 0$ (no information on π at all) and $\lim_{t\to T_{\mathrm{gen}}} \mathrm{SNR}(t) = \infty$ (no noise at all). Further, the scaling in time of the SNR schedule is solely determined by the function g, which also controls the time dependence in the localization rate (4.7). This observation leads us to fix $\sigma = R_\pi/\sqrt{d}$, assuming that R_π is known, to obtain a SNR profile $\mathrm{SNR}(t) = g^2(t)$ independent of the target distribution or the dimension[4]. In cases where the scalar variance of the target π is unknown, an estimator $\hat{R}_\pi$ can be used instead.

[3] Equation (4.13) and its consequence have been independently derived in the concurrent works Bortoli et al. [2024a], Akhound-Sadegh et al. [2024]. Both have been publicly available almost at the same time than the present paper.

[4] Note that our framework would be unchanged by taking $\sigma = 1$ and defining g up to a multiplicative constant.

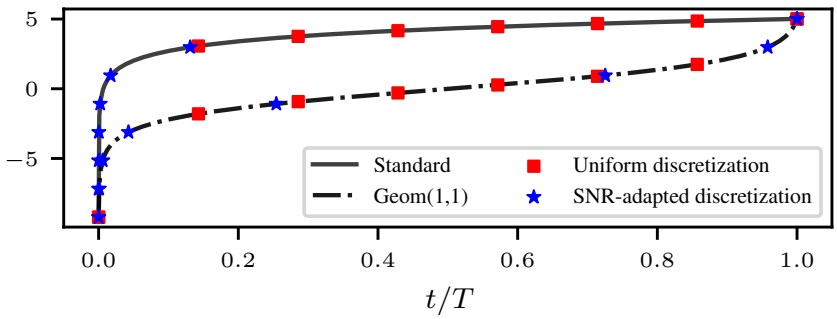


Figure 4.2: **SNR-adapted vs uniform time discretization for the schemes Standard and Geom$(1,1)$.** The uniform discretization leads to larger log-SNR differences between timesteps where the SNR increases rapidly.

The practical implementation of SL-based sampling requires also the choice of an effective time horizon $T < T_{\text{gen}}$ up to which we integrate the SDE (4.8), similarly to the maximum denoising time in diffusion models. It needs to be close enough to T_{gen} to consider $Y_T^\alpha/\alpha(T)$ as approximately distributed according to π and as small as possible to ensure computational efficiency. This trade-off is accounted for by selecting T based on reaching a predefined level of the logarithm of the SNR (log-SNR). Denoting by η a positive log-SNR threshold, we set $T = T_\eta$, where T_η is such that $\log \text{SNR}(T_\eta) = 2 \log g(T_\eta) = \eta$. In practice, the choice of η does not have a significant impact on performance (see Appendix C.4.3 Figure C.2).

Having reduced the set of hyper-parameters to tune, we investigate two main settings for the denoising schedule $g(t)$. Their main characteristics are summarized in Section 4.3.1 and examples of corresponding log-SNR profiles are plotted in Figure 4.1.

(a) **Asymptotic geometric schedule (Geom-∞)**, with $T_{\text{gen}} = \infty$ and $g(t) = t^{\alpha_1/2}$ for $\alpha_1 \geq 1$. This is a natural extension of the observation process discussed in Section 4.2 for which $\alpha_1 = 1$. We refer to it as the *standard* scheme.

(b) **Non-asymptotic geometric schedule (Geom)**, with $T_{\text{gen}} < \infty$ and $g(t) = (t/T_{\text{gen}})^{\alpha_1/2}(1-t/T_{\text{gen}})^{-\alpha_2/2}$, for $\alpha_1 \geq 1$ and $\alpha_2 > 0$. In practice, we only consider $T_{\text{gen}} = 1$. If $\alpha_1 = \alpha_2$, the log-SNR profile is similar to the one obtained in diffusion models via cosine noise scheduling [Nichol and Dhariwal, 2021]. Moreover, by taking $\alpha_2 < \alpha_1$, the profile becomes flatter near $t = 1$, similarly to the scheduling profiles presented in Kingma et al. [2021], Kingma and Gao [2023]. Following the framework of Albergo et al. [2023], $\{(1-t)^{\alpha_2/2} Y_t\}_{t\in[0,1]}$ is a stochastic interpolant between the Dirac mass at 0 and π.

The possibility to adapt the denoising schedule has not yet received attention in the application of SL for sampling. In the following section, we detail the design of our algorithm which adapts to any denoising schedule.

4.4 SLIPS sampling algorithm

In this section, we consider hyper-parameters σ, T and a denoising schedule g as given in Section 4.3.2 and expose our strategy for SL-based sampling. As already mentioned, sampling from π via SL consists in integrating the SDE (4.8) to obtain a realization $\{Y_t^\alpha\}_{t\in[0,T]}$ and set $Y_T^\alpha/\alpha(T)$ as an approximate sample from π. However, the SDE integration faces two main hindrances. First, the drift function $(u_t^\alpha)_{t\in[0,T]}$ defined in (4.9) is not tractable in general. Second, even if the drift function were known exactly, numerical integration incurs unavoidable discretization errors. We first tackle the latter issue in Section 4.4.1. We then present a Monte Carlo-based approach to tackle the estimation of u_t^α in Section 4.4.2. Finally, we present the key idea of our algorithm SLIPS in Section 4.4.3.

4.4.1 Handling the discretization error

Consider a time discretization of the interval $[0,T]$ defined by an increasing sequence of timesteps $(t_k)_{k=0}^K$ where $t_0 \geq 0$, $t_K = T$ and $K \geq 1$. We define a sequence of random variables $\{\tilde{Y}_{t_k}^\alpha\}_{k=0}^K$ approximating (4.8), defined for any $k \in \{0, \dots, K-1\}$ by the recursion

$$\tilde{Y}_{t_{k+1}}^\alpha = \tilde{Y}_{t_k}^\alpha + w_k u_{t_k}^\alpha(\tilde{Y}_{t_k}^\alpha) + \sigma\sqrt{\delta_k} Z_{k+1} \,, \tag{4.15}$$

where $\tilde{Y}^\alpha_{t_0} = 0$, $\delta_k = t_{k+1} - t_k$, $w_k = \alpha(t_{k+1}) - \alpha(t_k)$ and $(Z_k)_{k=1}^K$ is distributed according to the centered standard Gaussian distribution. Note that (4.15) results from an *Euler Maruyama* (EM) discretization scheme applied to (4.8), that corresponds for any $k \in \{0, \dots, K-1\}$ to the exact integration of the SDE

$$\mathrm{d}\tilde{Y}^\alpha_t = \dot{\alpha}(t) u^\alpha_{t_k}(\tilde{Y}^\alpha_{t_k})\mathrm{d}t + \sigma \mathrm{d}B_t \ , \ t \in [t_k, t_{k+1}] \ . \tag{4.16}$$

In practice, we rather start the integration from a timestep $t_0 > 0$ for reasons that we detail in the next section. We also adopt a time discretization adapted to the variations of the log-SNR on $[t_0, T]$. Define $\Delta_{\mathrm{SNR}} = \{\log \mathrm{SNR}(T) - \log \mathrm{SNR}(t_0)\}/K$, and for any $k \in \{0, \dots, K\}$, choose t_k such that

$$\log \mathrm{SNR}(t_k) = \log \mathrm{SNR}(t_0) + \Delta_{\mathrm{SNR}} k \ . \tag{4.17}$$

This discretization defines smaller step-sizes when the variation of $\log \mathrm{SNR}$ is high, *i.e.*, where the denoising process "accelerates" (see Figure 4.2). We observed that using uniform discretization leads to significantly greater numerical errors in our experiments, see Appendix C.4.

As described in the next section, we suggest to estimate $u^\alpha_{t_k}(\tilde{Y}^\alpha_{t_k})$ by Monte Carlo methods. Denoting by $\texttt{MC-Est}(u^\alpha_t(Y))$ an estimator of the drift function at time t evaluated at Y, we come to consider the joint sequence $\{(\tilde{U}^\alpha_{t_k}, \tilde{Y}^\alpha_{t_k})\}_{k=0}^K$ given by

$$\tilde{U}^\alpha_{t_k} = \texttt{MC-Est}(u^\alpha_{t_k}(\tilde{Y}^\alpha_{t_k})) \ , \tag{4.18}$$

$$\tilde{Y}^\alpha_{t_{k+1}} = \tilde{Y}^\alpha_{t_k} + w_k \tilde{U}^\alpha_{t_k} + \sigma\sqrt{\delta_k} Z_{k+1} \ , \tag{4.19}$$

where $(t_k)_{k=0}^K$ is the SNR-adapted time discretization of $[t_0, T]$ defined in (4.17). The initialization of $\tilde{Y}^\alpha_{t_0}$ is crucial and will be discussed in Section 4.4.3. We now detail our Monte Carlo estimation scheme defining $\tilde{U}^\alpha_{t_k}$.

4.4.2 Estimating the denoiser via MCMC

Monte Carlo estimation with posterior sampling. We aim to build accurate estimators of the quantities

$$u^\alpha_{t_k}(\tilde{Y}^\alpha_{t_k}) = \int_{\mathbb{R}^d} x \, q^\alpha_{t_k}(x|\tilde{Y}^\alpha_{t_k})\mathrm{d}x \ . \tag{4.20}$$

For ease of notation, we will denote the *random* posterior density $q_{t_k}(\cdot|\tilde{Y}^\alpha_{t_k})$ by μ_k in this section and recall that for any $k \in \{0, \dots, K\}$

$$\mu_k(x) \propto \pi(x)\mathrm{N}(x; \tilde{Y}^\alpha_{t_k}/\alpha(t_k), \sigma^2/g(t_k)^2 \, \mathrm{I}_d) \ . \tag{4.21}$$

One possible approach is to estimate $\tilde{U}^\alpha_{t_k}$ with Importance Sampling (IS), by choosing as instrumental distribution $\mathrm{N}(\tilde{Y}^\alpha_{t_k}/\alpha(t_k), \sigma^2/g(t_k)^2 \, \mathrm{I}_d)$. However, IS is known to suffer from large variance in high-dimension unless the proposal is closely adapted to π [Ceriotti et al., 2012]. This variance issue translates into sample-size requirements which may grow exponentially with d, see e.g., Chatterjee and Diaconis [2018].

Instead, we propose to approximately sample from μ_k with a MCMC algorithm, in the same fashion as Huang et al. [2024a], and compute $\tilde{U}^\alpha_{t_k}$ as the empirical mean of the obtained samples. More precisely, given a Markov chain $\{X^j_k\}_{j=1}^M$ targeting μ_k, we define $\tilde{U}^\alpha_{t_k} = (1/M)\sum_{j=1}^M X^j_k$.

In practice, we turn to the largely used *Metropolis-Adjusted Langevin Algorithm* (MALA) [Roberts and Tweedie, 1996] leveraging gradient-information of the target log-density. Nonetheless, to justify the use of MCMC here, sampling from μ_k has to be easier than the original problem of sampling from π. In particular, we establish in the next paragraph, conditions ensuring that μ_k is log-concave, implying a reasonable mixing time for MALA [Dwivedi et al., 2018].

Main issue in posterior sampling. To illustrate the main challenge in the sampling of μ_k, we focus on a particular setting. In the same spirit as [Saremi et al., 2024, Theorem 1], we consider the following assumption on π.

A 3 (Log-concavity outside a compact). *There exist $R > 0$ and $\tau > 0$ such that π is the convolution of μ and $\mathsf{N}(0, \tau^2 \, \mathrm{I}_d)$, where μ is a distribution whose scalar variance is bounded by dR^2, i.e., $\int_{\mathbb{R}^d} \|x - \mathbf{m}_\pi\|^2 \, \mathrm{d}\mu(x) \leq dR^2$.*

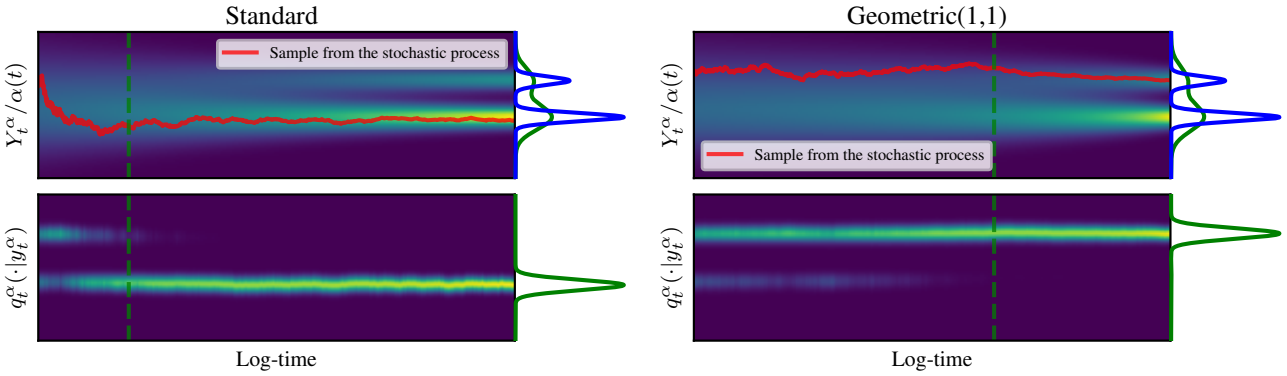


Figure 4.3: **Duality of log-concavity**: distribution of $Y_t^\alpha/\alpha(t)$ (**Up**) and $q_t^\alpha(\cdot|y_t^\alpha)$ (**Down**) for $t \in (0, T_{\text{gen}})$, where y_t^α is a realisation of the observation process (red line), for the standard scheme (**Left**) and the Geom$(1,1)$ scheme (**Right**). The target distribution π is a mixture of two 1D-Gaussian distributions $\mathrm{N}(-2/3, (0.05)^2)$ and $\mathrm{N}(4/3, (0.05)^2)$ with respective weights $2/3$ and $1/3$, which density is given by the blue line. The heat map represents the likelihood of the distributions and the green line on the right edge stands for the distributions taken at the time given by the dotted green line.

This assumption can be equivalently formulated as follows: for any random vector $X \sim \pi$, we have $X = U + G$, where $\mathbb{E}[\|U - \mathbf{m}_\pi\|^2] \leq dR^2$ and $G \sim \mathrm{N}(0, \tau^2 \mathrm{I}_d)$ is independent of U. In this setting, the scalar variance of π, defined in Section 4.3.2, verifies $R_\pi^2 = \mathbb{E}[\|U - \mathbf{m}_\pi\|^2] + d\tau^2$; therefore, we have the upper bound $R_\pi^2 \leq d(R^2 + \tau^2)$. In particular, this setting includes non log-concave distributions such as Gaussian mixtures, see Appendix C.3 for more details. Under Assumption **A**3, we obtain the following result whose formal statement is given in Appendix C.5.

Theorem 4. *Assume **A**3. Then, there exists a threshold time $t_{\text{q}} > 0$ (explicitly characterized in Appendix C.5) depending on g, d, R, and τ, such that for all $t > t_{\text{q}}$, the posterior $\mu_t = q_t^\alpha(\cdot|y)$ is strongly log-concave. Formally, for all $(x, y) \in \mathbb{R}^d \times \mathbb{R}^d$ and $t \in (0, T_{\text{gen}})$, the Hessian satisfies*

$$\nabla_x^2 \log q_t^\alpha(x|y) \preccurlyeq \zeta_q(t)\, \mathrm{I}_d,$$

where $\zeta_q : (0, T_{\text{gen}}) \to \mathbb{R}$ is a strictly decreasing function. This implies that the posterior becomes increasingly log-concave as t increases.

This result can be explained by the fact that, for large t_k, the Gaussian term dominates the target term in the posterior (4.21), since $\sigma^2/g(t_k)^2 \approx 0$ - which intuitively ensures increasing log-concavity. On the other hand, if t_k is close to 0, then μ_k is expected to be close to π, recalling that $q_0^\alpha = \pi$. In this case, for non log-concave target distributions satisfying **A**3, μ_k is non log-concave for small t_k. We illustrate this behaviour in Figure 4.3 for a bimodal Gaussian mixture: as depicted by the bottom row, the posterior distribution starts multi-modal (non log-concave) and eventually becomes unimodal (log-concave).

Therefore, for a wide variety of target distributions, our MALA-based posterior sampling approach on the time interval $[t_0, T]$ will fail at the very first steps if t_0 is chosen too small. We now explain how to bypass this issue in SLIPS.

4.4.3 Duality of log-concavity is all you need

Initialisation of the integration. Following the previous discussion, a reliable computation of our MCMC-based estimator given $\tilde{U}_{t_k}^\alpha$ requires to have $t_k > t_{\text{q}}$ in order to ensure log-concavity of the random posterior μ_k.

Thus, we aim to start the integration from a $t_0 > t_{\text{q}}$. To initialize the recursion (4.18) from t_0, one needs to sample the first iterate $\tilde{Y}_{t_0}^\alpha$ distributed according to $p_{t_0}^\alpha$. Given that $\tilde{U}_{t_0}^\alpha$ is a reliable estimator of the denoiser since $t_0 > t_{\text{q}}$, we can derive an approximation of the score $\nabla \log p_{t_0}^\alpha$, via Tweedie's formula (4.12). We use this estimate to sample from $p_{t_0}^\alpha$ via the *Unadjusted Langevin Algorithm* (ULA) [5] [Roberts and Tweedie, 1996]. In this initialization procedure, the successive evaluations of the score $\nabla \log p_{t_0}^\alpha$ in ULA can be computed by applying an inner loop of MALA on $q_{t_0}^\alpha$. This results in a Langevin-within-Langevin scheme, which is presented in Algorithm 2, and detailed in Appendix C.6.

[5] We do not use MALA since only the score of $p_{t_0}^\alpha$ is available.

For efficient sampling, ULA however requires a condition of log-concavity on $p^\alpha_{t_0}$ [Dalalyan, 2017, Durmus and Moulines, 2017]. In Theorem 5, we prove that this property is actually ensured for small values of t_0 under Assumption **A**3.

Theorem 5. *Assume **A**3. There exists $t_\mathrm{p} > 0$ (explicit in Appendix C.5), depending on g, d, R and τ such that if $t < t_\mathrm{p}$, p^α_t is strongly log-concave.*

We provide a formal version of this result in Appendix C.5. Intuitively, when t is small, the Gaussian noise in the observation process (4.6) overwhelms $\alpha(t)X$, which makes p^α_t log-concave. Note however that this log-concavity property is not verified for large t as p^α_t becomes as log-concave as π by the localization property, see (4.7). We illustrate this behaviour in the upper row of Figure 4.3.

Duality of log-concavity. Ideally, we would like to have (i) $t_0 > t_\mathrm{q}$ (to ensure that the estimation from $\tilde{U}^\alpha_{t_k}$ is correct) and (ii) $t_0 < t_\mathrm{p}$ (to ensure that $p^\alpha_{t_0}$ is log-concave). Under an additional assumption to **A**3, Theorem 6 demonstrates that such t_0 can exist. A proof of this result is given in Appendix C.5.

Theorem 6. *Assume **A**3, with $dR^2/\tau^2 < 2$. Then, $t_\mathrm{q} < t_\mathrm{p}$, where t_q and t_p are given in Theorem 4 and Theorem 5.*

Then, choosing t_0 boils down to finding a sweet spot where $p^\alpha_{t_0}$ and $q^\alpha_{t_0}$ are both approximately log-concave. Together, these requirements form what we call the *'duality of log-concavity'*. In Figure 4.3, we show that t_0 can be found for two different localization schemes when considering a Gaussian mixture that does not fit the assumption made in Theorem 6. Furthermore, we highlight in Appendix C.6.2 that a sweet spot generally exists for a wider variety of target distributions independently of the localization scheme.

In practice, we ensure the log-concavity of $q^\alpha_{t_0}$ first, as the Monte Carlo estimation of $u^\alpha_{t_0}$ is the starting point of the `SLIPS` algorithm.

4.4.4 Implementation of `SLIPS`.

By combining all the previous observations made above, we propose the *Stochastic Localization via Iterative Posterior Sampling* (`SLIPS`) algorithm, summarized in Algorithm 1, which we detail now.

The `SLIPS` algorithm has five inputs :

(a) the noising schedule in the observation process $t \mapsto \alpha(t)$ defined on a time interval $[0, T_\mathrm{gen})$,

(b) the *initial* integration time in (4.8): $t_0 \in (0, T_\mathrm{gen})$,

(c) the *final* integration time in (4.8): $T \in (t_0, T_\mathrm{gen})$, or equivalently $\eta > 0$ by log-SNR correspondence,

(d) the number of discretization steps in (4.8): $K \geq 1$,

(e) the number of samples for posterior sampling $M \geq 1$.

The algorithm can be decomposed in two main parts : (i) its initialization (summarized in Algorithm 2) and (ii) the run of the discretized SDE (4.18) via posterior sampling.

(i) The initialization of `SLIPS` consists in approximately sampling from $p^\alpha_{t_0}$. Since its score can be expressed via the posterior distribution $q^\alpha_{t_0}$, see (4.9) and (4.12), and $q^\alpha_{t_0}$ can itself be sampled from with MCMC methods (see Section 4.4.2), we propose to use a Langevin-within-Langevin procedure : each step of the outer loop corresponds to an iteration of the Langevin algorithm to sample from $p^\alpha_{t_0}$, and requires local MALA steps to estimate the score by sampling from the posterior $q^\alpha_{t_0}$. The exact computations are detailed in Appendix C.6. The final iterate of this algorithm is used as the initialization of the second part of `SLIPS`.

(ii) Once the initialization of `SLIPS` is done, we turn to the actual core part of `SLIPS`, which consists in running the discretized SDE given in (4.18). At step $k+1$ of this recursion (corresponding to the time t_{k+1} in the initial SDE), the denoiser term is approximated by running the MALA algorithm on the posterior distribution $q^\alpha_{t_k}$ conditioned on the k-th iterate of the recursion. Finally, the last iterate of this recursion is considered as an approximate sample from the target distribution.

Complexity of `SLIPS`. Note that running this procedure is meant to produce one sample from the target distribution. In practice, one may want to produce several samples. In this case, `SLIPS` can be easily parallelised by running simultaneously independent realizations of the procedure described above.

Algorithm 1 SLIPS

Input: α, t_0, η, K, M
▷ Set T and σ (see Section 4.3.2)
$T \leftarrow T_\eta$, $\sigma \leftarrow \hat{R}_\pi/\sqrt{d}$
▷ Define the SNR-adapted discretization (see Equation (4.17))
$(t_k)_{k=0}^K \leftarrow \mathrm{logSNR}^{-1}([t_0, T], K)$
▷ Initialize using Algorithm 2
$\tilde{Y}_{t_0}^\alpha \leftarrow \text{Algorithm2}(\alpha(t_0), t_0, \sigma, N, M)$
for $k \leftarrow 0$ **to** $K-1$ **do**
  ▷ Sample from the posterior μ_k with MALA (see Equation (4.21))
  $\{X_j^k\}_{j=1}^M \leftarrow \mathrm{MALA}(\mu_k, M)$
  ▷ Estimate the denoiser
  $\tilde{U}_{t_k}^\alpha \leftarrow \frac{1}{M}\sum_{j=1}^M X_j^k$
  ▷ Do a SDE discretization step
  $\delta_k = t_{k+1} - t_k$, $w_k = \alpha(t_{k+1}) - \alpha(t_k)$
  $\tilde{Y}_{t_{k+1}}^\alpha \sim \mathcal{N}\left(\tilde{Y}_{t_k}^\alpha + w_k \tilde{U}_{t_k}^\alpha, \sigma^2 \delta_k \, \mathrm{I}_d\right)$
end
Output: $\tilde{Y}_{t_K}^\alpha / \alpha(t_K)$

Algorithm 2 Langevin-within-Langevin Initialization

Input: $\alpha(t_0)$, t_0, σ, N, M
▷ Set the initial approximation and step-size
$Y^{(0)} \sim \mathcal{N}(0, \sigma^2 t_0 \, \mathrm{I}_d)$, $\lambda \leftarrow \sigma^2 t_0/2$
for $n \leftarrow 0$ **to** $N-1$ **do**
  ▷ Sample from $q_{t_0}^\alpha(\cdot \mid Y^{(n)})$ with MALA
  $\{X_j^{(n)}\}_{j=1}^M \leftarrow \mathrm{MALA}(q_{t_0}^\alpha(\cdot \mid Y^{(n)}), M)$
  ▷ Estimate the denoiser
  $U^{(n)} \leftarrow \frac{1}{M}\sum_{j=1}^M X_j^{(n)}$
  ▷ Estimate the score with Tweedie (see Equation (4.8))
  $\hat{s}_{t_0}^\alpha(Y^{(n)}) \leftarrow \frac{\alpha(t_0)U^{(n)} - Y^{(n)}}{\sigma^2 t_0}$
  ▷ Run a Langevin step on the marginal at t_0
  $Y^{(n+1)} \sim \mathcal{N}\left(Y^{(n)} + \lambda \hat{s}_{t_0}^\alpha(Y^{(n)}), 2\lambda \, \mathrm{I}_d\right)$
end
Output: $Y^{(N)}$

Moreover, we emphasize that the initialization of $\tilde{Y}_{t_0}^\alpha$ is the most challenging step in `SLIPS` as posterior sampling only gets easier afterwards, by Theorem 4. The computational cost of this initialization, involving a Langevin-within-Langevin procedure, remains however reasonable thanks to a persistent initialization of the ULA and MALA chains (see Appendix C.6). As such, only a few steps of each is needed in practice (see Appendix C.8.4).

Limitation of `SLIPS`. We highlight that t_0 is the only hyper-parameter in our algorithm that requires careful tuning. However, Figure C.5 and Figure C.6 in Appendix C.6.2 highlight moderate sensitivity to this hyper-parameter.

4.5 Related work

Score-based sampling with VI. Building upon score-based generative models and VI, recent works have proposed deep-learning approaches for density-based sampling. Those sampling schemes amount to discretized versions of denoising processes with a parameterized drift function. Two main settings may be distinguished. On one hand, the VI framework is seen as a stochastic optimal control problem involving the time-reversal of the denoising process, see e.g., Zhang and Chen [2022], Berner et al. [2023], Vargas et al. [2023b,a]. We refer to Richter et al. [2023] for a global overview of this approach and its extensions. On the other hand, another line of work has proposed parameterized extensions of Annealed Importance Sampling (AIS) Doucet et al. [2022], Geffner and Domke [2023], Vargas et al. [2024]. Since `SLIPS` is learning-free, these algorithms are not included in our numerical tests as it would be difficult to draw comparisons at equal computational budget.

Score-based sampling with Monte Carlo. Preliminary study of Monte Carlo score estimation was done by Huang et al. [2021], followed by Vargas et al. [2023b], who considered the Föllmer diffusion [Föllmer, 2005, 2006, Chen and Vanden-Eijnden, 2026] bridging δ_0 to π in a finite-time setting. However, their method relying on IS does not scale well with dimension, and suffers from numerical unstability. Note that it can be recasted as a particular example of our scheme Geom(1,1), via the stochastic interpolant analogy, see Section 4.3.2.

Closely related to this work, Huang et al. [2024a] recently proposed *Reverse Diffusion Monte Carlo* (RDMC), a sampling algorithm based on a Monte Carlo estimation of the drift of the time-reversal of a variance preserving diffusion. Many algorithmic choices in RDMC are similar to `SLIPS`, turning to Langevin for the drift estimation and using Langevin-within-Langevin for initialization. These authors also discuss the choice of the initial integration time, which plays a role similar to our t_0. Notably, under Lipschitz assumption on the score, they derive an upper bound on the overall complexity of RDMC, depending on this time. The present work complements the approach of Huang et al. [2024a]. We formalize the crucial trade-off in choosing t_0 according to our notion of "duality of log-concavity" (see Section 4.4.3), which is only briefly mentioned in Huang et al. [2024a].

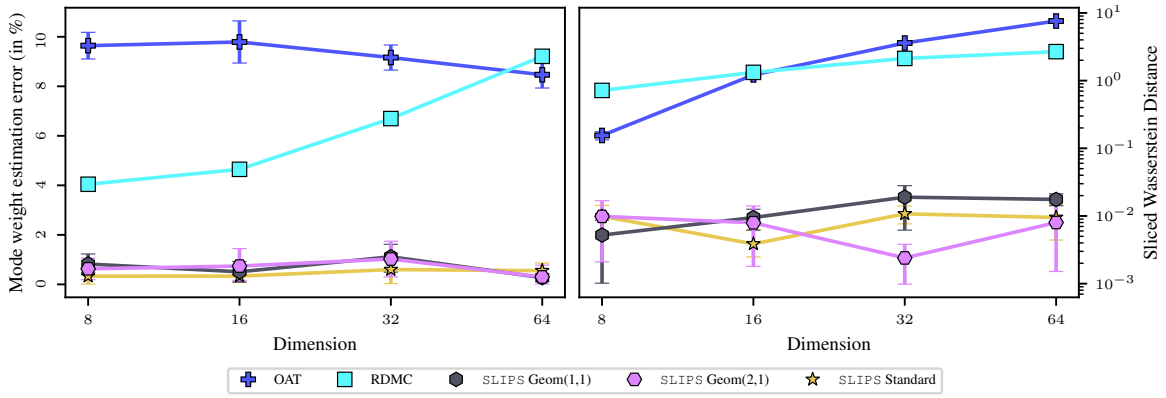


Figure 4.4: **Metrics when sampling a bimodal Gaussian mixture with d growing. Left**: Relative weight estimation error. **Right**: Sliced Wasserstein distance.

This focal point naturally arises from our numerical tests in high-dimension that allows us to assess the potential of SL-based sampling in practice, where Huang et al. [2024a] remained mostly theoretical. A precise comparison is provided in Appendix C.7.

Multi-Noise Measurements sampling. Recently, Saremi and Srivastava [2022] exploited a non-Markovian stochastic process to propose a novel data-based sampling scheme. The Multi-Noise Measurements (MNM) process $(Y^m)_{m=1}^M$ is defined by $Y^m = X + \sigma Z^m$ where $\sigma > 0$, $X \sim \pi$ and Z^m are independently sampled from $\mathrm{N}(0, \mathrm{I}_d)$. Here, the *measurement* Y^m plays the same role as the *observation* in standard SL[6]. In Saremi et al. [2024], the authors suggest to use this process in the density-based setting by computing $Y^{1:M}$ sequentially through the sampling of Y^m conditioned on $Y^{1:m-1}$. They show that those conditional distributions are increasingly log-concave, see [Saremi et al., 2024, Theorem 1], making their *Once-At-a-Time* (AOT) algorithm efficient. In contrast to our framework, Saremi et al. [2024] mainly assume that the score of the measurement process is analytically available. Although a Monte Carlo-based approach is proposed to estimate the score in realistic settings, results significantly degrade compared to the analytical case. As we detail in Appendix C.7, this can be explained by the fact that OAT faces but does not tackle the challenge of 'duality of log-concavity'.

4.6 Numerical experiments

In this section, we compare `SLIPS` against SMC, AIS, RDMC [Huang et al., 2024a] and OAT [Saremi et al., 2024]. Note that we deliberately omit an exhaustive comparison with standard MCMC methods, as they notoriously fail to sample from multi-modal distributions, see Appendix C.8.1. For `SLIPS`, we consider three different SL schemes : Standard, Geom(1,1) and Geom(2,1). Except for RDMC, all the algorithms are informed by the scalar variance R_π^2 of the target distribution (or an estimation). We tuned the hyper-parameters of each algorithm with coarse grid searches of similar size and similar computational budgets assuming access to an oracle distance metric to the target distribution (see details in Appendix C.8). For OAT, we estimated the intermediate scores with IS.

Toy target distributions. We first discuss standard target distributions including the *8-Gaussians* ($d = 2$), the *Rings* ($d = 2$) and the *Funnel* ($d = 10$) distributions. Details about their respective definitions are provided in Appendix C.8. Apart from Funnel, for which we provide results based on the sliced Kolmogorov-Smirnov (KS) distance as per Grenioux et al. [2023b], we compare the samples obtained with the algorithms against the ground truth using the entropic regularized 2-Wasserstein distance. The first four columns of Table 4.2 show that `SLIPS` is on par with most of its competitors on those toy distributions.

Bayesian Logistic Regression. Beyond toy distributions, we sample from the posterior of a Bayesian logistic regression model on two popular datasets : *Ionosphere* ($d = 34$) and *Sonar* ($d = 61$). More details on the design of the model are available in Appendix C.8. We evaluate the quality of sampling by computing the average predictive log-likelihood of the obtained samples. The last two columns of Table 4.2 show that, in these higher dimensions, `SLIPS` is slightly superior to its counterparts, especially OAT and RDMC.

[6]Indeed, both denoising processes have the same localization rate when $M = T$, see Appendix C.7 for more details.

Table 4.2: **Metrics when sampling toy distributions and posteriors from Bayesian logistic regression models.** Bold font indicates best results. The metric for 8-Gaussians and Rings ($d = 2$) is the entropic regularized 2-Wasserstein distance (with regularization hyper-parameter 0.05) (*the lowest, the best*), the metric for Funnel ($d = 10$) is the sliced Kolmogorov-Smirnov distance (*the lowest, the best*) and the metric for the Bayesian logistic regression on Sonar ($d = 34$) and Ionosphere ($d = 61$) datasets corresponds to the average predictive posterior log-likelihood (*the highest, the best*).

Algorithm	8-Gaussians (↓)	Rings (↓)	Funnel (↓)	Sonar (↑)	Ionosphere (↑)
AIS	1.10 ± 0.09	0.19 ± 0.02	0.037 ± 0.004	-111.04 ± 0.08	-87.92 ± 0.13
SMC	0.99 ± 0.16	0.28 ± 0.06	0.035 ± 0.005	-111.02 ± 0.13	-87.82 ± 0.16
OAT Saremi et al. [2024]	0.91 ± 0.09	0.18 ± 0.02	0.105 ± 0.005	-280.92 ± 1.34	-205.49 ± 0.61
RDMC Huang et al. [2024a]	1.01 ± 0.05	0.30 ± 0.01	0.082 ± 0.006	-129.88 ± 0.12	-109.84 ± 0.10
SLIPS Standard	0.76 ± 0.05	$\mathbf{0.19 \pm 0.01}$	$\mathbf{0.024 \pm 0.003}$	-109.25 ± 0.07	-86.65 ± 0.04
SLIPS Geom(1,1)	$\mathbf{0.74 \pm 0.12}$	0.20 ± 0.01	0.032 ± 0.002	$\mathbf{-109.14 \pm 0.09}$	$\mathbf{-86.32 \pm 0.10}$
SLIPS Geom(2,1)	0.75 ± 0.10	0.22 ± 0.02	0.040 ± 0.007	-110.24 ± 0.05	-86.78 ± 0.08

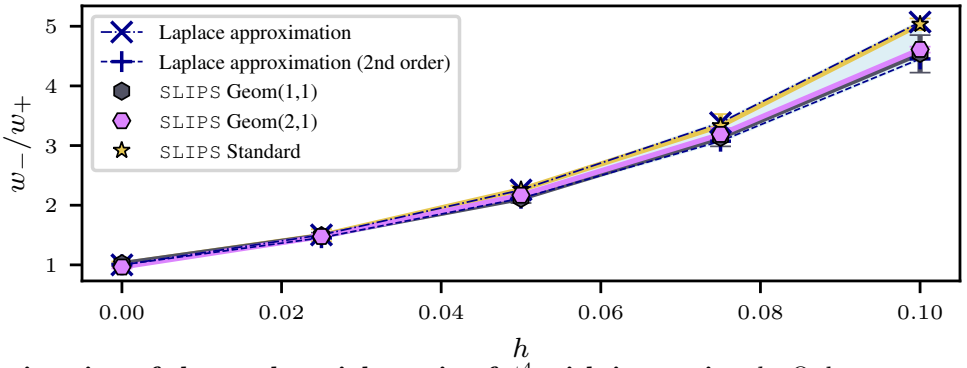


Figure 4.5: **Estimation of the mode weight ratio of ϕ^4 with increasing h.** Only SLIPS produced correct samples.

High-dimensional Gaussian Mixtures. As a challenging task, we seek to estimate the relative weight of a bimodal Gaussian mixture with modes $\mathrm{N}(x; -(2/3)\mathbf{1}_d, \Sigma)$ and $\mathrm{N}(x; (4/3)\mathbf{1}_d, \Sigma)$, where $\Sigma = 0.05\,\mathrm{I}_d$ and $\mathbf{1}_d$ is the d-dimensional vector with all components equal to 1, and respective weights $2/3$ and $1/3$. In our experiments, these weights are computed as $\hat{W}$ and $1 - \hat{W}$, where $\hat{W}$ is a Monte Carlo estimation of $\int_{\mathbb{R}^d} \mathbb{1}_{(-\infty,0)^d}(x)\mathrm{d}\pi(x)$. Here, we consider increasing values of d. Figure 4.4 (*left*) shows that SLIPS recovers the relative weight of the target at a 1% accuracy, even in high dimensions, whereas the other methods fail to give accurate estimates in most dimensions. In Figure 4.4 (*right*), we illustrate the superiority of SLIPS to capture the local properties of the distribution through the sliced Wasserstein distance to the target distribution. Due to high estimation error, AIS and SMC results are omitted in Figure 4.4 and displayed in Appendix C.8.

Field system ϕ^4 from statistical mechanics. Lastly, we sample the 1D ϕ^4 model, which was recently used as a benchmark in Gabrié et al. [2022]. At the chosen temperature, the distribution has two well distinct modes with relative weight that can be adjusted through a 'local-field' parameter h. We discretize this continuous model with a grid size of 100 (*i.e.*, $d = 100$). We estimate the relative weight between the two modes and compare the results with a Laplace approximation.

Figure 4.5 shows that the relative weight estimated by SLIPS lie between the 0-th and 2-nd order Laplace approximations. Due to high estimation error, the results from concurrent algorithms are omitted in Figure 4.5. We refer to Appendix C.8 for more details on the setting of this numerical experiment.

4.7 Discussion

In this paper, we introduced a Stochastic Localization (SL) scheme, that features flexible denoising schedule, in order to sample from any unnormalized target density. Relying on this framework, we proposed our algorithm, *Stochastic Localization via Iterative Posterior Sampling* (SLIPS) which leverages Monte Carlo estimation of the SL denoiser. Its design notably reveals the so-called "duality of log-concavity", that can be seen as a trade-off between sampling from the observation process and sampling from the SL posterior.

While our experiments focus on multi-modal distributions that are arguably simpler than those encountered in realistic applications, they serve as a necessary first step: any sampler should be able to handle these cases before being applied to more complex targets. We demonstrate that `SLIPS` successfully manages these distributions, whereas other methods fail, suggesting that `SLIPS` is a promising candidate for more challenging scenarios. Extending the evaluation to truly complex, high-dimensional multimodal distributions is left for future work.

As detailed in Sections 4.3.1 and 4.4, the current denoising schedule is parametric. Recent work on adaptive sampling methods [Zhou et al., 2016, Syed et al., 2021, 2022, 2024] shows that non-parametric adaptive schedules can significantly improve performance on multimodal targets. Some initial work has begun exploring similar adaptive scheduling for diffusion-based methods [Williams et al., 2024, Chen et al., 2025], and the adapted time discretization described in Section 4.4.1 can be seen as a first attempt in this direction. Developing more advanced, non-parametric adaptive schedules and systematically comparing them to the parametric approach remains an important direction for future work.

Finally, while Section 4.4.3 highlights sufficient conditions for the existence of the crucial t_0 parameter, important theoretical questions remain. In particular, we still lack rigorous guarantees for the Monte Carlo estimators produced by `SLIPS`, even when the duality of log-concavity holds. Although Huang et al. [2024a] provides convergence results in total variation distance under restrictive assumptions (e.g., Lipschitz continuity of the score), there is currently no framework for assessing Monte Carlo standard errors or deriving central limit theorems in this setting. Addressing these theoretical gaps constitutes a key avenue for future work.

Acknowledgements

The authors would like to thank Saeed Saremi and Francis Bach for insightful discussions during the course of this project. The work of LG and MG is supported by Hi! Paris. Part of the work of AD is funded by the European Union (ERC, Ocean, 101071601). Views and opinions expressed here are however those of the authors only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

Chapter 5

Learned Reference-based Diffusion Sampling for multi-modal distributions

5.1 Introduction

We consider the problem of sampling from a probability density known up to a normalizing constant. More precisely, we consider a target distribution $\pi \in \mathscr{P}(\mathbb{R}^d)$ with probability density given by $x \mapsto \gamma(x)/\mathcal{Z}$, where $\gamma : \mathbb{R}^d \to \mathbb{R}_+$ can be pointwise evaluated and the normalizing constant $\mathcal{Z} = \int_{\mathbb{R}^d} \gamma(x)\mathrm{d}x$ is intractable. This problem appears in a wide variety of applications such as Bayesian statistics [Liu and Liu, 2001, Kroese et al., 2011], statistical mechanics [Krauth, 2006] or molecular dynamics [Stoltz et al., 2010]. In particular, we are interested in sampling from *multi-modal* distributions, *i.e.*, distributions whose density admits multiple local maxima, called *modes*. Finding the modes of such distributions is a notoriously hard problem, yet, maybe surprisingly, even if the location of the modes is known, sampling π remains a very challenging problem [Noé et al., 2019, Pompe et al., 2020, Grenioux et al., 2023b]. In this work, we aim to address this specific issue and will assume that we have access to the location of the modes as prior information on π. However, we do not assume to have access *a priori* to ground truth samples from π.

Annealed MCMC. Markov Chain Monte Carlo (MCMC) samplers are among the most popular approaches for sampling. In particular, gradient-based methods based on discretizations of Langevin or Hamiltonian dynamics [Roberts and Tweedie, 1996, Neal et al., 2011, Hoffman and Gelman, 2014] are guaranteed to be efficient for high-dimensional target distributions that are log-concave or satisfy functional inequalities [Dalalyan, 2017, Durmus and Moulines, 2017]. However, when applied to multi-modal distributions, these MCMC methods suffer from high mixing time, that dramatically increases with the magnitude of the energy barriers between the modes, see Appendix D.9.3 for an illustration. To alleviate this issue, *annealing* strategies have been proposed. They consist in bridging an easy-to-sample *base* distribution π^{base} to the *target* distribution π via intermediate distributions $\{\pi_k\}_{k=0}^K$ with $K \geq 1$, $\pi_0 = \pi^{\text{base}}$ and $\pi_K = \pi$. This sequence should form a smooth path in the distribution space, so that sampling from π_{k+1} starting from samples from π_k is relatively simple. Typically, up to normalizing constants again, the densities of these annealed distributions are chosen as geometric interpolation between π^{base} and γ, and are sampled either sequentially [Neal, 2001, Del Moral et al., 2006] or in parallel [Geyer et al., 1991]. However, these approaches may be prone to mode switching, *i.e.*, the relative weight of the modes of the intermediate distributions may vary dramatically along the prescribed path [Woodard et al., 2009, Tawn et al., 2020, Syed et al., 2022]. Hence, recovering the proportions of the different modes of π is hard in practice.

Alternatively, the bridging distributions $\{\pi_k\}_{k=0}^K$ can be implicitly defined as the marginals of a diffusion process. A particular case of interest is the annealing path given by the time-reversal of a noising process, which is at the foundation of diffusion-based generative models [Sohl-Dickstein et al., 2015, Song et al., 2021b, Ho et al., 2020], and has been proved to be robust to mode switching [Phillips et al., 2024]. Following this approach, several diffusion-based samplers relying exclusively on Monte Carlo methods have recently shown promises for sampling from multi-modal distributions [Huang et al., 2024a,b, Grenioux et al., 2024]. However, these non-parametric approaches require the repeated estimation of an intractable drift at each draw and at each annealing step.

Annealed VI. Due to their ability to amortize inference, Variational Inference (VI) techniques are a popular alternative to MCMC methods. Given a class of parameterized (variational) distributions that are easy to sample from, they aim to find the optimal parameters that minimize a fixed discrepancy metric with respect to π. Standard VI settings involve mean-field approximations [Wainwright and Jordan, 2008], mixture models [Arenz et al., 2023], and more recently normalizing flows [Rezende et al., 2014, Papamakarios et al., 2021], that typically optimize a reverse Kullback-Leibler (KL) objective. However, these methods struggle with multi-modal target distributions as they suffer from mode collapse [Jerfel et al., 2021, Blessing et al., 2024].

To tackle this issue, it has been proposed to follow the annealing paradigm by considering an extended target distribution with marginals corresponding to the $\{\pi_k\}_{k=0}^K$. This approach has been explored following explicit bridging paths, *i.e.*, $\{\pi_k\}_{k=0}^K$ admit tractable unnormalized densities [Wu et al., 2020, Arbel et al., 2021, Geffner and Domke, 2021, Matthews et al., 2022, Doucet et al., 2022, Geffner and Domke, 2023, Vargas et al., 2024], but the resulting samplers may suffer from mode switching, as explained above. On the other hand, a recent class of annealed VI samplers using implicit diffusion-based paths has emerged [Tzen and Raginsky, 2019, Holdijk et al., 2023, Pavon, 2022, Zhang and Chen, 2022, Berner et al., 2023, Vargas et al., 2023a,b, Akhound-Sadegh et al., 2024, Phillips et al., 2024] and seems well-suited for multi-modal target distributions. As we will further evidence in the present work, the promising results obtained with the latter methods are nevertheless mitigated by the need for careful tuning of their hyperparameters, which requires access to ground truth samples.

Contributions Within this context, we propose the *Learned Reference-based Diffusion Sampler* (LRDS), a variational diffusion-based sampler specifically designed for multi-modal distributions, which extends works from Zhang and Chen [2022], Vargas et al. [2023a], Richter et al. [2023]:

- In the multi-modal setting, we highlight the sensitivity of previous variational diffusion-based methods with respect to their hyperparameters, which restrains their use in practice.
- To address this limitation, LRDS leverages the knowledge of the mode locations following two approaches: (a) GMM-LRDS relies on Gaussian Mixture Models and is well suited for a large variety of target distributions while being relatively lightweight, and (b) EBM-LRDS, a more computationally intensive scheme, takes advantage of Energy-Based Models for harder sampling problems.
- We show that GMM-LRDS and EBM-LRDS outperform previous diffusion-based samplers on challenging multi-modal settings.

Notation. For any measurable space $(\mathsf{X}, \mathcal{X})$, we denote by $\mathscr{P}(\mathsf{X})$ the space of probability measures defined on $(\mathsf{X}, \mathcal{X})$. Unless specified, if X is a topological space, $\mathcal{X}$ is defined as the Borel σ-field of X. For any $T > 0$, we denote by $\mathbf{C}_T = \mathrm{C}([0,T], \mathbb{R}^d)$ the space of continuous functions from $[0,T]$ to $\mathbb{R}^d$ endowed with the uniform topology; hence $\mathscr{P}(\mathbf{C}_T)$ is the set of continuous-time stochastic processes (or path measures) defined on $[0,T]$. For any $\mathbb{P} \in \mathscr{P}(\mathbf{C}_T)$, we denote by $\mathbb{P}^R$ its *time-reversal*, defined such that if $(X_t)_{t\in[0,T]} \sim \mathbb{P}$, then $(X_{T-t})_{t\in[0,T]} \sim \mathbb{P}^R$. For any $t \in [0,T]$, we denote by $\mathbb{P}_t$ the marginal distribution of $\mathbb{P}$ at time t. The density of the Gaussian distribution with mean $\mathbf{m} \in \mathbb{R}^d$ and covariance $\Sigma \in \mathbb{R}^{d\times d}$ is denoted by $x \mapsto \mathrm{N}(x; \mathbf{m}, \Sigma)$ and Δ_J refers to the J-dimensional simplex, *i.e.*, $\Delta_J = \{(w_j)_{j=1}^J \in [0,1]^J \,:\, \sum_{j=1}^J w_j = 1\}$ where $J \geq 1$.

Finally, for ease of understanding, we will denote *noising* diffusion processes by $(X_t)_{t\in[0,T]}$, driven by Brownian motion $(W_t)_{t\in[0,T]}$, while $(Y_t)_{t\in[0,T]}$ will refer to *denoising* diffusion processes, driven by Brownian motion $(B_t)_{t\in[0,T]}$. In this paper, Θ will refer to as the variational parameter space.

5.2 Reference-based Diffusion Sampling

5.2.1 Theoretical framework

We first recall the variational diffusion-based framework formulated by Richter et al. [2023] in continuous time, that generalizes the approaches from Zhang and Chen [2022] and Vargas et al. [2023a].

Time-reversed sampling process. Consider a general *noising* diffusion process defined on $[0,T]$ by the linear SDE

$$\mathrm{d}X_t = f(t)X_t\mathrm{d}t + g(t)\mathrm{d}W_t \;,\; X_0 \sim \pi \;, \tag{5.1}$$

where the horizon $T > 0$ is fixed, $(W_t)_{t\in[0,T]}$ is a standard d-dimensional Brownian motion, $g : [0,T] \to (0,\infty)$ and $f : [0,T] \to \mathbb{R}$. We denote by $\mathbb{P}^\star$ the path measure associated to the SDE (5.1). With appropriate choices of

f and g, this SDE admits explicit transition kernels and X_T is approximately, or exactly, distributed according to an easy-to-sample *base* distribution π^{base}.

In the case where $\mathbb{P}_T^\star = \pi^{\text{base}}$, we refer to this setting as an 'exact' noising scheme. One particular instance of this setting the Pinned Brownian Motion (PBM), considered by Tzen and Raginsky [2019], Zhang and Chen [2022], Vargas et al. [2023b], where $\pi^{\text{base}} = \delta_0$. Otherwise, when we only have $\mathbb{P}_T^\star \approx \pi^{\text{base}}$, we define this setting as an 'ergodic' noising scheme. A well-known example is the Variance Preserving (VP) noising process for which $\pi^{\text{base}} = \mathrm{N}(0, \sigma^2 \mathrm{I}_d)$ with $\sigma > 0$, proposed by Song et al. [2021b] for score-based generative models and Vargas et al. [2023a] for sampling. Numerical experiments presented in this paper focus on these two schemes to provide fair comparison with previous methods, but the presented approach is applicable for an arbitrary noising scheme. We refer to Appendix D.3 for more details.

Denote by $p_t^\star$ the density of $\mathbb{P}_t^\star$ w.r.t. the Lebesgue measure. Under mild assumptions on π, f and g, see e.g., Cattiaux et al. [2023], the time-reversal of the noising process $\mathbb{P}^\star$, *i.e.*, the distribution of $(X_{T-t})_{t\in[0,T]}$ and denoted by $(\mathbb{P}^\star)^R$, is associated to the SDE

$$\mathrm{d}Y_t = -f(T-t)Y_t\mathrm{d}t + g^2(T-t)\nabla \log p_{T-t}^\star(Y_t)\mathrm{d}t + g(T-t)\mathrm{d}B_t \ , \ Y_0 \sim \mathbb{P}_T^\star \ , \tag{5.2}$$

where $(B_t)_{t\in[0,T]}$ is another standard d-dimensional Brownian motion. By definition of the time-reversal, it holds that $Y_T \sim \pi$. Therefore, if we were able to simulate this diffusion process, we would obtain approximate samples from π. We therefore refer to $(Y_t)_{t\in[0,T]}$ as the *target process*. However, the scores $(\nabla \log p_t^\star)_{t\in[0,T]}$ involved in the drift function of (5.2) are intractable in general. In addition, these scores cannot be estimated via usual score matching techniques used in generative modeling [Hyvärinen, 2005, Vincent, 2011] since samples from π are not available *a priori* in the setting at hand. This point has been addressed by Richter et al. [2023] who alternatively proposed a general variational approach on path measure space, which requires the definition of a reference process as described next.

Variational reference-based approach. This approach relies on a *reference process* that is solution of SDE (5.1) with initial condition $X_0 \sim \pi^{\text{ref}}$, where $\pi^{\text{ref}} \in \mathscr{P}(\mathbb{R}^d)$ is chosen such that the marginal scores of the associated path measure $\mathbb{P}^{\text{ref}}$ are *tractable*. Similarly to π, we assume that the probability density of π^{ref} is known up to a multiplicative constant and is given by $x \mapsto \gamma^{\text{ref}}(x)/\mathcal{Z}^{\text{ref}}$, where $\gamma^{\text{ref}} : \mathbb{R}^d \to \mathbb{R}_+$ and $\mathcal{Z}^{\text{ref}} = \int_{\mathbb{R}^d} \gamma^{\text{ref}}(x)\mathrm{d}x$ is *not necessarily tractable*. Finally, for any $t \in [0,T]$, we denote the marginal scores $s_t^{\text{ref}} = \nabla \log p_t^{\text{ref}}$, with p_t^{ref} being the density of $\mathbb{P}_t^{\text{ref}}$ w.r.t. the Lebesgue measure. Based on the tractable reference scores, the SDE describing the target process (5.2) can be rewritten as

$$\mathrm{d}Y_t = -f(T-t)Y_t\mathrm{d}t + g^2(T-t)\{s_{T-t}^{\text{ref}}(Y_t) + \phi_{T-t}(Y_t)\}\mathrm{d}t + g(T-t)\mathrm{d}B_t \ , \ Y_0 \sim \mathbb{P}_T^\star \ , \tag{5.3}$$

where $\phi_t = \nabla \log p_t^\star / p_t^{\text{ref}}$ is now the only intractable term. This formulation of the time-reversed SDE has already been largely used in the diffusion model literature, especially for conditional generative models; see e.g., Dhariwal and Nichol [2021]. In this specific context, ϕ_t is called a guidance term. From a probabilistic perspective, ϕ_t can be interpreted as a Doob-h transform control, see Appendix D.9.1. Exploiting the reference-based formulation given by (5.3), Richter et al. [2023] propose to estimate the guidance terms $(\phi_t)_{t\in[0,T]}$ rather than the target scores $(\nabla \log p_t^\star)_{t\in[0,T]}$.

To this end, the authors consider the class of *variational* path measures $(\mathbb{P}^\theta)_{\theta\in\Theta} \subset \mathscr{P}(\mathbf{C}_T)$ where for any $\theta \in \Theta$, $\mathbb{P}^\theta$ is associated to the diffusion process

$$\mathrm{d}Y_t^\theta = -f(T-t)Y_t^\theta\mathrm{d}t + g^2(T-t)\{s_{T-t}^{\text{ref}}(Y_t^\theta) + \phi_{T-t}^\theta(Y_t^\theta)\}\mathrm{d}t + g(T-t)\mathrm{d}B_t \ , Y_0^\theta \sim \pi^{\text{base}} \ , \tag{5.4}$$

where $\phi^\theta : [0,T]\times\mathbb{R}^d \to \mathbb{R}^d$ is typically a neural network. In particular, choosing π^{ref} as a Gaussian distribution recovers exactly the previous VI methods *Path Integral Sampler* (PIS) [Zhang and Chen, 2022] and *Denoising Diffusion Sampler* (DDS) [Vargas et al., 2023a], see Table 5.1. Furthermore, Richter et al. [2023] propose to minimize the following LV-based objective

$$\mathcal{L}_{\text{LV}}(\theta) = \mathrm{Var}\left[\log\{(\mathrm{d}\mathbb{P}^\theta/\mathrm{d}(\mathbb{P}^\star)^R)(Y_{[0,T]}^{\hat\theta})\}\right] \ , \ Y_{[0,T]}^{\hat\theta} \sim \mathbb{P}^{\hat\theta} \ , \tag{5.5}$$

where $\hat\theta$ is a detached version of θ, meaning that gradients with respect to θ will not see $\hat\theta$. In contrast to the reverse KL divergence (previously used in PIS and DDS), which is notably known to suffer from mode collapse [Midgley et al., 2023b], the LV divergence [Nüsken and Richter, 2021] has the benefits to improve mode exploration, to avoid costly computations of the loss gradients and to have zero variance at the optimal solution [Richter et al., 2023, Section 2.3]. Finally, the LV loss (5.5) can be further explicited, as detailed in the following proposition.

Table 5.1: **Connection to prior works.** Here, $E^\psi : [0,T] \times \mathbb{R}^d \to \mathbb{R}$ refers to a neural network.

Method	Noising	$\gamma^{\text{ref}}(x)$	$\mathcal{Z}^{\text{ref}}$	$s_t^{\text{ref}}(x)$	Reference parameters
PIS [Zhang and Chen, 2022]	PBM(σ)	$\mathrm{N}(x; 0, \sigma^2 T \mathrm{I}_d)$	1	$-x/\sigma^2(T-t)$	σ (*tuned*)
DDS [Vargas et al., 2023a]	VP(σ)	$\mathrm{N}(x; 0, \sigma^2 \mathrm{I}_d)$	1	$-x/\sigma^2$	σ (*tuned*)
GMM-LRDS (Section 5.3.2)	arbitrary	$\sum_{j=1}^J w_j \mathrm{N}(x; \mathbf{m}_j, \Sigma_j)$	1	analytical	$\{w_j, \mathbf{m}_j, \Sigma_j\}_{j=1}^J$ (learned)
EBM-LRDS (Section 5.3.3)	arbitrary	$\exp(-E^\psi(t=0,x))$	unknown	$-\nabla_x E^\psi(t,x)$	ψ (learned)

Proposition 7. *Assume that $\mathbb{P}_T^\star = \mathbb{P}_T^{ref} = \pi^{base}$ and there exists $\theta^\star \in \Theta$ such that $\phi_t^\theta = \phi_t$. Then, the loss defined in (5.5) achieves optimal solution at $\theta^\star$ and, setting $\varrho = \log(\gamma^{ref}/\gamma)$, it simplifies as*

$$\mathcal{L}_{\text{LV}}(\theta) = \mathrm{Var}\left[\int_0^T \frac{g^2(T-t)}{2} \left\|\phi_{T-t}^\theta(Y_t^{\hat\theta})\right\|^2 \mathrm{d}t + \int_0^T g(T-t)\phi_{T-t}^\theta(Y_t^{\hat\theta})^\top \mathrm{d}B_t + \varrho(Y_T^{\hat\theta})\right] . \tag{5.6}$$

This result is an adaptation of [Richter et al., 2023, Lemma 3.1], which proof is restated in Appendix D.4.1 for completeness. We also emphasize that the assumption made in Proposition 7 is only needed for ergodic noising schemes. In practice, Richter et al. [2023] only consider two specific versions of this continuous-time variational loss, where π^{ref} is determined by the PIS and DDS settings. In the following, we will refer to these extensions of PIS and DDS as LV-PIS and LV-DDS. In contrast, we keep a general perspective, by letting π^{ref} completely arbitrary, and describe our proposition of practical implementation of a Reference-based Diffusion Sampler (RDS) in this context.

5.2.2 Reference-based Diffusion Sampling in practice

Discrete-time setting. In practice, the parameterized process $(Y_t^{\hat\theta})_{t\in[0,T]}$ cannot be simulated exactly and SDE (5.4) can only be numerically solved with a small step-size using, for example, the Euler-Maruyama (EM) or Exponential Integration (EI) [Durmus and Moulines, 2015] discretization schemes, see Appendix D.2 for additional details. Consider a time discretization of the interval $[0,T]$ given by an increasing sequence of timesteps $\{t_k\}_{k=0}^K$ such that $t_0 = 0$, $t_K = T$ and $K \geq 1$. Given an *arbitrary* reference distribution π^{ref}, we propose to approximate the continuous-time objective defined in (5.6) by the following *discrete time* RDS objective

$$\mathcal{L}_{\text{RDS}}(\theta) = \mathrm{Var}\left[\sum_{k=0}^{K-1} w_k \phi_{T-t_k}^\theta(Y_k)^\top \left\{\phi_{T-t_k}^{\hat\theta}(Y_k) - \frac{1}{2}\phi_{T-t_k}^\theta(Y_k)\right\} + \sum_{k=0}^{K-1} \sqrt{w_k}\phi_{T-t_k}^\theta(Y_k)^\top Z_k + \varrho(Y_K)\right] , \tag{5.7}$$

where $\{Z_k\}_{k=0}^{K-1}$ are independently distributed according to $\mathrm{N}(0, \mathrm{I}_d)$, $\{Y_k\}_{k=0}^K$ is recursively defined by $Y_0 \sim \pi^{\text{base}}$ and for any $k \in \{0, \dots, K-1\}$,

$$Y_{k+1} = a_k Y_k + b_k\{s_{T-t_k}^{\text{ref}}(Y_k) + \phi_{T-t_k}^{\hat\theta}(Y_k)\} + \sqrt{c_k} Z_k , \tag{5.8}$$

with $\hat\theta$ being a detached version of θ and $\{w_k, a_k, b_k, c_k\}_{k=0}^{K-1}$ being *tractable* coefficients that depend on the choice of discretization and noising schemes. In particular, if K is sufficiently large, $\{Y_k\}_{k=0}^K$ has approximately the same distribution as $\{Y_{t_k}^{\hat\theta}\}_{k=0}^K$. We refer to Appendix D.4.2 for a detailed explanation on the computation of our loss and explicit values of the coefficients. We emphasize that this variational objective can be implemented with any reference distribution. In comparison, the practical loss functions[1] exhibited by Richter et al. [2023] are only designed for LV-PIS and LV-DDS, and may be seen as particular instances of (5.7).

Training and sampling with RDS. The training stage of RDS consists in running N iterations of Stochastic Gradient Descent (SGD) to minimize the discrete-time objective (5.7), for some $N \geq 1$. Like other variational diffusion-based objectives, the RDS loss is *simulation-based*, meaning that, the whole process $\{Y_k\}_{k=0}^K$ defined in (5.8) must be simulated at each SGD step. The complete training procedure of RDS is summarized in Algorithm 5, stated in Appendix D.1. After training, the final parameter $\theta^\star$ is fixed, and approximate samples from π are obtained by simulating $\{Y_k\}_{k=0}^K$ from $\phi^{\theta^\star}$, using (5.8), and taking Y_K.

[1] Although the authors do not provide any expression of a discrete-time version of (5.6) in their paper, a discrete-time loss is implemented for LV-PIS and LV-DDS in their official codebase available at https://github.com/juliusberner/sde_sampler.

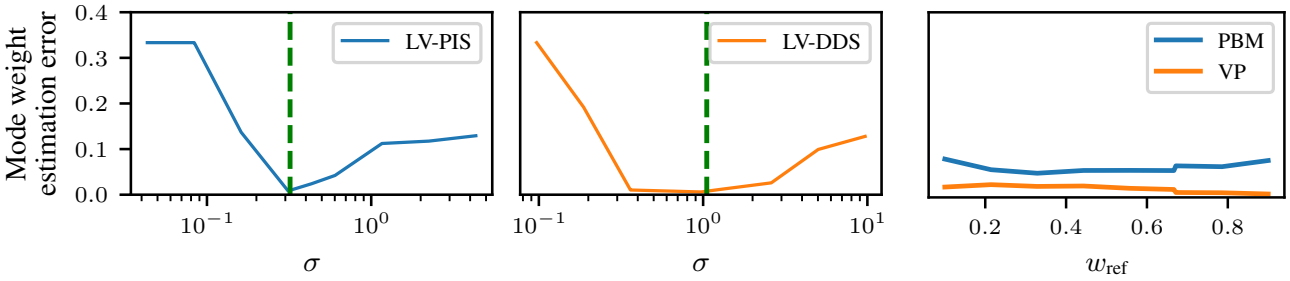


Figure 5.1: **Illustration of the decisive role of the reference distribution**. Here, we target a 16-dimensional Gaussian mixture with two modes, that have respective weights $w = 2/3$ and $1 - w = 1/3$, and display the estimation error of w with different methods. **(Left)**: Results for LV-PIS and LV-DDS when varying the value of their hyperparameter σ (which directly determines π^{ref} as shown in Table 5.1). The green dotted line represents the optimal variance for Gaussian approximation of π, see Appendix D.9.2 for related computations. **(Right)** Results for RDS in PBM and VP settings when setting π^{ref} as a Gaussian mixture with the same modes as π, but w is replaced by w_{ref}. Details on the design of this experiment are given in Appendix D.8.1.

5.3 Learned RDS for multi-modal distributions

In this section, we highlight the crucial role played by the reference distribution π^{ref} in RDS when targeting multi-modal distributions. We first show in Section 5.3.1 the limitations of previous variational reference-based methods and give intuition on how π^{ref} should be chosen. Based on this observation, we introduce *Learned Reference-based Diffusion Sampler* (LRDS), a novel diffusion-based sampler defined as a practical instance of RDS, where π^{ref} is learned from prior knowledge on π, namely the location of their modes. We present two approaches based on Gaussian Mixture Models (GMM-LRDS), see Section 5.3.2, and Energy-Based Models (EBM-LRDS), see Section 5.3.3.

5.3.1 On the importance of the reference distribution

Choosing a Gaussian distribution for π^{ref} ensures that the entire reference process is Gaussian. Therefore, the scores $(s_t^{\text{ref}})_{t\in[0,T]}$ can be analytically computed. We will refer to this case as *Gaussian RDS* (G-RDS). In this setting, previous variational reference-based methods (PIS, DDS and their log-variance version) focused on using isotropic centered Gaussian distributions as reference distributions (see Table 5.1), *i.e.*, $\pi^{\text{ref}} = \mathrm{N}(0, \sigma^2 \mathrm{I}_d)$ where the hyperparameter is $\sigma \in (0, \infty)$. This parameter is fixed before running the optimization over θ and a priori disconnected from the target distribution. To assess the sensitivity of the algorithms with respect to the hyperparameter σ, we run LV-DDS and LV-PIS on a simple multi-modal distribution – a 16-dimensional bi-modal Gaussian mixture with two distinct weights – and examine the accuracy of estimation of these weights (see Figure 5.1, left). We observe that the performance is highly dependent on the value of σ, with an optimal accuracy achieved when the isotropic variance is directly related to statistics of the target distribution; see Appendix D.9.2 for details. However, it is hard to estimate this quantity without access to ground truth samples.

Towards a multi-modal reference distribution. The previous experiment suggests that choosing a reference distribution that is close to the target distribution improves the performance of the VI sampler. To validate this intuition, we target the same Gaussian mixture as above using RDS with PBM and VP noising schemes, setting π^{ref} as a Gaussian mixture with the same components as the target with their *relative weights* as hyperparameter (see Figure 5.1, right). We find that this design of π^{ref}, with a well adjusted Gaussian mixture reference, enables RDS to robustly recover the information of the relative weight of the target modes. These observations motivate us to adopt the following paradigm for RDS: given a multi-modal target distribution π, we aim to design π^{ref} close to π, sharing *the same multi-modality characteristics*. Below, we present *Learned Reference-based Diffusion Sampler* (LRDS), a complete sampling methodology relying on the RDS framework that leverages prior knowledge on the target modes to learn a well-suited reference distribution.

LRDS pipeline. Our algorithm LRDS proceeds in three main steps:

(a) **Obtain *reference* samples**. Given a standard MCMC sampler which targets π, such as the *Metropolis-Adjusted Langevin Algorithm* (MALA) [Roberts and Tweedie, 1996], we simulate multiple Markov chains that

are initialized in the target mode locations. We refer to the obtained samples as the *reference samples* and denote by $\hat{\pi}^{\text{ref}}$ the corresponding empirical distribution. As recalled in Appendix D.9.3, in presence of high energy barriers between the modes, $\hat{\pi}^{\text{ref}}$ fails at estimating the global energy landscape. Moreover, $\hat{\pi}^{\text{ref}}$ does not admit a density w.r.t. the Lebesgue measure.

(b) **Define the reference process.** In this stage, we aim at (i) learning π^{ref} as a continuous approximation of $\hat{\pi}^{\text{ref}}$, and (ii) computing the marginal scores at times $\{T - t_k\}_{k=0}^{K-1}$ of the induced reference process $\mathbb{P}^{\text{ref}}$, *i.e.*, the noising process defined by SDE (5.1), where $X_0 \sim \pi^{\text{ref}}$. In practice, we set π^{ref} either as a Gaussian Mixture Model, which relies on a light parameterization, or as an Energy-Based Model, which is more expressive at the cost of higher computational budget.

(c) **Run the RDS variational optimization.** Once the reference distribution is learned and the reference scores are computed, we run the RDS procedure described in Section 5.2.2.

In the next sections, we describe two practical implementations of stage (b) in LRDS using different parameterizations of the reference distribution.

5.3.2 Gaussian Mixture Model LRDS

We first propose to set π^{ref} as a *Gaussian mixture* to integrate multi-modality in RDS. In particular, we have $\mathcal{Z}^{\text{ref}} = 1$ and any reference density is parameterized as $\gamma^{\text{ref}}(\cdot) = \sum_{j=1}^{J} w_j \mathrm{N}(\cdot; \mathbf{m}_j, \Sigma_j)$ where $J \geq 1$ is the number of reference modes, $\{w_j\}_{j=1}^{J} \in \Delta_J$ are the reference weights, $\mathbf{m}_j \in \mathbb{R}^d$ and $\Sigma_j \in \mathbb{R}^{d \times d}$ are respectively the mean and the covariance of the j-th reference mode. In this setting, each marginal of $\mathbb{P}^{\text{ref}}$ is also a Gaussian mixture whose parameters can be simply deduced from γ^{ref}, making their score fully tractable, see Appendix D.2. Therefore, the only difficulty at stage (b) lies in the estimation of the reference parameters $\psi = \{w_j, \mathbf{m}_j, \Sigma_j\}_{j=1}^{J}$ for some $J \geq 1$ fixed in advance. If $J > 1$, we refer to this version of the algorithm as *Gaussian Mixture Model LRDS* (GMM-LRDS). In this case, we learn ψ by running the Expectation Minimization (EM) algorithm [Dempster et al., 1977] with samples from $\hat{\pi}^{\text{ref}}$. If $J = 1$, LRDS reduces to *Gaussian LRDS* (G-LRDS), and we learn parameters $(\mathbf{m}_1, \Sigma_1)$ by Maximum Likelihood (ML) estimation. In practice, we observe that setting J equal or larger to the number of target modes can lead to better performance. We summarize the whole GMM-LRDS methodology in Algorithm 6, provided in Appendix D.1, and illustrate in Figure 5.2 the behaviours of GMM-LRDS and G-LRDS for a bi-modal target distribution. Our results show that choosing a Gaussian reference distribution leads to mode collapse, while defining π^{ref} as in GMM-LDRS enables RDS to recover the relative weights accurately.

5.3.3 Energy-Based Model LRDS

Although GMM-LRDS efficiently introduces multi-modality in the reference distribution while conserving the tractability of the reference scores, we anticipate that in some cases, a Gaussian mixture may not provide a correct approximation of $\hat{\pi}^{\text{ref}}$, regardless of the number of reference modes J, see Appendix D.9.4 for illustrations. To provide more flexibility, we propose a second version of LRDS where the reference distribution is parameterized by an *Energy-Based Model* (EBM), *i.e.*, $\gamma^{\text{ref}}(x) = \exp(-E^{\psi}(x))$ where $E^{\psi} : \mathbb{R}^d \to \mathbb{R}$ is a neural network with parameters ψ and the normalizing constant $\mathcal{Z}^{\text{ref}} = \int_{\mathbb{R}^d} \exp(-E^{\psi}(x))\mathrm{d}x$ is *intractable*. In contrast to GMM-LRDS, the corresponding reference scores cannot be computed analytically anymore; hence, we suggest to *estimate* them with neural networks. In the following, we will denote π^{ref} by p^{ψ} to insist on its parametric nature.

At first sight, it seems natural to learn the density γ^{ref} and the scores $(s_t^{\text{ref}})_{t\in[0,T]}$ independently from each other. Indeed, following the diffusion model literature [Song et al., 2021b, Karras et al., 2022], the reference scores could be easily learned using a *Score Matching* (SM) objective with samples from $\hat{\pi}^{\text{ref}}$ [Hyvärinen, 2005, Vincent, 2011]. On the other hand, one could learn p^{ψ} by maximizing the standard *Maximum Likelihood* (ML) objective $\mathcal{L}^{\text{ML}} : \psi \mapsto \mathbb{E}\left[\log p^{\psi}(X)\right]$, where $X \sim \hat{\pi}^{\text{ref}}$, with corresponding gradient given by $\psi \mapsto \mathbb{E}[\nabla_{\psi} E^{\psi}(X^-)] - \mathbb{E}[\nabla_{\psi} E^{\psi}(X^+)]$, where $X^- \sim p^{\psi}$ and $X^+ \sim \hat{\pi}^{\text{ref}}$. Nevertheless, computing gradients of objective $\mathcal{L}^{\text{ML}}$ requires to sample from p^{ψ} (*negative sampling*), which is a well known hurdle in EBM training since p^{ψ} is expected to be as multi-modal as $\hat{\pi}^{\text{ref}}$, and thus π.

Therefore, we rather suggest to learn a path of parametric distributions $(p_t^{\psi})_{t\in[0,T]}$ as the marginal distributions of $(\hat{X}_t^{\text{ref}})_{t\in[0,T]}$, defined as the noising process induced by SDE (5.1), starting at $\hat{X}_0^{\text{ref}} \sim \hat{\pi}^{\text{ref}}$. To do so, we set $(p_t^{\psi})_{t\in[0,T]}$ as a *multi-level* EBM, *i.e.*, $p_t^{\psi}(x) = \exp(-E^{\psi}(t,x))/\mathcal{Z}_t^{\psi}$, where $E^{\psi} : [0,T] \times \mathbb{R}^d \to \mathbb{R}$ is now a *time-dependent* neural network with parameters ψ and the normalizing constants $\mathcal{Z}_t^{\psi} = \int_{\mathbb{R}^d} \exp(-E^{\psi}(t,x))\mathrm{d}x$

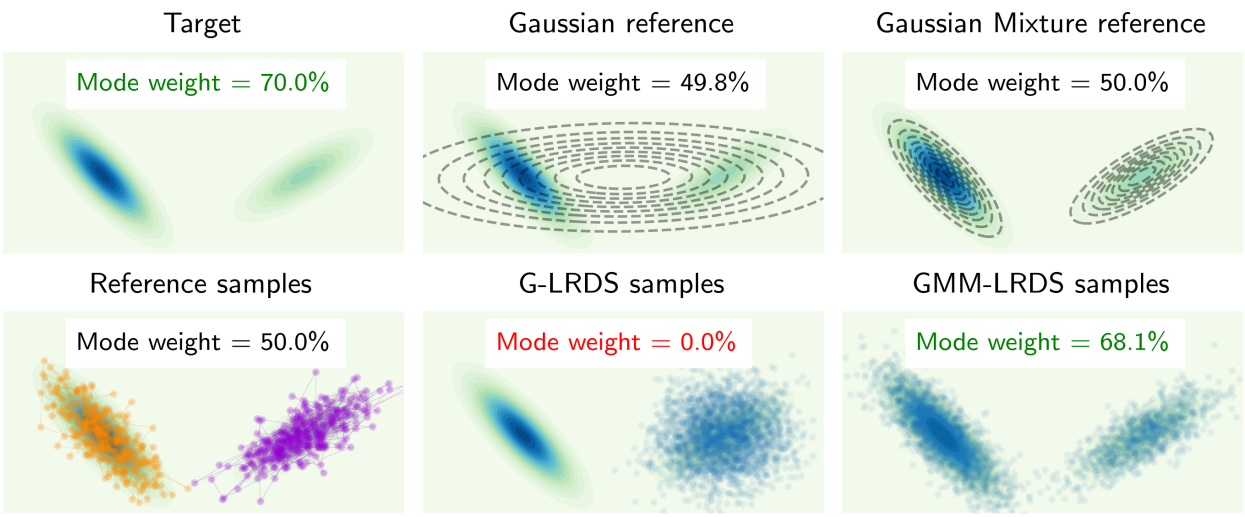
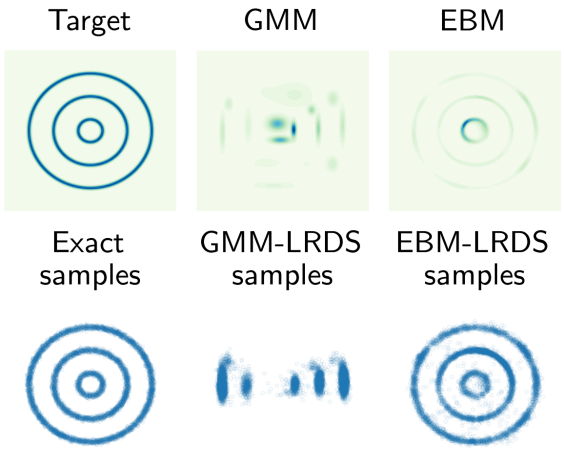


Figure 5.2: **Comparison between G-LRDS and GMM-LRDS.** Here, the target distribution is the same 16-dimensional Gaussian mixture as in Figure 5.1 (top left), see Appendix D.8.1 for more details. For illustration purpose, projections along the first two coordinates are used. In each cell, the value of 'Mode weight' refers to the effective weight of the left mode. Reference samples (bottom left) are obtained by running MALA sampler initialized in both target modes: each color (orange/purple) depicts one MALA Markov chain. In particular, none of them mixes between the modes. Running G-LRDS with an ML-estimated Gaussian reference (top middle) leads to mode collapse (bottom middle). Conversely, GMM-LRDS with an EM-estimated Gaussian mixture reference (top right) appropriately recovers the target distribution and the true mode proportions (bottom right).

are still intractable. In the RDS framework, this amounts to consider $\gamma^{\text{ref}}(x) = \exp(-E^{\psi}(0,x))$ and $s_t^{\text{ref}}(x) = -\nabla_x E^{\psi}(t,x)$. To learn this multi-level EBM, we propose to maximize the integrated ML objective $\psi \mapsto \int_0^T \mathbb{E}[\log p_t^{\psi}(\hat{X}_t^{\text{ref}})]\mathrm{d}t$. In this case, we can leverage the correlations between the single-level EBMs to alleviate their individual issue of negative sampling, a strategy that has already been investigated in several works [Gao et al., 2021, Zhu et al., 2024, Zhang et al., 2023].

Here, since $(\hat{X}_t^{\text{ref}})_{t\in[0,T]}$ defines a path of increasingly simpler distributions, *annealed MCMC samplers* can be conveniently used to sample from the multi-level EBM densities. This allows us to kill two birds in one stone: (i) we obtain negative samples for each single-level EBM, which is needed to compute the gradient of the ML objective, and (ii) we overcome the individual sampling issues at every level thanks to annealing. This method is completely detailed in Algorithm 14 of Appendix D.6 together with previous literature on multi-level EBMs. Morever, Appendix D.5 provides details on annealed MCMC samplers.



This version of LRDS called *Energy-Based Model LRDS* (EBM-LRDS) is summarized in Algorithm 7 of Appendix D.1. In Figure 5.3, we illustrate the superiority of EBM-LRDS over GMM-LRDS for a target distribution π that exhibits complex geometry. While the GMM fails at capturing the local energy landscape of π, which results in a poor performance of GMM-LRDS, EBM-LRDS captures well the target distribution since the reference EBM recovers the complex geometry of the target.

Figure 5.3: **Comparison between GMM-LRDS and EBM-LRDS in a multi-modal setting**. Here, we target the 2-dimensional Rings distribution, which has 3 *unbalanced* modes represented by the rings. **(Left)**: Target density (top) and exact samples (bottom). **(Middle)**: 16-component GMM reference distribution (top) and resulting GMM-LRDS samples (bottom). **(Right)**: EBM reference distribution (top) and resulting EBM-LRDS samples (bottom).

5.4 Related works

In the past years, numerous parametric methods have been proposed to approximate a diffusion process $(Y_t)_{t\in[0,T]} \sim \mathbb{P}$ induced by an SDE with intractable drift, such as (5.2), bridging an easy-to-sample distribution π^{base} to a target distribution π. Based on deep learning techniques, they suggest to learn a path measure $\mathbb{P}^\theta$, induced by a *neural* SDE that admits a neural network (parameterized by $\theta \in \Theta$ to be optimized) as drift function. Below, we review two main classes of such algorithms.

Variational diffusion-based methods. To optimize θ, several approaches have adopted a variational formulation on path measure space, which consists in minimizing a divergence-based loss with samples from $\mathbb{P}^\theta$. We adopt the same perspective in the present work. For instance, a recent line of works has defined $\mathbb{P}$ as the time-reversal of a noising diffusion process, such as PIS [Zhang and Chen, 2022], DDS [Vargas et al., 2023a] and *Time-reversed Diffusion Sampler* (DIS) [Berner et al., 2023]. In these methods, the neural network used in $\mathbb{P}^\theta$ is crucially required to be *target-informed*, *i.e.*, it is parameterized with the score of the target distribution. For a large number of variational training steps, this may result in a costly procedure, since each training step requires a full simulation of $\mathbb{P}^\theta$. While those algorithms were originally implemented with the largely used reverse KL divergence, Richter et al. [2023] demonstrate the benefits of using the Log-Variance (LV) divergence to avoid mode collapse in the variational optimization stage. However, even combined with a LV-based loss, these variational methods still require a target-informed parameterization. On the other hand, Vargas et al. [2024] propose a different perspective by defining $\mathbb{P}$ as a controlled version of an *annealed Langevin* diffusion, that is expected to follow a prescribed path of tractable marginal densities. Here, both $\mathbb{P}$ and $\mathbb{P}^\theta$ are induced by neural SDEs. The authors present two versions of their algorithm, *Controlled Monte Carlo Diffusion* (CMCD), using either reverse KL or LV divergence[2]. For clear comparison with the RDS discrete-time setting presented in Section 5.2.2, we describe all of these approaches (PIS, DDS, DIS, CMCD) under the discrete time scope in Appendix D.4.2.

Adaptive diffusion-based approaches. To alleviate the computational difficulties of divergence-based losses, recent works have proposed to learn $\mathbb{P}^\theta$ with an adaptive procedure [Phillips et al., 2024, Akhound-Sadegh et al., 2024]. These methods iterate two steps which consist of (a) obtaining approximate target samples by sampling from $\mathbb{P}^\theta$ and (b) optimizing θ using those samples via learning techniques usually restricted to generative modeling. In particular, Akhound-Sadegh et al. [2024] present *Iterated Denoising Energy Matching* (iDEM), where $\mathbb{P}$ corresponds to the time-reversal of a Variance-Exploding noising diffusion process [Song et al., 2021b]. In their setting, stage (a) is conducted by running the SDE induced by $\mathbb{P}^\theta$ and stage (b) relies on a novel energy-matching loss which directly depends on π. On the other hand, the *Particle Denoising Diffusion Sampler* (PDDS) [Phillips et al., 2024] (a) introduces a SMC-based scheme when sampling from $\mathbb{P}^\theta$, see Appendix D.5 for more details, and (b) implements an extension of the Target Score Matching loss [Bortoli et al., 2024a]. When it comes to practice, we find that both of these algorithms face significant limitations: while PDDS performance is highly sensitive to the choice of the initial $\mathbb{P}^\theta$, the iDEM energy-matching loss suffers from very high variance.

5.5 Numerical evaluation of RDS

To validate our approach, we compare GMM-LRDS and EBM-LRDS on a variety of multi-modal distributions against the following annealed methods: (a) **annealed MCMC methods** – *Sequential Monte Carlo* (SMC) [Del Moral et al., 2006] and *Replica Exchange* (RE) [Geyer et al., 1991]; (b) **variational diffusion-based methods**, implemented with the LV loss – LV-PIS [Zhang and Chen, 2022], LV-DDS [Vargas et al., 2023a], LV-DIS [Berner et al., 2023] and LV-CMCD [Vargas et al., 2024]; (c) **adaptive diffusion-based approaches** – iDEM [Akhound-Sadegh et al., 2024] and PDDS [Phillips et al., 2024].

To assess the performance of each sampler in multi-modal settings, we will evaluate how well the obtained samples are able to recover the weights of the target modes[3]. The details of each target distribution can be found in Appendix D.8.1.

[2] We note that Domingo-Enrich et al. [2026] recently introduced a version of CMCD specifically designed for sampling problems involving exponentially tilted dynamics, in which the target density is formulated as a tilting of a base distribution. However, we exclude this approach from our comparison because it does not align with the general sampling framework considered here.

[3] In Appendix D.7, we give expressions of additional variational metrics proposed by Blessing et al. [2024] to quantify mode collapse in the specific case of RDS.

Table 5.2: **Absolute mode weight estimation error** for a bi-modal Gaussian mixture with growing d, averaged over 16 sampling runs. Bold font indicates best result, orange cells refer to settings with uninformative mode weight estimation (*i.e.*, uniform mixture), red cells denote mode collapse. N/A denotes settings with numerical issues.

Algorithm	$d = 16 \downarrow$	$d = 32 \downarrow$	$d = 64 \downarrow$
SMC	11.4%±9.1%	15.8%±8.5%	15.2%±7.5%
RE	16.5%±1.3%	15.9%±1.4%	17.0%±1.4%
LV-PIS	6.0%±3.4%	33.2%±0.1%	33.0%±0.1%
LV-DDS	11.8%±9.3%	31.5%±2.9%	33.1%±0.1%
LV-DIS	16.4%±0.5%	16.5%±0.4%	16.8%±0.6%
LV-CMCD	32.2%±15.4%	50.1%±8.8%	16.3%±10.6%
iDEM	33.3%±0.0%	66.7%±0.0%	11.7%±0.4%
PDDS	**0.8**%±**0.6**%	66.7%±0.0%	N/A
GMM-LRDS	1.7%±0.6%	**2.7**%±**0.8**%	**4.1**%±**0.6**%

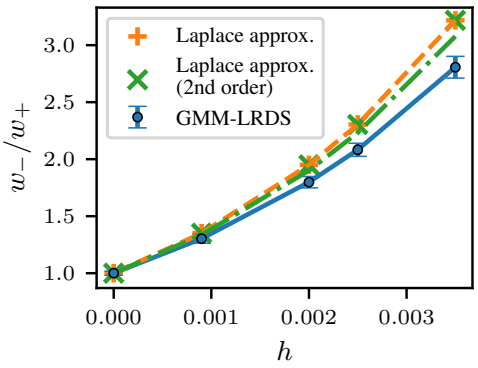


Figure 5.4: **Estimation of the relative weight** of ϕ^4 modes with increasing h, averaged over 16 sampling runs.

General experimental setting. To ensure fair comparison with previous approaches, we make sure that all competing methods are as informed as LRDS of prior knowledge on the target distribution. More specifically: we set π^{base} as a Gaussian approximation of $\hat{\pi}^{\text{ref}}$ in SMC, RE or CMCD; we choose σ based on a Gaussian isotropic approximation of $\hat{\pi}^{\text{ref}}$ for PIS, DDS or DIS (see Section 5.3.1); we standardize the target distributions using the empirical mean and variances of $\hat{\pi}^{\text{ref}}$ for iDEM and PDDS. Additionally, we pre-fill iDEM's training buffer with samples from $\hat{\pi}^{\text{ref}}$. Note that all competing methods use the score of the target distribution, either in their training procedure when computing the loss (iDEM, PDDS) or in the sampling procedure : through a target-informed neural network parameterization in PIS, DDS and DIS; through the target-informed base drift in CMCD; through the MCMC gradient steps in SMC or RE. In contrast, LRDS only requires evaluations of the target density, which makes it an interesting alternative in settings where the score of π is expensive to compute. We refer to Appendix D.8.3 for complete details of the implementation of these methods.

High-dimensional Gaussian mixtures. We first consider a synthetic but challenging setting, where π is a bi-modal Gaussian mixture whose modes are $\mathrm{N}(-\mathbf{1}_d, \Sigma_1)$, with weight $w_1 = 2/3$, and $\mathrm{N}(\mathbf{1}_d, \Sigma_2)$, with weight $w_2 = 1/3$, where $\mathbf{1}_d$ is the d-dimensional vector with all components equal to 1, and for $i \in \{1, 2\}$, $\Sigma_i \in \mathbb{R}^{d\times d}$ is a diagonal positive matrix with conditioning number equal to 100. For each method, the ground truth weight w_1 is estimated by a Monte Carlo estimator $\hat{w}_1$. We report the estimation error $|w_1 - \hat{w}_1|$ for increasing values of d in Table 5.2. Note that mode collapse occurs when $\hat{w}_1 \in \{0, 1\}$. As in Chapter 3, we emphasize that this benchmark is a deliberately simplified instance of multimodality far less intricate than real-world distributions that may feature exponentially many modes or severely ill-conditioned energy landscapes. Nonetheless, it serves as a minimal yet essential test: methods that fail on such controlled examples are unlikely to perform well on more realistic, high-complexity targets. Encouragingly, GMM-LRDS consistently outperforms all competing approaches across dimensions, thus passing this critical sanity check that most alternatives do not, even though we do not claim it would directly address the full complexity of real-world multimodal problems. In Appendix D.9.5, we complete these mode weight results with probability metrics (see Appendix D.2), present further experiments with bi-modal Gaussian mixtures with lower condition numbers and dimension, which show that competing algorithms are able to perform on par with LRDS in these simpler settings, and conduct ablation studies on this two-mode type of target (by lightly perturbing the reference distribution or increasing the distance between the modes).To further assess the performance and robustness of LRDS, we also consider the extension to more than two modes in Appendix D.9.5, where LRDS still outperforms its competitors.

Field system ϕ^4 from statistical mechanics. Next, we sample from the 1D ϕ^4 model, previously studied [Gabrié et al., 2022, Grenioux et al., 2024]. At the chosen temperature, the distribution has two well distinct modes with respective weights w_- and w_+ such that the relative weight w_-/w_+ can be adjusted through a 'local-field' parameter h. We discretize this continuous model with a grid size of 32 (*i.e.*, $d = 32$). For each method, we compute a Monte Carlo estimation of w_-/w_+ and compare the results with a Laplace approximation (0-th and 2-nd orders), see Appendix D.8.1 for the computations. In this setting, all competing approaches suffer from mode collapse while GMM-LRDS is close to the ground truth as shown by Figure 5.4.

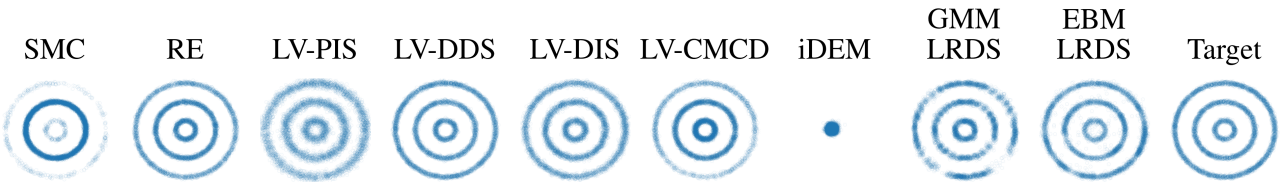


Figure 5.5: **Samples obtained for Rings distribution**. Reasonable results could not be obtained with PDDS due to numerical issues.

Complete results showing the failure of the competing methods are provided in Appendix D.9.5. Given the satisfying results obtained with the lightweight GMM-LRDS sampler, we did not run EBM-LRDS for this target distribution.

Compactly supported multi-modal distributions. Then, we aim to sample from 2-dimensional multi-modal distributions with complex geometries. We consider (a) *Rings distribution*, which has 3 ring-shaped modes, and (b) *Checkerboard distribution*, which has 8 square-shaped modes. In both cases, the modes are not evenly weighted. In this setting, we consider $J = 64$ components for GMM-LRDS and leverage the Replica Exchange algorithm as backbone annealed MCMC sampler in the EBM-LRDS training algorithm. Apart from adaptive methods, the structure of Rings modes are well recovered by all approaches, see Figure 5.5. However, we observe that non diffusion-based approaches fail to recover the ground truth mode weights. On the other hand, LRDS is the only sampling method to obtain samples that are close to exact for the Checkerboard distribution, see Figure 5.6, while being able to correctly estimate the ground truth mode weights.

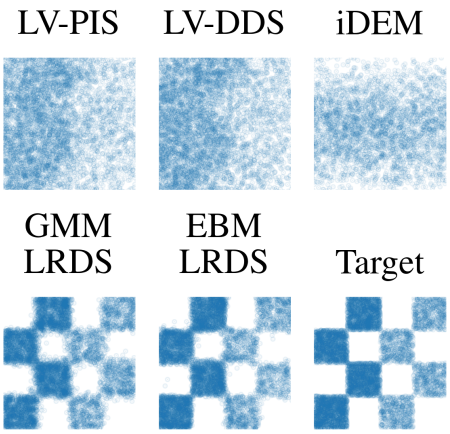


Figure 5.6: **Samples obtained for Checkerboard distribution.** All sampling methods except LRDS fail to provide accurate samples.

Bayesian logistic regression models. Lastly, we provide in Appendix D.9.5 sampling results on standard Bayesian posterior distributions, which are not however explicitly multi-modal.

5.6 Discussion

Building on the class of recently developed annealed VI methods, this paper presents the *Learned Reference-based Diffusion Sampler* (LRDS), which specifically addresses the challenging case of multi-modal target distributions. In essence, LRDS aims at learning a reference process – based on GMMs or multi-level EBMs – adapted to the target distribution by using samples obtained via local MCMC samplers initialized in the modes. Our numerical experiments show that GMM-LRDS accurately recovers the global information of the relative weights of the modes in several tens of dimensions, unlike competing methods, and that EBM-LRDS can help tackling distributions with non-Gaussian properties such as sharp supports. While the multi-modal distributions studied here do not capture the full spectrum of challenges encountered in real-world multimodal sampling, their simplicity makes them a useful first step. Distributions of this type can be seen as fundamental benchmarks that any sampler should be able to handle before being applied to more complex targets. In this context, LRDS demonstrates practical relevance relative to competing approaches. At the same time, it is important to note that SMC and RE algorithms could be enhanced with adaptive temperature schedules, which have been shown to substantially improve performance in challenging multimodal landscapes [Zhou et al., 2016, Syed et al., 2021, 2022, 2024], potentially allowing them to achieve comparable or superior results relative to LRDS. Evaluating LRDS on more complex, high-dimensional multimodal tasks and integrating these advanced adaptive variants of SMC and RE remain important directions for future work. However, this advantage of LRDS comes at the computational cost of the necessary pre-training of the reference process model. Concerning future work, we note in particular that EBM-LRDS is a promising tool for real-world sampling tasks on non-euclidean spaces which may benefit from the flexible definition of the reference process as an EBM. An interesting test case of this kind that shall be considered is the sampling of the Boltzmann distribution of proteins in internal coordinates.

Acknowledgements

We would like to thank Lorenz Richter, Julius Berner, Denis Blessing, Junhua Chen, Francisco Vargas and Yazid Janati for many stimulating discussions. LG and MG acknowledge funding from Hi! Paris. AD and MN would like to thank the Isaac Newton Institute for Mathematical Sciences and the Alan Turing Institute for support and hospitality during the programme *Diffusions in machine learning: Foundations, generative models and non-convex optimisation* when work on this paper was undertaken. Part of the work of AD is funded by the European Union (ERC, Ocean, 101071601). Views and opinions expressed here are however those of the authors only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them. This work was partially performed using HPC resources from GENCI–IDRIS (AD011015234).

Chapter 6

Diffusion-based Annealed Boltzmann Generators : benefits, pitfalls and hopes

6.1 Introduction

Sampling configurations from the Boltzmann distribution of a system $\pi(x) \propto \exp(-\mathcal{E}(x))$, where $\mathcal{E}(x)$ denotes the potential energy of configuration x, is a foundational and long-standing challenge. Reliable access to samples from π underpins the estimation of many key observables which, in turn, govern macroscopic behavior. Hence, efficient Boltzmann sampling is central to a broad range of applications, from characterizing biomolecular function and accelerating drug discovery to materials design and the study of complex statistical-physics models [Liu and Liu, 2001, Krauth, 2006, Stoltz et al., 2010, Ohno et al., 2018, Frenkel and Smit, 2023].

The core difficulty of sampling stems from the geometry of realistic energy landscapes. In many practical settings, the energy $\mathcal{E}$ is high-dimensional, non-smooth, and highly rugged, with numerous metastable basins (referred to as "modes") separated by high barriers. This structure severely challenges classical simulation-based approaches such as Molecular Dynamics (MD) and Markov Chain Monte Carlo (MCMC), whose generated samples follow dynamics prone to trapping in local minima, thus requiring a computationally prohibitive number of successive steps to mix across modes. The resulting samples are strongly correlated, leading to large statistical inefficiencies.

Boltzmann Generators (BGs) [Noé et al., 2019] address this bottleneck by amortizing sampling cost through training a generative model p^θ to approximate π, followed by a correction step that turns proposals from p^θ into samples from the target π. Modern BGs predominantly rely on normalizing flows (NFs), either discrete (DNFs) [Rezende and Mohamed, 2015, Papamakarios et al., 2021] or the more expressive continuous variant (CNFs) [Chen et al., 2018a, Grathwohl et al., 2019], because they support efficient sampling and (in principle) tractable density evaluations. For NFs, the natural correction mechanism is to embed proposals into Monte Carlo (MC) schemes, most prominently Importance Sampling (IS) [Müller et al., 2019, Noé et al., 2019, Köhler et al., 2020, Klein et al., 2023, Klein and Noé, 2024] and MCMC [Albergo et al., 2019, Gabrié et al., 2022, Del Debbio et al., 2022, Brofos et al., 2022, Samsonov et al., 2022, Cabezas et al., 2024]. However, these strategies are highly sensitive to the overlap between p^θ and π [Agapiou et al., 2017, Grenioux et al., 2023b]: in high dimension or for highly multi-modal targets, even small modeling errors can yield extremely poor correction capabilities. Moreover, in the CNF setting, evaluating $p^\theta(x)$ accurately is itself expensive, as it requires solving a neural ODE.

Recently introduced Diffusion Models (DMs) [Sohl-Dickstein et al., 2015, Ho et al., 2020, Song et al., 2021b] are generative models that have achieved state-of-the-art performance across many data modalities [Kong et al., 2021, Ho et al., 2022, Karras et al., 2024, Abramson et al., 2024], and thus provide a natural alternative to NFs as the backbone of Boltzmann generators. We review DMs in detail in Section 6.2.1. Their core principle is to learn how to remove noise from corrupted samples; training across many noise levels yields a sequential generation procedure that maps pure noise to structured data. While DMs often produce higher-fidelity samples than NFs on complex distributions, their inference mechanism does not integrate directly into classical BG pipelines, most

notably because their likelihood is typically not available in a tractable form.

This work reviews and extends approaches that turn DMs into BGs by leveraging annealed Monte Carlo (aMC) methods, introduced in Section 6.2.2 as refinements of classical MC schemes. The key idea of aMC is to replace a hard sampling problem by a sequence of easier ones, relying on a user-defined path of intermediate densities that bridges a simple base distribution to the target π. While many such paths are possible, several recent works have shown that DMs suggest a particularly natural construction; we unify and review these strategies in Section 6.3. Our overarching objective is to address the following question:

How can Diffusion Models yield accurate and efficient Boltzmann Generators?

To explore this question, we examine two complementary experimental regimes:

(A) Idealized regime: we assume that the DM is perfectly learned, thus isolating the statistical inference errors induced by aMC from errors due to imperfect training;

(B) Realistic regime: the DM is trained from biased data, reflecting practical settings.

Our main contributions are the following:

• We present a unified review of existing approaches that integrate Diffusion Models into annealed Monte Carlo to build Boltzmann Generators. These methods exploit the sequence of marginal distributions induced by the DM's denoising process as intermediate densities in aMC. In idealized regime **(A)**, we show that such DM-informed constructions consistently outperform traditional aMC designs.

• We further analyze strategies that leverage the conditional structure of the denoising process, which is naturally available from DMs. In practice, this is achieved through Gaussian approximations of the conditional distributions between consecutive noise levels. We distinguish *first-order* approximations, which match only the conditional mean, from *second-order* approximations, which also incorporate covariance information. In idealized setting **(A)**, we find that first-order approximations offer no improvement over a naive, correlation-free baseline (i.e., using marginal densities alone), despite additional access to exact knowledge of conditional means, whereas second-order approximations yield substantial performance gains.

• We propose a complementary alternative to Gaussian approximations by introducing deterministic transport maps. Importantly, these maps integrate seamlessly into the aMC framework and require only access to the previously mentioned conditional mean. In idealized regime **(A)**, this deterministic approach achieves performance comparable to second-order stochastic methods, at the cost of a small computational overhead, but without requiring covariance estimates.

• In realistic regime **(B)**, where all DM's components are learned from data, we observe a significant performance degradation across all DM-based aMC-BG methods compared to idealized regime **(A)**. Our empirical results indicate that this gap could primarily be due to inaccuracies the approximation of intermediate DM's densities.

Although BGs are often benchmarked on molecular systems, we instead focus on controlled yet challenging Gaussian mixture distributions. These widely used targets enable systematic comparison under precisely controlled levels of difficulty [Grenioux et al., 2025, Noble et al., 2025], and crucially allow exact computation of the quantities required in idealized setting **(A)**.

Multi-modal target distributions under consideration

We consider: *(i)* the bimodal distribution of Grenioux et al. [2025], denoted *TwoModes*, which allows one to control both the system's dimensionality, denoted dim, and the separation between imbalanced modes through a parameter $a > 0$ (larger a implies a larger gap); and *(ii)* the multi-modal target of [Noble et al., 2025, Appendix H.1], denoted *ManyModes*, which features a variable number of modes with non-uniform weights. Formal definitions are recalled in Appendix E.4.1.

For each target family, we select three representative "edge-case" configurations that combine high dimensionality with strong multi-modality, and are therefore particularly challenging. For *TwoModes*, we consider: close modes in high dimension ($a = 1.0$, dim $= 128$), distant modes in low dimension ($a = 10.0$, dim $= 16$), and an intermediate case ($a = 5.0$, dim $= 64$). For *ManyModes*, we use 16, 32, and 64 modes with dimension fixed to 32. To improve numerical stability and avoid target-specific hyperparameter tuning, all targets are standardized to have zero mean and unit covariance. We report Sliced Wasserstein Distance [Bonneel et al., 2015] as the primary metric in the main paper; additional metrics and ablations are provided in Appendix E.4.3.

Notation. For any measurable space $(\mathrm{X}, \mathcal{X})$, we denote by $\mathscr{P}(\mathrm{X})$ the space of probability measures defined on $(\mathrm{X}, \mathcal{X})$. Unless specified, if X is a topological space, then $\mathcal{X}$ is defined as the Borel σ-field of X. For simplicity, we use the same notation to refer both to a probability distribution and its density wrt the Lebesgue measure when it is defined. In our paper, π^{base} denotes a simple distribution that is easy to sample from (for instance, Gaussian), and is referred to as the "base" distribution. We denote $\mathrm{N}(\mu, \Sigma)$ with $\mu \in \mathbb{R}^d$ and $\Sigma \in \mathrm{S}_d^{++}$ the multivariate Gaussian distribution with mean μ and covariance Σ. For any Markov kernel $Q : \mathcal{B}(\mathbb{R}^d) \times \mathbb{R}^d \to [0, 1]$, we denote its conditional density $q(y|x) = Q(\mathrm{d}y, x)/\mathrm{d}y$ for any $(x, y) \in \mathbb{R}^d \times \mathbb{R}^d$. Moreover, for any probability distribution $\mu \in \mathscr{P}(\mathbb{R}^d)$, we denote by $\mu Q \in \mathscr{P}(\mathbb{R}^d)$ the distribution obtained by applying the kernel Q to μ, defined by

$$(\mu Q)(\mathrm{d}y) = \int_{\mathbb{R}^d} Q(\mathrm{d}y, x) \mathrm{d}\mu(x) \ .$$

For ease of reading, we may use the same notation for $Q(y, x)$ and $q(y|x)$ throughout the paper. For any C^1-diffeomorphism $T : \mathbb{R}^d \to \mathbb{R}^d$, we denote by $J_{\mathrm{T}}(x)$ the Jacobian matrix of T evaluated at x, and by $\mathrm{T}_{\#}\mu$ the pushforward of the distribution μ by T. Hence, if $X \sim \mu$, then $\mathrm{T}(X) \sim \mathrm{T}_{\#}\mu$. By the change-of-variable formula, the density of $\mathrm{T}_{\#}\mu$ wrt the Lebesgue measure is given by

$$\mathrm{T}_{\#}\mu(x) = \mu(\mathrm{T}^{-1}(x)) \left|\det J_{\mathrm{T}^{-1}}(x)\right| \ . \tag{6.1}$$

6.2 Background

Before detailing existing DM-based aMC-BG methods (Section 6.3) and presenting our deterministic version (Section 6.4), we first review the key ingredients that underpin these approaches: diffusion models (Section 6.2.1) and annealed sampling techniques (Section 6.2.2). Throughout this section, for both generative and sampling frameworks, π and π^{base} will respectively refer to the target and the base distributions.

6.2.1 Diffusion models

Forward process. The stochastic "noising" process of DMs that gradually corrupts the data with increasing Gaussian noise is described by a linear SDE of the form

$$\mathrm{d}X_t = f(t)X_t\mathrm{d}t + g(t)\mathrm{d}W_t, \quad X_0 \sim \pi, \quad t \in [0, T] \ , \tag{6.2}$$

where $(W_t)_{t\geq 0}$ is a standard Brownian motion, and $f : [0, T] \to \mathbb{R}$ and $g : [0, T] \to (0, \infty)$ are given schedule functions. Marginally, this forward diffusion process can be explicitly defined by

$$X_t \overset{\mathrm{d}}{=} S(t)X_0 + S(t)\sigma(t)Z, \quad X_0 \sim \pi, \quad Z \sim \mathrm{N}(0, \mathrm{I}_d) \ , \tag{6.3}$$

where $S(t) = \exp(\int_0^t f(u)\mathrm{d}u)$ and $\sigma^2(t) = \int_0^t g^2(u)/S^2(u)\mathrm{d}u$. As a result, the marginal density of X_t, denoted by p_t, is a convolution of π with a Gaussian kernel that writes as

$$p_t(x) = \int_{\mathbb{R}^d} \mathrm{N}(x; S(t)x_0, S(t)^2\sigma^2(t)\mathrm{I}_d)\mathrm{d}\pi(x_0) \ . \tag{6.4}$$

With an appropriate choice of schedules f and g (or equivalently, S and σ), the forward process interpolates between $p_0 = \pi$ and $p_T = \pi^{\text{base}}$, where π^{base} is a Gaussian distribution independent of π. We refer to Song et al. [2021b] and Karras et al. [2022] for a detailed presentation of commonly chosen noising schemes. In Appendix E.2, we detail computations related to the widely used Variance Preserving (VP) and Variance Exploding (VE) settings. In practice, the integral in (6.4) generally cannot be computed in closed form, rendering the marginal density p_t intractable for an arbitrary target distribution π.

Backward process. To generate new data, the idea is to reverse time in SDE (6.2) so as to denoise samples from π^{base} into samples from π. Under mild regularity conditions on f, g, and π, it can be shown [Anderson, 1982] that the reverse-time dynamics of the noising SDE is itself governed by another SDE, commonly referred to as the reverse-time or denoising SDE

$$\mathrm{d}X_t = \left[f(t)X_t - g^2(t)\nabla \log p_t(X_t)\right] \mathrm{d}t + g(t)\mathrm{d}\tilde{B}_t, \quad X_T \sim \pi^{\text{base}} \ , \tag{6.5}$$

where $(\tilde{B}_t)_{t\geq 0}$ is a reverse-time standard Brownian motion. Interestingly, the stochastic process induced by the denoising SDE has the same marginal distributions $(p_t)_{t\in[0,T]}$ as the stochastic process induced by its deterministic counterpart, called the probability flow ODE (PF-ODE) [Song et al., 2021b]

$$\mathrm{d}X_t = \left[f(t)X_t - \frac{g^2(t)}{2}\nabla \log p_t(X_t) \right] \mathrm{d}t, \quad X_T \sim \pi^{\text{base}} \,. \tag{6.6}$$

Thus, to obtain samples from π at inference, one needs to either solve the SDE (6.5) or the ODE (6.6) backward in time (*i.e.*, from $t = T$ to $t = 0$), starting from noise samples drawn from π^{base}. Below, we detail the denoising transition kernels and transport maps associated with approximate numerical solvers for, respectively, the SDE (6.5) and the ODE (6.6).

Stochastic transition kernels. For $0 \leq s < t \leq T$, the conditional distribution of X_t given $X_s = x_s$ is a tractable Gaussian distribution $q_{t|s}(\cdot|x_s)$, called *noising* transition kernel, that writes as

$$q_{t|s}(\cdot \mid x_s) = \mathrm{N}\left(\alpha_{t|s}\, x_s, \sigma^2_{t|s}\, \mathrm{I}_d\right) \,, \tag{6.7}$$

where $\alpha_{t|s} = S(t)/S(s)$ and $\sigma^2_{t|s} = S^2(t)[\sigma^2(t) - \sigma^2(s)]$. In contrast, the conditional distribution of X_s given $X_t = x_t$ induced by the denoising SDE (6.5), denoted $q_{s|t}(\cdot|x_t)$ and called *denoising* transition kernel, does not have a closed-form expression in general. and is usually approximated by a Gaussian distribution.

A natural way to approximate the denoising transition kernel $q_{s|t}$ is to numerically solve the denoising SDE (6.5) using integration schemes such as Euler-Maruyama (EM). This yields the approximate kernel

$$q^{\text{EM}}_{s|t}(\cdot|x_t) = \mathrm{N}\left(x_t + (t-s)\{-f(t)x_t + g^2(t)\nabla \log p_t(x_t)\}, g^2(t)(t-s)\mathrm{I}_d\right) \,, \tag{6.8}$$

which requires access only to the score function $\nabla \log p_t$. More advanced schemes such as Exponential Integration (EI) can be employed for improved accuracy which is the approach adopted in this work (see Appendix E.2). These methods are coined "first-order" because they rely solely on the score.

In contrast to first-order methods, "second-order" approximations of the denoising kernel $q_{s|t}(\cdot|x_t)$ require access to the Hessian $\nabla^2 \log p_t(x_t)$ in addition to the score $\nabla \log p_t(x_t)$. A natural approach is to approximate $q_{s|t}$ with a Gaussian distribution whose mean and covariance are given by a second-order expansion of Tweedie's formula [Grenioux et al., 2024, Appendix A, Lemma 4]

$$\begin{aligned} m_{s|t}(x_t) &= \mathbb{E}[X_s|X_t = x_t] = \frac{x_t + \sigma^2_{t|s}\nabla \log p_t(x_t)}{\alpha_{t|s}} \,, \\ \Sigma_{s|t}(x_t) &= \mathrm{Cov}[X_s|X_t = x_t] = \sigma^2_{t|s}\frac{\mathrm{I}_d + \sigma^2_{t|s}\nabla^2 \log p_t(x_t)}{\alpha^2_{t|s}} \,. \end{aligned} \tag{6.9}$$

This leads to the second-order Gaussian approximation

$$q^{\text{DDPM}}_{s|t}(\cdot|x_t) = \mathrm{N}(m_{s|t}(x_t), \Sigma_{s|t}(x_t)) \,. \tag{6.10}$$

We refer to this second-order Gaussian approximation as the Skip-Step Denoising Diffusion Probabilistic Models (DDPM) kernel, following the terminology introduced by Ou et al. [2025].

Deterministic transition maps. In the case of the PF-ODE (6.6), $q_{s|t}$ degenerates to a Dirac mass, *i.e.*, $X_s = \mathrm{T}_{s|t}(x_t)$ where $\mathrm{T}_{s|t} : \mathbb{R}^d \to \mathbb{R}^d$ is the deterministic map that solves the ODE (6.6) backward in time on $[s, t]$. In practice, $\mathrm{T}_{s|t}$ is intractable too, but may be approximated via first-order integration methods. For instance, using the Euler scheme leads to

$$\mathrm{T}^{\text{EM}}_{s|t}(x_t) = x_t - f(t)(t-s)x_t + \frac{g^2(t)}{2}(t-s)\nabla \log p_t(x_t) \,. \tag{6.11}$$

Similarly to the stochastic setting, EI versions of such transition maps can be derived to reduce discretization error, see Appendix E.2 for more details.

Training DMs. In practice, the score functions $\{\nabla \log p_t\}_{t\in[0,T]}$ and, for second-order methods, the corresponding Hessians $\{\nabla^2 \log p_t\}_{t\in[0,T]}$, are not available in closed form for general target distributions and must therefore be estimated. As a result, data generation relies on approximate dynamics: first, the SDE (6.5) or ODE (6.6) is approximated through estimated scores (yielding an *estimation error*); second, these approximate dynamics are numerically solved using the tools described above (yielding a *discretization error*).

Score functions are typically learned from data using score-matching approaches [Hyvärinen, 2005, Vincent, 2011, Song et al., 2021b, Bortoli et al., 2024a]. While Hessians can in principle be obtained by differentiating the learned score network, this is computationally prohibitive in practice. Early methods therefore relied on scalar, state-independent approximations [Ho et al., 2020], whereas more recent works have proposed more accurate diagonal or full-matrix approximations—possibly state-dependent—learned via dedicated objectives using a pre-trained score model [Nichol and Dhariwal, 2021, Bao et al., 2022b,a]. A recent overview is provided in Ou et al. [2025].

Another line of research aims at rather approximating the log-densities $\{\log p_t\}_{t\in[0,T]}$ with neural networks, and then taking the derivative with respect to the input to obtain score or Hessian approximations. Various related objectives have been recently designed, either based on maximum likelihood [Gao et al., 2021, Zhang et al., 2023, Zhu et al., 2024, Noble et al., 2025], consistency via Fokker-Planck equation [Shi et al., 2024, Plainer et al., 2025], consistency via Bayes's rule [He et al., 2026] or multi-label classification [Yadin et al., 2024]. In practice, the dominant strategy remains the score matching approach, which indirectly approximates the DM log-densities by training a neural network to match their gradient [Song and Kingma, 2021, Salimans and Ho, 2021, Du et al., 2023, Phillips et al., 2024, Thornton et al., 2025] or their time derivative [Guth et al., 2025, Yu et al., 2025] : for the latter, we will refer to it as "time" score matching.

Diffusion model under consideration

In all experiments presented below, the noising diffusion process is chosen to be the linear Variance Preserving (VP) diffusion path [Song et al., 2021b] with hyperparameters $(\beta_{\min}, \beta_{\max}, T) = (0.1, 20, 1)$, whose exact noising kernel (6.7) is computed in Lemma 100 (Appendix E.2.2). When using stochastic denoising kernels, we consider the EI scheme given in Lemma 101 (Appendix E.2.2) for first-order approaches and the DDPM scheme given in (6.9) for second-order approaches; when using noising and denoising transport maps, we consider the EI-based ODE integration schemes detailed in Lemmas 102 and 103 (Appendix E.2.2). Moreover, we set the time discretization $\{t_k\}_{k=0}^K \subset [0,T]$ so as to be constant in log-SNR increments (see Appendix E.1.2 for more details). We will refer to the induced sequence of densities $\{p_{t_k}\}_{k=0}^{K-1}$ (also denoted $\{p_k\}_{k=0}^{K-1}$), marginally defined by (6.4), as the "diffusion" path.

6.2.2 Standard Monte Carlo & Annealed sampling

This section presents the Monte Carlo tools that are central to all BG methods presented below. We recall that the original purpose of these methods is to generate samples from π, with only access to its energy function $\mathcal{E}$ up to an additive constant. We begin by reviewing classic techniques, which serve as foundation for the aMC methods introduced afterwards.

Importance Sampling. Importance Sampling (IS) is a fundamental Monte Carlo method that approximates expectations taken under π using samples drawn from a proposal distribution ρ whose density is tractable. Assuming that $\mathrm{Supp}(\rho) \subset \mathrm{Supp}(\pi)$, any π-integrable function ϕ satisfies

$$\mathbb{E}_\pi[\phi(X)] = \mathbb{E}_\rho\left[w(X)\phi(X)\right], \quad \text{where } w(x) = \frac{\pi(x)}{\rho(x)} \text{ is the importance weight.}$$

In practice, this means that sampling from π via IS reduces to (i) sample N particles $\{x^i\}_{i=1}^N$ from ρ and (ii) reweight them using the importance weights $\{w(x^i)\}_{i=1}^N$ [1]. Although IS is simple to implement, its accuracy critically depends on how well ρ matches π. In particular, the variance of the importance weights can grow rapidly, potentially exponentially with the dimension, when the mismatch is large [Agapiou et al., 2017].

[1] When the density π is only known up to a normalizing constant, as it is often the case in practice, one turns to the *self-normalized* weights $\bar{w}(x^i) = w(x^i)/\sum_{j=1}^N w(x^j)$, which however leads to a biased estimator.

Markov Chain Monte Carlo. Markov Chain Monte Carlo (MCMC) methods are designed to simulate a Markov chain whose stationary distribution is π, hence generating asymptotically accurate samples.

MCMC methods typically construct their transition mechanism using a proposal distribution $q(y|x)$, which suggests a new state y from the current state x. The Metropolis-Hastings (MH) algorithm then corrects this proposal via an acceptance-rejection step to ensure that the chain targets the desired distribution π. Specifically, given x, the proposed $y \sim q(\cdot|x)$ is accepted with probability

$$\alpha(x, y) = \min\left(1, \frac{q(x|y)\pi(y)}{q(y|x)\pi(x)}\right) = \min\left(1, \frac{q(x|y)\exp(-\mathcal{E}(y))}{q(y|x)\exp(-\mathcal{E}(x))}\right) , \tag{6.12}$$

otherwise the new state is set as x. Note that the MH algorithm can be extended to the deterministic case, when $q(\cdot|x) = \delta_{\mathrm{T}(x)}$ for a diffeomorphism $\mathrm{T} : \mathbb{R}^d \to \mathbb{R}^d$ that is required to be involutive, *i.e.*, $\mathrm{T} \circ \mathrm{T} = \mathrm{I}_d$. In this case, the acceptance probability only depends on the previous state x and writes

$$\alpha(x) = \min\left(1, \frac{\mathrm{T}_{\#}\pi(x)}{\pi(x)}\right) = \min\left(1, \frac{\exp(-\mathcal{E}(\mathrm{T}(x))\,|\det J_T(x)|}{\exp(-\mathcal{E}(x))}\right) . \tag{6.13}$$

This deterministic formulation encompasses the popular Hamiltonian Monte Carlo (HMC) algorithm [Neal et al., 2011]. As with IS, the performance of such MH-based samplers hinges on the quality of the proposal. For instance, independent proposals scale poorly with dimension [Grenioux et al., 2023b], and multi-modal targets pose additional challenges, as proposals must efficiently explore both within and across the modes. Modern MH variants [Metropolis et al., 1953, Duane et al., 1987], including the Metropolis-Adjusted Langevin Algorithm (MALA) [Roberts and Tweedie, 1996], leverage gradient information to improve local mixing but still struggle with global exploration.

While IS and MCMC are fundamental sampling tools, they often fail in high-dimensional or multi-modal settings. *Annealed* sampling specifically addresses this limitation by breaking the original sampling problem into K sampling problems with gradual complexity, by introducing a sequence of distributions $\{p_k\}_{k=0}^K$ that smoothly bridge between a simple base distribution $p_K = \pi^{\text{base}}$ and the target $p_0 = \pi$. We consider such sequence in the rest of this section. By leveraging correlations across this sequence, it is possible to gradually transform samples from π^{base} into samples from π while avoiding the pitfalls of standard MC methods.

Annealed Importance Sampling. Annealed Importance Sampling (AIS) [Neal, 2001] extends classic IS by defining a joint target distribution $\pi_{0:K}$ over a sequence of variables $(x_0, \dots, x_K)$ such that its 0-th marginal is the target distribution π. Similarly, a joint proposal distribution $\rho_{0:K}$ is built such that its K-th marginal is the base distribution π_{prior}. Both of these joint distributions are designed recursively as follows

$$\pi_{0:K}(x_{0:K}) = \pi(x_0)\prod_{k=0}^{K-1} q_{k+1|k}(x_{k+1}|x_k), \quad \rho_{0:K}(x_{0:K}) = \pi^{\text{base}}(x_K)\prod_{k=0}^{K-1} q_{k|k+1}(x_k|x_{k+1}) , \tag{6.14}$$

where $q_{k+1|k}$ and $q_{k|k+1}$ respectively denote *forward* and *backward* Markov transition kernels. In this case, the importance weights are defined by

$$w^{\text{AIS}}(x_{0:K}) = \frac{\pi_{0:K}(x_{0:K})}{\rho_{0:K}(x_{0:K})} \tag{6.15}$$

Analogously to IS, sampling from π reduces to (i) sample N trajectories of particles $\{x^i_{0:K}\}_{i=1}^N$ from $\rho_{0:K}$ and (ii) reweight the particles $\{x^i_0\}_{i=1}^N$ with the importance weights $w^{\text{AIS}}(x_{0:K})$ [2]. However, while easier to achieve than classic IS, the efficiency of AIS also depends on how closely $\rho_{0:K}$ matches $\pi_{0:K}$. In particular, if there exists a sequence of bridging distributions $\{p_k\}_{k=0}^K$ (*i.e.*, such that $p_0 = \pi$ and $p_K = \pi^{\text{base}}$) for which the forward and backward kernels satisfy the Bayes rule defined as

$$p_k(x_k)q_{k+1|k}(x_{k+1}|x_k) = p_{k+1}(x_{k+1})q_{k|k+1}(x_k|x_{k+1}), \quad \forall k \in \{0, \dots, K-1\} , \tag{6.16}$$

then it holds exactly that $\pi_{0:K} = \rho_{0:K}$, *i.e.*, the estimator has minimal variance.

In standard AIS [Neal, 2001], the forward and backward kernels are typically chosen to be identical reversible MCMC kernels with respect to a given density path $\{p_k\}_{k=0}^K$ interpolating π to π^{base}, which simplifies the importance weights given in (6.15) but violates the Bayes consistency condition (6.16).

[2] In practice, these weights are also self-normalized as in classic IS.

Sequential Monte Carlo. Sequential Monte Carlo (SMC) methods [Doucet et al., 2001, Del Moral et al., 2006] address a major limitation of AIS, namely weight degeneracy, where importance weights progressively concentrate on a few particles—an effect that is particularly severe in high-dimensional settings. While SMC relies on the same forward and backward kernels as AIS, it introduces intermediate resampling steps that effectively decompose a single long AIS trajectory from p_K to p_0 into two consecutive AIS procedures. Concretely, an initial AIS run propagates particles from p_K to an intermediate distribution p_k for some $k \in \{1, \dots, K-1\}$; particles are then resampled according to their importance weights to obtain a population representative of p_k. A second AIS run, initialized from these resampled particles, subsequently propagates the system from p_k to p_0. This mid-trajectory realignment prevents particle collapse, maintains diversity, and significantly reduces weight degeneracy. The construction naturally extends to multiple resampling points by partitioning the path between π^{base} and π into shorter AIS segments, which substantially reduces the variance of the AIS estimator without increasing the cost of importance-weight evaluations. In practice, SMC methods are often further augmented with MCMC rejuvenation steps at each stage to better align particles with the intermediate distributions, at the expense of additional computational cost.

Replica Exchange. Replica Exchange (RE) [Swendsen and Wang, 1986, Geyer et al., 1991, Hukushima and Nemoto, 1996] is an annealed sampling method that predates AIS and SMC. Unlike these sequential methods, RE correlates the distributions $\{p_k\}_{k=0}^K$ in parallel, rather than through a recursion. The goal is to construct a MCMC algorithm targeting the extended distribution $\bar{\pi}_{0:K}(x_{0:K}) = p_0(x_0)p_1(x_1)\dots p_K(x_K)$. Its transition kernel is composed of two parts: (i) an exploration kernel that independently applies standard MCMC updates to each p_k in parallel, and (ii) a communication kernel that correlates the different marginals. A basic communication move consists of a deterministic "swap" between two consecutive levels k and $k+1$, mapping $(x_0, \dots, x_k, x_{k+1}, \dots, x_K)$ to $(x_0, \dots, x_{k+1}, x_k, \dots, x_K)$. Since this mapping is involutive, it can be used within the Metropolis–Hastings correction to ensure that the joint distribution $\bar{\pi}_{0:K}$ is stationary. The corresponding acceptance rate obtained from (6.13) is given by

$$\alpha_k^{\text{RE}}(x_{0:K}) = \min\left(1, \frac{p_{k+1}(x_k)p_k(x_{k+1})}{p_k(x_k)p_{k+1}(x_{k+1})}\right). \tag{6.17}$$

By applying these MH-calibrated swaps in parallel between even or odd pairs of indices in $\{0, \dots, K\}$, one defines the even and odd communication kernels, respectively. These are commonly combined using a uniform mixture to build the full communication kernel. However, recent work suggests that deterministically alternating between even and odd kernels is more effective [Okabe et al., 2001, Syed et al., 2022]. We adopt this so-called *non-reversible* strategy in the rest of the paper.

Standard designs of interpolation density paths. A central component of all aMC methods is the design of the interpolation density path. This path is critical to ensure good performance: in AIS and SMC, it governs the overlap between consecutive distributions, which directly affects the variance of the estimators; in RE, the consecutive overlap controls the probability of accepting swap moves between adjacent levels. When only the unnormalized density of π is available, a common choice is the geometric interpolation path [Neal, 2001, Gelman and Meng, 1998], defined for all $x \in \mathbb{R}^d$ by

$$p_k(x) \propto \pi(x)^{\beta_k}\pi^{\text{base}}(x)^{1-\beta_k}, \tag{6.18}$$

where the annealing schedule $\{\beta_k\}_{k=0}^K$ is decreasing, and satisfies $(\beta_0, \beta_K) = (1, 0)$. We will refer to the collection of unnormalized densities obtained via (6.18) as the "tempering" path. The major benefit of the tempering paths is their computational efficiency, as they allow for simple evaluations of the scores $\{\nabla \log p_k\}_{k=0}^K$, which are frequently required in MCMC transition kernels, via a linear combination of $\nabla \log \pi$ and $\nabla \log \pi^{\text{base}}$.

However, these paths are usually pathological for multi-modal targets, as they suffer from mass teleportation (also referred to as mode switching), which reflects sudden shifts in probability mass between modes along the interpolation path [Máté and Fleuret, 2023]. In practice, such sudden shifts undermine the assumed proximity between bridging densities, leading to instability in aMC. Mitigating this issue usually requires either carefully tuning the annealing schedule $\{\beta_k\}_{k=0}^K$ for each target or using a large number of intermediate levels K, which can incur significant computational cost.

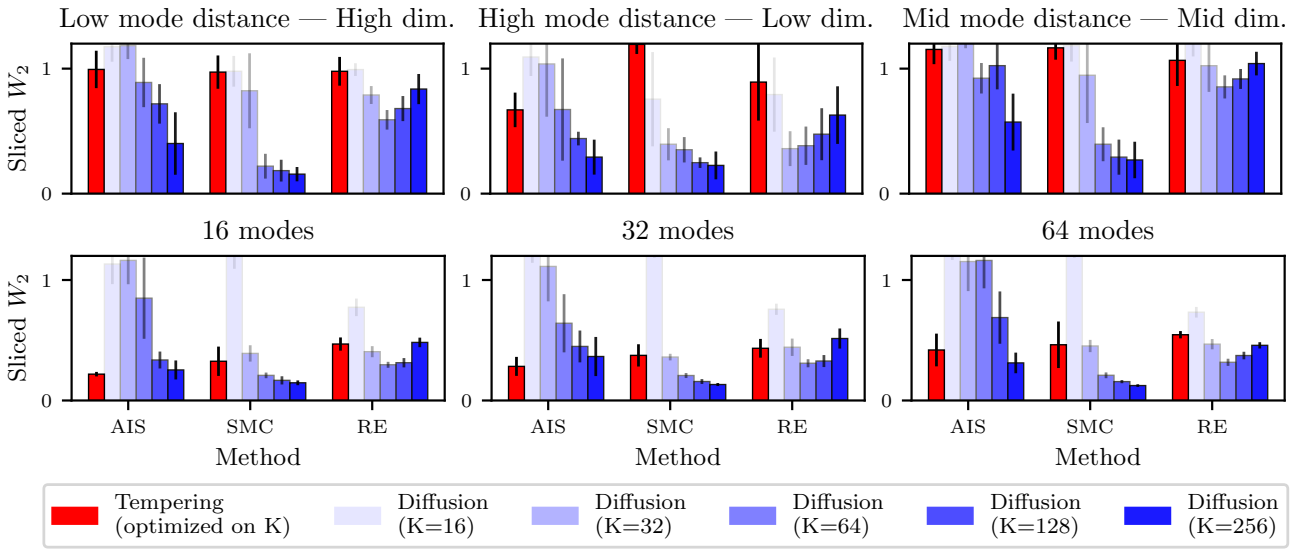


Figure 6.1: **Sampling results for classic annealed samplers with diffusion (blue) and tempering (red) density paths**, when targeting *TwoModes* **(Top)** and *ManyModes* **(Bottom)** distributions in idealized setting **(A)**. For tempering paths, we display the best-performing result among all values of K. For diffusion paths, we display the results for all values of K : the darker the bar, the higher K. In particular, these configurations do not share the same computational budget. Each result is averaged over 8 runs with 8,192 samples per run. We observe that for almost all target settings and samplers, there exists a number of annealing levels K for which the noising density path outperforms the best tempering baseline.

6.3 Diffusion-based aMC as a Boltzmann Generator backbone : benefits and pitfalls

Diffusion models are a natural fit for aMC schemes, as they inherently define a sequence of intermediate densities that can be leveraged in sampling algorithms such as AIS, SMC, or RE. In Section 6.3.1, we show that even a naive integration, simply using the sequence of DM densities as a direct replacement for the classic tempering sequence, can already deliver strong performance, thanks to the favorable properties of the Gaussian convolution paths induced by DMs. In Section 6.3.2, we review related methodologies, that additionally propose to "enhance" standard aMC tools using DM stochastic transition kernels. However, we demonstrate in Section 6.3.3 that those designs are fundamentally limited in challenging multi-modal scenarios.

We emphasize that, although the presented methods involve different hyperparameters, we focus our numerical evaluation solely on the effect of the number of annealing levels (defined as $K \in \{16, 32, 64, 128, 256\}$), common to all methods, because it directly controls the overlap between consecutive distributions along the annealing path, a factor highlighted as crucial to the performance of aMC. We leave investigation of automatic tuning methods [Syed et al., 2022, 2024] for future work.

6.3.1 Of the interest of diffusion-based density paths

As noted by Máté and Fleuret [2023], diffusion paths are typically well conditioned and avoid common pitfalls of tempering paths, such as abrupt mode switching. In particular, DM-induced paths preserve the relative mass of different modes throughout the annealing process, leading to more stable sampling dynamics. This explains why diffusion paths consistently outperform tempering paths in aMC, as illustrated in idealized setting **(A)** by Figure 6.1. Across all standard aMC methods considered (see Section 6.2.2), we observe that there exists a number of annealing levels K for which a perfectly learned diffusion path outperforms an optimally tuned tempering path. For AIS and SMC, performance generally improves with increasing K, with best results attained at the largest value tested ($K = 256$). For RE-based samplers, the dependence on K is less monotonic: while larger K improves local overlap and facilitates swaps, it can also hinder long-range communication between levels, leading to degraded performance beyond a certain point. Overall, these results highlight the strength of diffusion over tempering paths, motivating their use when a learned DM is available.

6.3.2 Review of existing diffusion-based aMC-BGs

Interestingly, DMs provide more than a sequence of intermediate densities: they also grant access to noising and denoising stochastic transition kernels (see, e.g., (6.7) and (6.8), (6.10)), which can be strategically exploited to improve both efficiency and robustness. In this section, we review existing extensions of aMC that leverage this additional structure. These approaches assume access to a DM defined on a discrete time grid $\{t_k\}_{k=0}^K \subset [0,T]$, enabling the additional evaluation of the associated noising kernels $\{q_{k+1|k}\}_{k=0}^{K-1}$ and denoising kernels $\{q_{k|k+1}\}_{k=0}^{K-1}$.

Diffusion-based AIS. DMs have been successfully integrated into AIS frameworks in recent work [Zhang et al., 2024b, 2025]. The core idea consists in using the exact noising transition kernels (6.7) as forward kernels, and *first-order* denoising transition kernels[3], similar to (6.8), as backward kernels, to respectively define the extended target and proposal distributions, see (6.14). By doing so, only the score functions are needed, not the log-densities. A key advantage of this approach is that, when the backward kernels match the exact denoising kernels, the forward and backward transitions satisfy the optimal Bayes condition (6.16), which ensures that the importance weights exhibit minimal variance.

Diffusion-based SMC. The exact same use of DM transition kernels has recently been extended to the SMC setting through the *Particle Denoising Diffusion Sampler* (PDDS) [Phillips et al., 2024]. In contrast to AIS, however, the SMC formulation additionally requires the intermediate log-densities, up to normalizing constants, in order to perform resampling.

Diffusion-based RE. In the spirit of PDDS, Zhang et al. [2026] lately explored the use of DM transition kernels within the RE framework to propose the *Diffusion-based Accelerated Parallel Tempering* (Diff-APT) sampler. In Diff-APT, the traditional RE swaps between adjacent levels are combined with stochastic refinements inherited from those kernels. Given current states x_k and x_{k+1} at levels k and $k+1$, Diff-APT first samples proposal states $y_{k+1} \sim q_{k+1|k}(\cdot|x_k)$ and $y_k \sim q_{k|k+1}(\cdot|x_{k+1})$, where $q_{k+1|k}$ and $q_{k|k+1}$ respectively denote the exact noising (forward) kernel, see (6.7), and a *first-order* denoising (backward) kernel, see (6.8) for instance, between times t_k and t_{k+1}. By exploiting the underlying correlation between noise levels, each chain is moved closer to its corresponding target distribution, respectively p_{k+1} and p_k. Then, this stochastic-based swap is calibrated using the MH correction, resulting in the following acceptance probability

$$\alpha_k^{\mathrm{RE}}(x_{0:K}, y_{0:K}) = \min\left(1, \frac{p_k(y_k)p_{k+1}(y_{k+1})q_{k+1|k}(x_{k+1}|y_k)q_{k|k+1}(x_k|y_{k+1})}{p_k(x_k)p_{k+1}(x_{k+1})q_{k+1|k}(y_{k+1}|x_k)q_{k|k+1}(y_k|x_{k+1})}\right), \tag{6.19}$$

defined for any $(x_{0:K}, y_{0,K}) \in \mathbb{R}^{(K+1)d} \times \mathbb{R}^{(K+1)d}$. Compared to the standard RE acceptance ratio (6.17), this novel expression features four additional terms, which correspond to symmetric evaluations of forward and backward kernels. As in AIS and SMC, if the forward and backward kernels satisfy the Bayes condition (6.16) the proposed swap is systematically accepted, *i.e.*, the acceptance probability (6.19) always equals one.

6.3.3 Current approaches fail due to first-order approximations of DM stochastic dynamics

Although theoretically well motivated, the existing DM-based aMC-BGs reviewed in Section 6.3.2 do not yield noticeable improvements over the standard baseline studied in Section 6.3.1, in idealized setting **(A)** where both log-densities and score functions are assumed to be perfectly known. In practice, this deficiency is expected to be further exacerbated by the additional errors introduced by learning approximations. In Figure 6.3, we evaluate the aforementioned methods in this perfect-learning regime on target distributions of increasing difficulty and report the resulting errors. The figure shows that this unfavorable behavior persists across all considered targets, corresponding to the different rows. We argue that this systematically poor performance stems from the use of *first-order* denoising kernels as backward kernels within aMC schemes.

Indeed, if we leverage access to Hessian functions [4] to build *second-order* denoising kernels such as (6.9), we consistently obtain significant gains over both the baseline and their first-order counterparts. This additional result highlights the value of higher-order information for guiding transitions along the diffusion density path.

[3] Although Zhang et al. [2025] propose to adjust the covariance of the denoising kernels via additional learning, we still consider this approach as 'first-order' as it does not rely on the Hessian functions $\{\nabla^2 \log p_k\}_{k=0}^K$.

[4] In our experiments, we only exploit the diagonal of the exact Hessians to ensure a good compromise between accuracy and computational efficiency in high dimension.

However, using such second-order kernels in practice requires additional covariance estimations [Ou et al., 2025], which is beyond the scope of most of DM training methods, where only approximations of log-densities and/or scores are available.

6.4 Exploiting deterministic transitions of DMs in aMC methods : a new hope ?

In this section, we propose investigating the design of a deterministic diffusion-based aMC-BG. We first describe its general principle in Section 6.4.1 and detail in Section 6.4.2 how to instantiate it concretely for DMs. We demonstrate that, in idealized setting **(A)**, our method outperforms previous first-order approaches, while being on par with the second-order stochastic ones. Similarly to Section 6.3.2, we assume throughout this section that we have access to scores and log-densities from a DM associated to a certain time discretization $\{t_k\}_{k=0}^K \subset [0, T]$.

6.4.1 General methodology

From stochastic to deterministic DM dynamics. To further exploit the potential of aMC sampling methods, we propose to use *deterministic* kernels, by replacing stochastic transition kernels with their deterministic counterparts, which approximate the PF-ODE (6.6) rather than the denoising SDE (6.5). Below, we explain how the aMC framework presented in Section 6.2.2 naturally extends to this setting.

Annealed samplers with deterministic transitions. In this paragraph, we consider K pairs of candidate transport maps, divided between *forward* maps $\{\mathrm{T}_{k+1|k}\}_{k=0}^{K-1}$ and *backward* maps $\{\mathrm{T}_{k|k+1}\}_{k=0}^{K-1}$. Moreover, we assume that **(a)** these maps are C^1-diffeomorphisms, and **(b)** verify the *per-level mutual invertibility* property, defined for any $k \in \{0, \dots, K-1\}$ by

$$\mathrm{T}_{k+1|k} \circ \mathrm{T}_{k|k+1} = \mathrm{T}_{k|k+1} \circ \mathrm{T}_{k+1|k} = \mathrm{Id} \,. \tag{6.20}$$

To exploit the use of these transport maps into aMC samplers, we simply propose to set the forward Markov kernels $\{q_{k+1|k}\}_{k=0}^{K-1}$ and backward Markov kernels $\{q_{k|k+1}\}_{k=0}^{K-1}$ (used as transition kernels between adjacent levels in aMC methods) as Dirac masses defined for any $k \in \{0, \dots, K-1\}$ by $q_{k+1|k} = \delta_{\mathrm{T}_{k+1|k}}$ and $q_{k|k+1} = \delta_{\mathrm{T}_{k|k+1}}$ respectively.

Adaptation to AIS/SMC instance. Under this setting, the AIS framework boils down to standard IS targeting π with the push-forward of π^{base} through all backward maps as proposal. Using the change-of-variables formula, the AIS weight (6.15) admits the following deterministic version, solely depending on the state x_0 :

$$w^{\mathrm{AIS}}(x_0) = \frac{\pi(x_0)}{(\mathrm{T}_{0:K})_{\#}\pi^{\mathrm{base}}(x_0)}, \qquad \text{with } \mathrm{T}_{0:K} = \mathrm{T}_{0|1} \circ \mathrm{T}_{1|2} \dots \circ \mathrm{T}_{K-1|K}. \tag{6.21}$$

Using the chain rule, the determinant of the Jacobian of the full map $\mathrm{T}_{0:K}$ appearing in $(\mathrm{T}_{0:K})_{\#}\pi^{\mathrm{base}}$ (see (6.1)) can be written as a product of the determinants of Jacobian of the individual maps $\mathrm{T}_{k|k+1}$ for $k \in \{0, K-1\}$.

Adaptation to RE instance. By substituting Markov kernels with Dirac masses, the resulting swap in RE sampling procedure defines a deterministic map on the full extended space

$$\bar{\mathrm{T}}_k(x_{0:K}) = (x_0, \dots, \mathrm{T}_{k|k+1}(x_{k+1}), \mathrm{T}_{k+1|k}(x_k), \dots, x_K) \,,$$

which is guaranteed to be involutive due to assumption **(b)**. In particular, this property ensures that $\bar{\mathrm{T}}_k$ can effectively be integrated within the Metropolis–Hastings algorithm with deterministic proposal, see (6.13).

Using the identity $(\bar{\mathrm{T}}_k)_{\#}\bar{\pi}(x_{0:K}) = p_0(x_0) \dots (\mathrm{T}_{k|k+1})_{\#}p_{k+1}(x_k)(\mathrm{T}_{k+1|k})_{\#}p_k(x_{k+1}) \dots p_K(x_K)$, we obtain the following acceptance probability:

$$\alpha_k^{\mathrm{RE}}(x_{0:K}) = \min\left(1, \frac{(\mathrm{T}_{k|k+1})_{\#}p_{k+1}(x_k)(\mathrm{T}_{k+1|k})_{\#}p_k(x_{k+1})}{p_k(x_k)p_{k+1}(x_{k+1})}\right) \,. \tag{6.22}$$

This swapping mechanism is illustrated in Figure 6.2. Note that setting both $\mathrm{T}_{k|k+1}$ and $\mathrm{T}_{k+1|k}$ as the identity map recovers the standard RE algorithm as a special case.

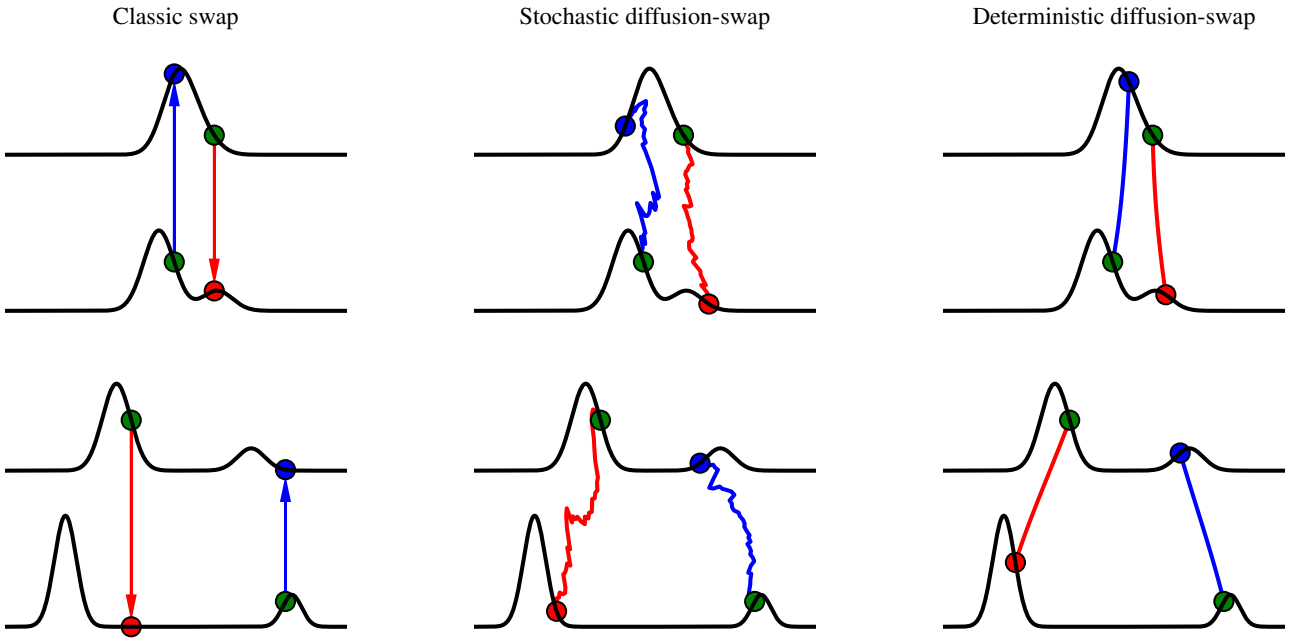


Figure 6.2: **Different diffusion-based swapping mechanisms for Replica Exchange. (Left)** standard swap scheme, see Section 6.2.2, where samples are exchanged directly across noise levels without guidance, potentially moving into low-probability regions. **(Middle)** DM-based swaps using forward and backward Markov kernels, as proposed by Zhang et al. [2026] (coined Diff-APT), see Section 6.3.2. **(Right)** DM-based swaps using forward and backward transport maps under the deterministic framework introduced in Section 6.4.1. In each panel, black lines denote noise levels; green dots mark the original samples; blue and red paths indicate forward (low to high noise) and backward (high to low noise) trajectories, respectively; and the swapped samples are shown as the resulting blue and green dots. The DM-based swaps better preserve high-probability regions during exchange, enabling theoretically more effective sampling.

Effective application to diffusion models. For all aMC methods, the choice of the forward maps $\{\mathrm{T}_{k+1|k}\}_{k=0}^{K-1}$ and backward maps $\{\mathrm{T}_{k|k+1}\}_{k=0}^{K-1}$ is optimal if those verify the deterministic version of the Bayes rule (6.16) given by

$$(\mathrm{T}_{k+1|k})_{\#}p_k = p_{k+1}, \quad (\mathrm{T}_{k|k+1})_{\#}p_{k+1} = p_k, \qquad \forall k \in \{0, \dots, K-1\} .$$

Indeed, in the case of AIS/SMC samplers, satisfying this identity would enable to get zero-variance in the estimator, while this would ensure to maximize the acceptance rate in the RE sampler. Intuitively, this rule reflects the fact that the maps should be chosen so as to perfectly transport particles between adjacent levels to match their target distribution.

The next section discusses two key challenges that arise when implementing these methods in practice using DM's ingredients :

1. *How to design transition maps that verify the invertibility condition* (6.20) *?*
2. *Given those maps, how to compute efficiently the push-forward densities appearing in* (6.21) *and* (6.22) *?*

6.4.2 The key components needed for efficient implementation

In this section, we first describe how to construct invertible transport maps that approximately solve the probability flow ODE (6.6). We then present a practical methodology, inspired by residual NFs, for obtaining unbiased estimates of the push-forward density terms appearing in (6.21) and (6.22). As made explicit by the change-of-variables formula (6.1), this approach requires (i) the ability to evaluate the transport maps and (ii) the computation (or unbiased estimation) of their Jacobian determinants. The construction directly extends to the case where the score in the PF-ODE is replaced by an estimate, by simply substituting the estimated score throughout. Our main contributions are summarized in Table 6.1. In what follows, we focus on two adjacent noise levels k and $k+1$.

Building invertible transport maps. To guarantee that the forward map $\mathrm{T}_{k+1|k}$ and the backward map $\mathrm{T}_{k|k+1}$ are mutually inverse, see (6.20), we cannot simply rely on *explicit* ODE integrators of the form (6.11). Indeed, forward and backward maps inherited from such first-order approximations do not, in general, compose to the identity[5]. This motivates us to move towards the class of *implicit* integrators, which are extensively used to simulate Hamiltonian dynamics where trajectory invertibility is often a desirable feature. In particular, we propose to design our transition maps via the *Implicit Midpoint* (IM) integrator, as presented in Proposition 8 for the Euler scheme.

Proposition 8 (IM integrator with Euler scheme)**.** *Let* $\mathrm{T}_{k+1|k} : \mathbb{R}^d \to \mathbb{R}^d$ *and* $\mathrm{T}_{k|k+1} : \mathbb{R}^d \to \mathbb{R}^d$ *be implicitly defined as*

$$\mathrm{T}_{k+1|k} : x_k \mapsto x_k + \delta_k v\left(t_{k+1/2}, \frac{x_k + \mathrm{T}_{k+1|k}(x_k)}{2}\right),$$
$$\mathrm{T}_{k|k+1} : x_{k+1} \mapsto x_{k+1} - \delta_k v\left(t_{k+1/2}, \frac{\mathrm{T}_{k|k+1}(x_{k+1}) + x_{k+1}}{2}\right),$$

where $t_{k+1/2} = (t_k + t_{k+1})/2$, $\delta_k = t_{k+1} - t_k$ *and* $v(t, x) = f(t)x - (g(t)^2/2)\nabla \log p_t(x)$ *is the velocity field of PF-ODE* (6.6)*. Then, these maps are valid forward and backward integrators of PF-ODE* (6.6) *on time interval* $[t_k, t_{k+1}]$ *and satisfy the mutual invertibility condition* (6.20)*.*

In Appendix E.3, we provide the proof of the above proposition along with its generalization using the Exponential Integration scheme in Appendix E.2, which offers improved accuracy compared to the Euler scheme when δ_k is relatively large. Although the maps defined in Proposition 8 cannot be evaluated in closed form as they are by nature *implicit*, they can still be approximated in practice using fixed-point iterations as described in Proposition 9, which guarantees convergence of this scheme under certain assumptions detailed below. We refer to Appendix E.3 for the proof of this result as well as Appendix E.2 for its EI generalization.

A 4 (Score smoothness & discretization error)**.** *(a) There exists* $L_k > 0$ *such that* $\nabla \log p_{t_{k+1/2}}$ *is* L_k*-Lipschitz and (b) the step-size* δ_k *is sufficiently small* [6]*, that is* $\delta_k = O(1/L_k)$*.*

Proposition 9 (Fixed-point approximation of the IM integrator)**.** *Following the same notation as in Proposition 8, under* ***A****4, for any inputs* x_k *and* x_{k+1}*, the sequences* $\{\mathrm{T}^{(n)}_{k+1|k}(x_k)\}_{n\in\mathbb{N}}$ *and* $\{\mathrm{T}^{(n)}_{k|k+1}(x_{k+1})\}_{n\in\mathbb{N}}$ *that are recursively defined as*

$$\mathrm{T}^{(0)}_{k+1|k}(x_k) = x_k, \quad \mathrm{T}^{(n+1)}_{k+1|k}(x_k) = x_k + \delta_k v\left(t_{k+1/2}, \frac{x_k + \mathrm{T}^{(n)}_{k+1|k}(x_k)}{2}\right), \tag{6.23}$$

$$\mathrm{T}^{(0)}_{k|k+1}(x_{k+1}) = x_{k+1}, \quad \mathrm{T}^{(n+1)}_{k|k+1}(x_{k+1}) = x_{k+1} - \delta_k v\left(t_{k+1/2}, \frac{\mathrm{T}^{(n)}_{k|k+1}(x_{k+1}) + x_{k+1}}{2}\right), \tag{6.24}$$

converge linearly to $\mathrm{T}_{k+1|k}(x_k)$ *and* $\mathrm{T}_{k|k+1}(x_{k+1})$*, respectively.*

In practice, we only compute the sequences from Proposition 9 up to a range $M \geq 1$ that ensures a prescribed fixed-point convergence tolerance $\varepsilon > 0$, that is, M is of the first order such that

$$\left\|\mathrm{T}^{(M+1)}_{k+1|k}(x_k) - \mathrm{T}^{(M)}_{k+1|k}(x_k)\right\|_2 \leq \varepsilon \text{ and } \left\|\mathrm{T}^{(M+1)}_{k|k+1}(x_{k+1}) - \mathrm{T}^{(M)}_{k|k+1}(x_{k+1})\right\|_2 \leq \varepsilon, \tag{6.25}$$

and we approximate $\mathrm{T}_{k+1|k}(x_k)$, resp. $\mathrm{T}_{k|k+1}(x_{k+1})$, by the $(M+1)$-th term $\mathrm{T}^{(M)}_{k+1|k}(x_k)$, resp. $\mathrm{T}^{(M)}_{k|k+1}(x_{k+1})$. While this iterative scheme may introduce numerical errors, we note that potential violations of the invertibility property (6.20) could be mitigated through an additional optional rejection step as proposed by Noble et al. [2023b]. We leave the implementation of such a safeguard to future work.

Estimating the Jacobian log-determinants. We now address the second component of (6.1), namely the computation of the Jacobian determinants (more precisely, their logarithm). Since this quantity is generally intractable, we propose a numerical procedure that yields an *unbiased* estimator, thereby preserving the statistical guarantees of aMC. In the case of RE, one can for instance show that the resulting estimator of the MH acceptance rate still guarantees that the target distribution remains invariant, see Andrieu and Roberts [2009]

[5] Note that this reasoning also applies in the case of EI-based first-order integrators.
[6] We provide the exact numerical constants related to this informal assumption in Appendix E.3.

Transition method	Forward design	Backward design	Needs $\nabla^2 \log p_k$
1st order kernel	Exact noising kernel (6.7)	EM/EI approx. (6.8)	✗
2nd order kernel	Exact noising kernel (6.7)	DDPM approx. (6.10)	✓
IM map via Hutchinson	Fixed-point approx. (6.23)	Fixed-point approx. (6.24)	✗
IM map via Hessian	Fixed-point approx. (6.23)	Fixed-point approx. (6.24)	✓

Table 6.1: **Summary of DM-based transitions used in annealed sampling methods.** This table presents stochastic transitions (top two lines), related to prior work, and deterministic transitions (bottom two lines), which we develop in Section 6.4, that can be built from a DM, to be then used within aMC samplers. The last column specifies whether access to the Hessians of the marginal log-densities is required. We recall that the acronym IM stands for *Implicit Midpoint*.

for details. Our approach exploits the recursive structure of the Jacobian associated with the IM integrator, expressing it as a power series, a technique previously explored in the context of contractive residual normalizing flows [Behrmann et al., 2019, Chen et al., 2019]. These considerations lead to the following proposition, whose proof is detailed in Appendix E.3.

Proposition 10 (Approximation of the Jacobian log-determinants via power series)**.** *Following the same notation as in Propositions 8 and 9, under **A**4, for any inputs x_k and x_{k+1}, and any prescribed fixed-point range $M \geq 1$ satisfying (6.25), the following approximation is obtained*

$$\log \left|\det J_{\mathrm{T}_{k+1|k}}(x_k)\right| \approx \sum_{i=0}^{I} a_{k,i} \operatorname{Tr}([A^{(M)}(x_k)]^i) \,,$$
$$\log \left|\det J_{\mathrm{T}_{k|k+1}}(x_{k+1})\right| \approx \sum_{i=0}^{I} b_{k,i} \operatorname{Tr}([B^{(M)}(x_{k+1})]^i) \,,$$

where $I \geq 1$ is a prescribed truncation order,

$$A^{(M)}(x_k) = \mathbf{H}_{t_{k+1/2}}\left(\frac{x_k + \mathrm{T}^{(M)}_{k+1|k}(x_k)}{2}\right), \quad B^{(M)}(x_{k+1}) = \mathbf{H}_{t_{k+1/2}}\left(\frac{x_{k+1} + \mathrm{T}^{(M)}_{k|k+1}(x_{k+1})}{2}\right),$$

$\mathbf{H}_{t_{k+1/2}}$ is the Hessian of $\log p_{t_{k+1/2}}$ and $\{a_{k,i}, b_{k,i}\}_{i=0}^{I}$ are non-zero numerical coefficients given in Proposition 99 (see Appendix E.2.1).

As a consequence of Proposition 10, our deterministic approach formally requires evaluating the Hessian of the midpoint log-densities (more specifically, their trace) and can therefore be viewed, from a theoretical standpoint, as a second-order approach. Nevertheless, we stress that the proposed method remains tractable in realistic settings where these Hessians are not explicitly available. Indeed, their traces can be efficiently approximated using the Hutchinson estimator [Hutchinson, 1989, Avron and Toledo, 2011], which relies on the identity $\mathrm{Tr}(\mathrm{M}) = \mathbb{E}_v\left[v^\top \mathrm{M} v\right]$ for any matrix $\mathrm{M} \in \mathbb{R}^{d\times d}$ and any a random vector $v \in \mathbb{R}^d$ satisfying $\mathbb{E}[v] = 0$ and $\mathrm{Cov}[v] = \mathrm{I}_d$ (in practice, one often chooses $v \sim \mathrm{N}(0, \mathrm{I}_d)$). This approach only requires vector–Jacobian products, which can be computed efficiently via reverse-mode automatic differentiation, and provides an unbiased estimate of the trace, resulting in an *unbiased estimate of the Jacobian log-determinants*. While Behrmann et al. [2019] show that the error induced by truncating the expansion series can be safely bounded, Chen et al. [2019] propose using a Russian roulette estimator to further reduce the variance. We leave the exploration of such variance reduction techniques for future work.

6.4.3 Empirical comparison between DM-based stochastic and deterministic transitions in aMC samplers

In Figure 6.3, we report the performance of the deterministic methodology applied to AIS, SMC and RE in idealized setting **(A)** for the *TwoModes* target, and compare it to methods that are based on stochastic kernels (first-order and second-order). In Appendix E.4.3, we provide further results in all considered multi-modal scenarios. Based on those results, we can make the following observations:

(i) *When the Hessian is available*, using deterministic transitions (pink bars) performs on par with the second-order stochastic approach (green bars). Interestingly, it appears that the deterministic method provides even better results for low values of K with AIS/SMC samplers, offering a significant computational advantage. In the case of RE, we however note a relative under-performance of the deterministic approach, which we believe to be due to the amplification of estimation error of the two Jacobian log-determinants (while AIS/SMC only requires the estimation of one term) with the expansion series.

(ii) *When the Hessian is not available*, the first-order deterministic variant relying solely on the score functions via the Hutchinson estimator (yellow bars) consistently improves over the standard baseline (red bars) and the use of first-order stochastic kernels (blue bars) presented in prior work, for each value of K. This highlights the promise of deterministic mappings in aMC samplers. Remarkably, we observe that the performance gap with the second-order deterministic scheme is relatively small in all multi-modal scenarios, proving the effectiveness of the Hutchinson estimation to fully exploit first-order information. Remarkably, the computational overhead induced by the computation of the log-determinant expansion series is very limited for AIS/SMC sampler; on the contrary, it incurs a non negligible burden for RE sampler.

Remark on second-order stochastic kernels. For the Gaussian kernels in (6.10), the Hessian appears in the covariance term. As a result, sampling only involves Jacobian–vector products, which can be handled with standard automatic differentiation tools. In contrast, likelihood evaluation additionally requires inverse–Jacobian–vector products through the term $[\Sigma_{s|t}(x_t)]^{-1}(x_s - m_{s|t}(x_t))$, which is substantially more challenging to implement efficiently. While recent work has begun to address this computational bottleneck [Siskind, 2019], developing practical implementations is an open and promising direction for future research.

6.5 Related Works

Normalizing flows into annealed sampling. Normalizing flows have been previously integrated into aMC frameworks. For instance, Arbel et al. [2021] and Matthews et al. [2022] incorporate flows as forward and backward kernels within AIS and SMC algorithms combined with tempering density paths Other works such as Midgley et al. [2021, 2023b,a] consider AIS schemes where the sequence of densities use π^{base} as a NF, allowing for better conditioned path. On the other hand, Invernizzi et al. [2022] propose an extension of RE of the form of (6.22), with the key difference being that their deterministic transformations are parameterized by NFs rather than derived from DM-based dynamics.

Using DMs in aMC for sampling. This idea has recently seen a growing interest in the generative modeling community. Some works have built upon the AIS backbone with specific choices of transition operators. For instance, Zhang et al. [2024b] propose to design both forward and backward stochastic kernels as a mix of exact noising kernels and first-order *explicit* integrators of the PF-ODE, in order to take advantage of the efficiency of deterministic mappings. On the other hand, Zhang et al. [2025] design the backward transition kernels as Gaussian denoising kernels with a flexible scalar variance that is learned, in the same spirit as second-order kernels. Taking SMC as a sampling backbone, Phillips et al. [2024] present a end-to-end algorithm that aims to sample from a target distribution by learning the corresponding DM. This procedure alternates between (i) building a BG toward the target via DM-based SMC (here, the backward transitions are defined as first-order EI kernels) and (ii) updating this DM by minimizing a score matching objective with the samples from stage (i). To be able to evaluate the intermediate log-densities, the DM is parameterized as a multi-level energy-based model. More recently, Zhang et al. [2026] explore the use of DM-based kernels as forward and backward stochastic transitions within a RE framework. Similarly to Phillips et al. [2024], they propose an iterative sampling approach , that involves RE combined with first-order stochastic kernels.

Combination of annealed sampling and DMs beyond BGs. Diffusion models have also been combined with aMC methods, though not primarily for building BGs. Instead, these approaches leverage DMs for various downstream tasks. For instance, SMC-based approaches have been proposed for conditional generation [Wu et al., 2023], posterior sampling in Bayesian inverse problems [Cardoso et al., 2024, Dou and Song, 2024, Janati et al., 2024, 2025], reward-guided generation and fine-tuning [Uehara et al., 2024, Kim et al., 2025d, Singhal et al., 2025], as well as compositional and controlled generation tasks [Thornton et al., 2025, Skreta et al., 2025]. While these methods rely on advanced sampling techniques, their primary focus lies in enhancing/extending generation capabilities rather than reweighting DMs with respect to a given target unnormalized density.

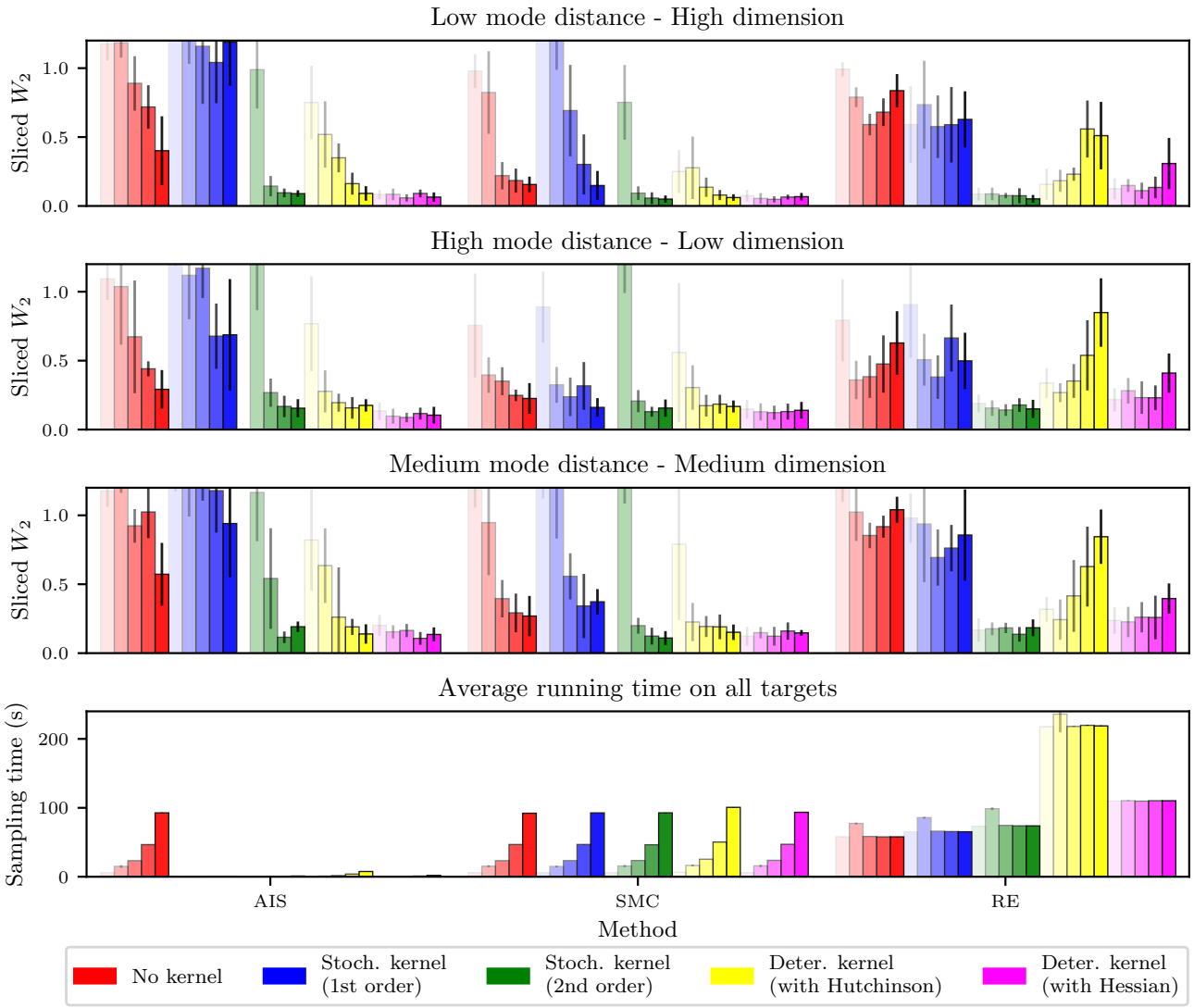


Figure 6.3: **DM-based aMC-BG results with annealed samplers using different mechanisms**, when targeting *TwoModes* distribution in idealized setting **(A)** : **(First row)** Low distance, high dimension, **(Second row)** High distance, low dimension, **(Third row)** Middle distance, middle dimension, **(Fourth row)** Average running time of each sampler over the three *TwoModes* variants. Each group of bars with the same color corresponds to a specific aMC method. Within each group, red bars refer to methods that do not use DM-based transition kernels (standard baseline); blue and green refer to methods that exploit 1st-order and 2nd-order stochastic kernels (prior work); pink and yellow correspond to variants with deterministic maps, where the log-determinant term is respectively computed from the ground truth diagonal Hessian or via the Hutchinson trick. For all settings, we display the results for all values of K : the darker the bar, the higher K (same range as Figure 6.1). In particular, configurations within each bar group do not share the same computational budget. Each result is averaged over 8 runs with 8,192 samples per run. We observe that using 1st-order stochastic kernels (in blue) does not always lead to better performance than the baseline (in red), while 2nd-order stochastic kernels (in green) provide consistent improvements. Moreover, using deterministic transitions combined with the Hutchinson trick (in yellow) brings better performance than 1st-order stochastic kernels (both only rely on the use of the scores), while having access to the Hessians (in pink) performs comparably to the 2nd-order stochastic methods. In the case of AIS/SMC sampler, using deterministic transitions incur only a little computational overhead compared to the stochastic case, but this cost is stringer for RE sampler. Similar results are given in Appendix E.4.3 for *ManyModes* targets.

6.6 Numerical experiments in a realistic setting

In this section, we evaluate the performance of DM-based aMC-BGs in realistic setting **(B)**. This implies that the true dynamics are no longer available and are instead replaced by estimated dynamics driven by learned log-densities and scores. In particular, we assume that we do not have access to second-order information (*i.e.*, the Hessians of the log-densities), as it is often the case in practice. The purpose of this approach is to compare the practical performance of these samplers with their ideal counterparts described in Section 6.3 and Section 6.4, which are affected only by statistical and time-discretization errors. We detail precisely our experimental framework in Section 6.6.1, where we consider three versions of aMC methods : (a) using only learned log-densities (standard aMC setting), using learned log-densities and scores with (b) first-order stochastic transition kernels (as done in prior work) or (c) first-order deterministic transition maps (as proposed in Section 6.4).

6.6.1 Log-density and score learning framework

Architecture design. To evaluate DM-aMC BGs under realistic constraints, we first learn DM log-densities and scores simultaneously using available samples. To do so, we model the log-density $\log p_t(x)$ by a scalar-valued neural network $(t, x) \mapsto -\mathcal{E}_t^\theta(x)$, and deduce an approximation of the score function $\nabla \log p_t(x)$ by taking the negative gradient of $\mathcal{E}^\theta$, denoted by $\mathbf{s}_t^\theta(x) = -\nabla_x \mathcal{E}_t^\theta(x)$. To ensure correctness at $t = t_0$ close to 0, we compare two common architectures.

(a) *Pinned*: we first consider the pinned architecture [Phillips et al., 2024, Zhang et al., 2026], defined as

$$\mathcal{E}_t^\theta(x) = (1 - f^\theta(t) + f^\theta(t_0))\mathcal{E}(x) + (f^\theta(t) - f^\theta(t_0))g_t^\theta(x) \ , \tag{6.26}$$

where $f^\theta : [0, T] \to \mathbb{R}$ is a neural network that solely takes time as input, and $g^\theta : [0, T] \times \mathbb{R}^d \to \mathbb{R}$ is another neural network conditioned on both t and x. While this setting ensures exact recovery of the target distribution π at t_0, it is known to be difficult to train [Du et al., 2025], motivating the consideration of the next architecture.

(b) *Hardcoded*: the second architecture is an unconstrained variant inspired by the preconditioned score network used in [Karras et al., 2022, Thornton et al., 2025]. Since it does not enforce any boundary condition at $t = t_0$, we explicitly correct this during sampling by replacing $\mathbf{s}_{t_0}^\theta$ with $-\nabla\mathcal{E}$ and $\mathcal{E}_{t_0}^\theta$ with $\mathcal{E}$. While this approach offers more flexibility during training, it may lead to inaccurate behavior at inference.

Loss design. For each neural network, we consider six learning approaches. We restate their expression in Appendix E.1.4 and provide training details in Appendix E.4.2. These losses are denoted as follows:

- (DSM): *Denoising Score Matching* objective [Song et al., 2021b, Karras et al., 2022],
- (TSM+DSM): *Target Score Matching* objective [Bortoli et al., 2024a] with DSM regularization,
- (tSM+DSM): *Time Score Matching* objective [Yu et al., 2025, Guth et al., 2025] with DSM regularization,
- (LFPE+DSM): DSM objective with *Log-density Fokker Planck Equation* regularization [Shi et al., 2024],
- (aLFPE+DSM): DSM objective with *approximated LFPE* regularization [Plainer et al., 2025],
- (RNE+DSM): DSM objective with *Radon-Nikodym Estimator* regularization [He et al., 2026].

6.6.2 DM-BGs via aMC seem inherently limited by log-density approximation

DM-BGs fail in practice. Figure 6.4 compares all first-order DM-based aMC-BGs in realistic setting **(B)**, for a subselection of DM training objectives presented above. It additionally displays

- tempering-based classic aMC samplers, which have been proved to be less accurate than DM-based classic aMC samplers in idealized setting **(A)** (see Section 6.3.1),
- ideal DM-base aMC-BGs from setting **(A)**, that only exploit first-order information, either with stochastic (see Section 6.3.2) or deterministic kernels (see Section 6.4),
- simulations of reverse DM stochastic and deterministic dynamics, which only use the score functions.

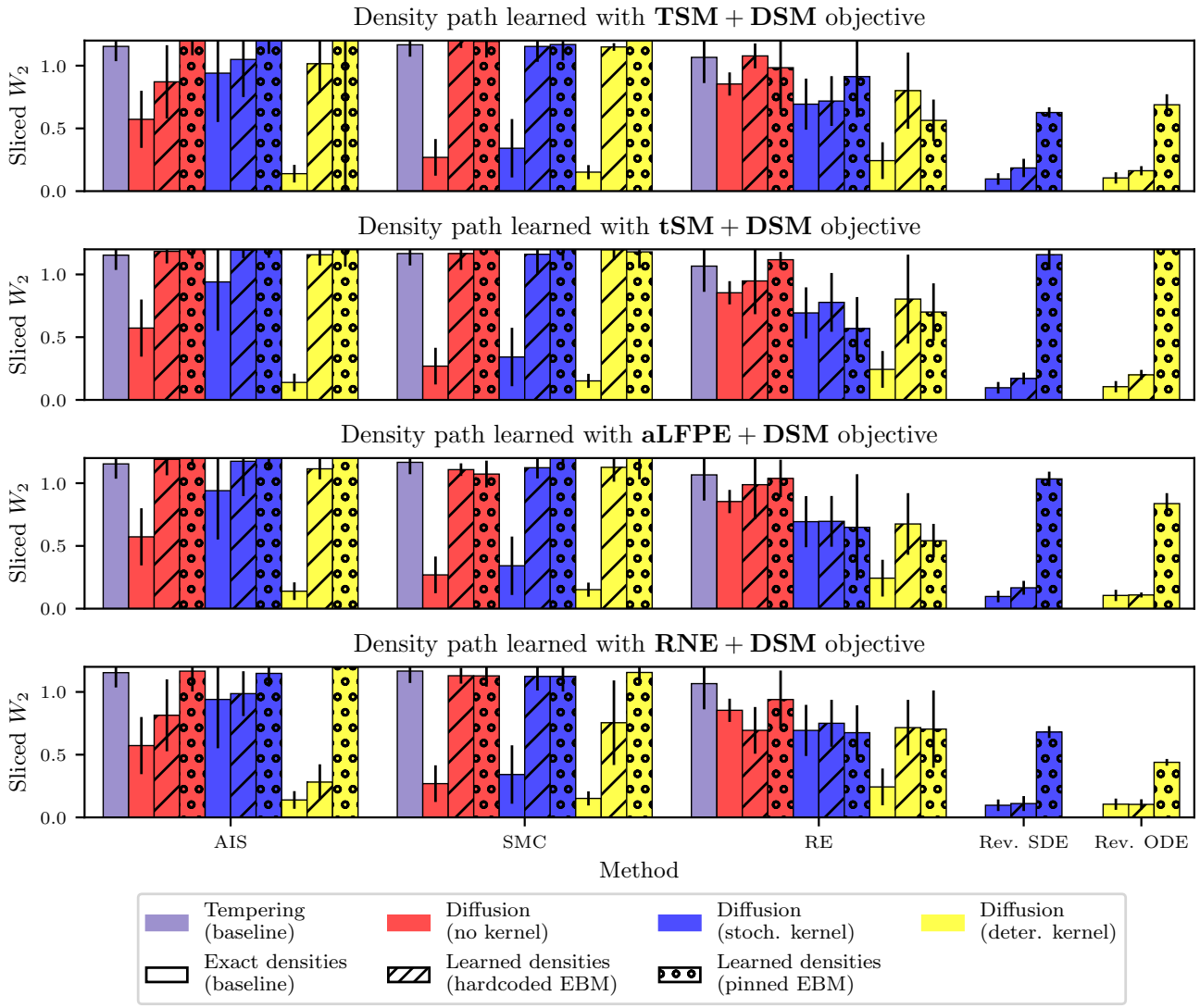


Figure 6.4: **Realistic results of DM-based aMC-BG**, when targeting *TwoModes* distribution with intermediate difficulty (middle mode distance, middle dimension) in setting **(B)** : **(From top to bottom)** the DM is trained via TSM+DSM, tSM+DSM, aLFPE+DSM or RNE+DSM objective with identical computational budget. Each group of bars with the same color corresponds to a specific aMC method, except for the last two groups on the right which shows the baseline obtained by directly simulating the reverse SDE (6.5) and reverse ODE (6.6). Bar colors indicate the type of density path used for the sampling method, and are consistent with those displayed in Figure 6.3: purple for the model-free tempering path (same baseline for all models), red for the diffusion density path solely exploiting the log-densities, blue, respectively yellow, for the diffusion density path additionally exploiting DM first-order stochastic kernels, respectively DM first-order deterministic kernels. On the other hand, hatching denotes the nature of log densities used in aMC (exact as in Figure 6.3, learned with hardcoded EBM or learned with pinned EBM). For each method and density path, the number of levels K was optimized individually so as to display the best expected result from the considered approach if K was carefully tuned. The results show a clear performance gap between learned and ideal paths. Notably, DM-BGs generally underperform compared to directly to the DM alone (*i.e.*simulating the reverse SDE/ODE), and rarely surpass the classic tempering methods. Similar results are given in Appendix E.4.3 for other *TwoModes* and *ManyModes* targets, and the other DM training objectives.

Within each setting, we present the best results obtained among all values of the number of levels $K \in \{16, 32, 64, 128, 256\}$ to improve clarity. Figure 6.4 notably demonstrates that the Harcoded architecture consistently and largely underperform compared (i) to their analogs of idealized setting **(A)** but also compared to (ii) the learned reverse SDE/ODE baseline (which nonetheless matches its ideal analog). Hence, these observations highlight that the poor performance of corresponding aMC methods is not imputable to the quality of the learned scores, but rather to the learned log-densities. In the case of the Pinned architecture, our findings are even more negative, since the simulated reverse SDE/ODE additionally produces highly biased samples, proving that the score functions are not well learned in the considered multi-modal scenario. Finally, we note that all DM-based aMC-BGs approaches often fail to improve (and sometimes degrade) results from standard tempering approaches that do not use any DM.

DM-BG failure cases could be attributed to mode switching in learned log-densities. We hypothesize that a primary contributor to the poor performance of learning-based approaches is the inherent *mode blindness* of presented DM training objectives, which likely leads to mode switching in the learned density path. This issue, well-documented in Wenliang and Kanagawa [2021], Zhang et al. [2022], Shi et al. [2024], is a known limitation of score-based methods to learn accurately log-densities, and affects all divergences derived from the Fisher divergence or Stein discrepancy. Indeed, when applied on multi-modal distributions with well-separated modes, these divergences fail to distinguish distributions that have the same mode locations but differ in mode proportions. This is because the score, being independent of the normalizing constant, is unaware of other modes when evaluated within a single mode. As a result, learning the log-density via score matching often produces the correct shape within each mode (*i.e.*, accurate gradients) but with incorrect relative weights. In Figure 6.6, we illustrate this phenomenon for the same subset of DM training objectives as in Figure 6.4 for 1D Gaussian mixtures (all results being given in Appendix E.4.3). The obtained results can be compared with the ground truth diffusion and tempering density paths displayed in Figure 6.5.

The ubiquitous curse of mode blindness for log-density estimation ?

Although mode blindness has been empirically documented for the DSM objective, we emphasize that it also applies to the TSM objective too, as a consequence of its score-based formulation. Nevertheless, recent diffusion-based sampling methods have employed TSM to learn log-densities [Phillips et al., 2024, Zhang et al., 2026] for using DMs within aMC. By the same reasoning, mode blindness also affects log-density estimation via score-based distillation losses [Thornton et al., 2025, Akhound-Sadegh et al., 2025], probably hurting the accuracy of SMC sampling for related inference-time alignment tasks. Based on the results displayed in Figure 6.6 and their score-based formulation (see Appendix E.1.4), we conjecture that mode blindness is prone to occur when using LFPE-based regularizations, despite their adoption in multiple diffusion-based sampling frameworks [Shi et al., 2024, Plainer et al., 2025, Sun et al., 2024]. Although mode blindness seems less straightforward for tSM and RNE objectives, we believe it is still a substantial barrier to accurate log-density estimation, as depicted by Figure 6.6, and leave the theoretical investigation of this question to future work.

Overall, we conjecture that mode switching significantly hinders aMC methods, as well for learned diffusion density paths as for tempering density paths. In SMC (including DM-enhanced variants), resampling must continually correct for imbalanced mode weights, which becomes increasingly challenging in high dimensions. Similarly, in RE, communication between chains is disrupted when mode alignment across levels is inconsistent, although we observe that it may be compensated for by the possibility of moving back and forth between levels during sampling procedure. This instability explains the poor performance of the learned path in Figure 6.4, even when the forward and backward transition kernels (both deterministic and stochastic) are accurate due to well-learned scores.

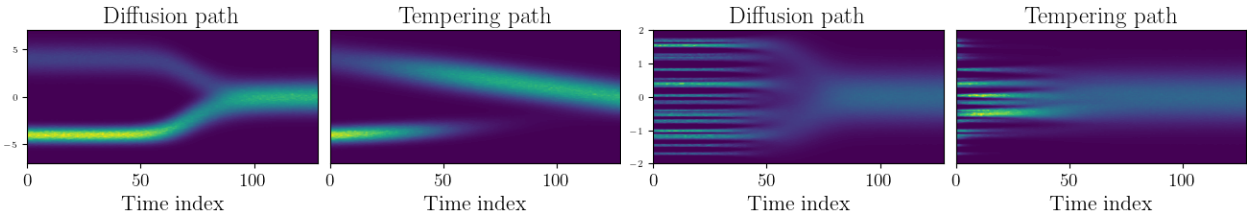


Figure 6.5: **Exact density paths bridging π^{base} (last time index) to 1D Gaussian mixtures (first time index). (Left)**: the target is an instance of *TwoModes* defined as $(3/4)\mathrm{N}(-4, 0.5^2) + (1/4)\mathrm{N}(+4, 1)$, **(Right)** the target is the 1D instance of *ManyModes* Noble et al. [2025] with 32 modes, **(First and third columns)** diffusion density path, **(Second and fourth columns)** tempering density path. Given a discretization of 128 timesteps in [0,1], we plot each of the induced marginal densities using importance sampling with 1,000,000 particles and a uniform proposal. We observe that the tempering path shows clear mode switching for both of the targets: in the case of the *TwoModes* target, the strongest mode emerges abruptly, while the weakest modes appear rapidly for the *ManyModes* target. On the other hand, the mode weights in the diffusion path remain stable over time, making it more favorable for aMC.

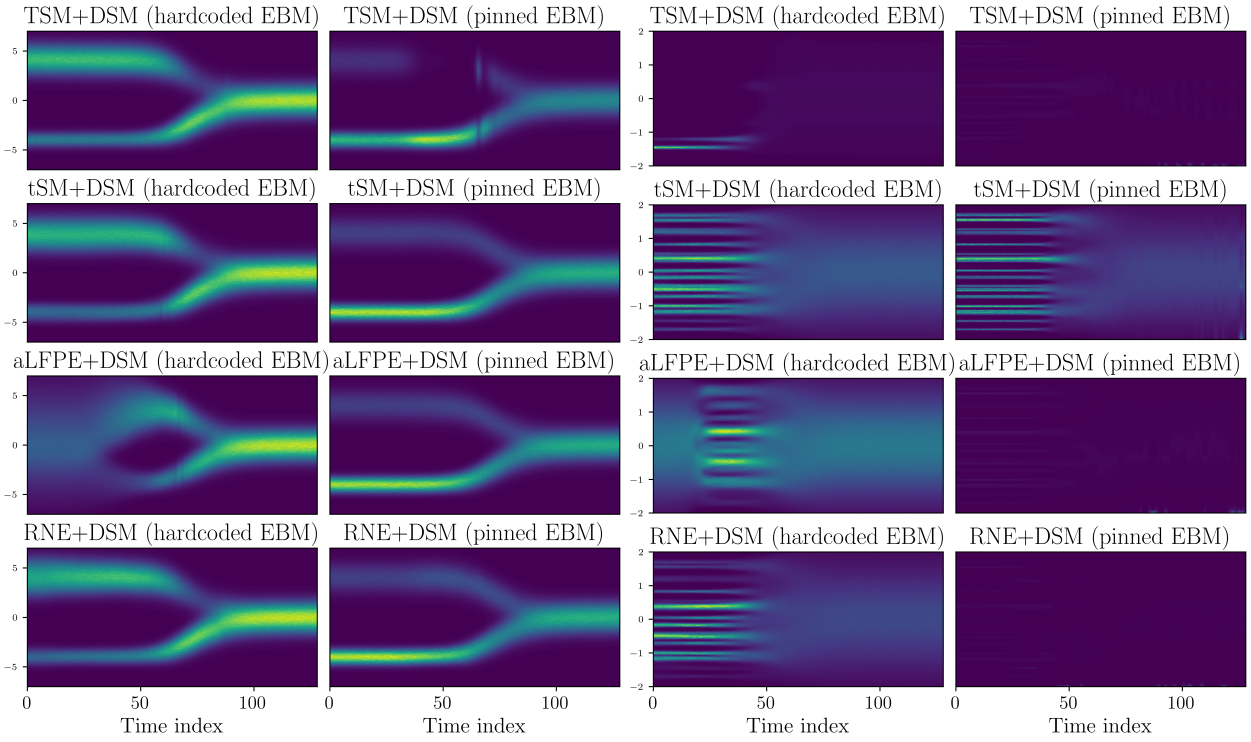


Figure 6.6: **Learned diffusion density paths bridging π^{base} (last time index) to the same targets as in Figure 6.5. (From top to bottom)** the DM is trained via TSM+DSM, tSM+DSM, aLFPE+DSM or RNE+DSM objective, with identical computational budget, **(Left)** *TwoModes* target, **(Right)** *ManyModes* target, **(First and third columns)** use of Hardcoded EBM, **(Second and fourth columns)** use of Pinned EBM. The same plotting configuration is used as in Figure 6.5. When using the Hardcoded architecture for both targets, we observe that the density is well learned for large times (near π^{base}), but systematically fails to recover the exact mode weights for small times, thus highlighting the mode blindness of related training objectives. While using the Pinned architecture enables to closely recover the exact diffusion path for the *TwoModes* target, we observe that this strategy is not successful when increasing the number of modes, with the notable exception of tSM+DSM objective. This under-performance of the Pinned architecture may notably explain its systematic failure in our experiments run in realistic setting **(B)**, where the targets are highly challenging.

6.7 Conclusion & Limitations

This work revisits the design of Boltzmann Generators by replacing the standard normalizing-flow/importance-sampling backbone with a diffusion-model backbone embedded in annealed Monte Carlo. We first unify and review prior DM-aMC approaches, which exploit diffusion-induced *stochastic* denoising kernels to facilitate transitions between annealing levels, and we then introduce and study *deterministic* counterparts based on diffusion-derived transport maps. To compare these methods, we conduct an empirical study on multi-modal target distributions, emphasizing challenging characteristics such as inter-mode separation, number of modes, and dimensionality. Our analysis proceeds in two stages: we (i) isolate inference effects by assuming a perfectly learned DM, and (ii) turn to a realistic setting where the DM is trained from data.

In the idealized regime, empirical metrics reveal a non-zero discrepancy between the ground-truth target and the distribution induced by the resulting BG, despite perfect model knowledge. This indicates that aMC inference error alone can produce measurable bias. In this setting, *first-order* stochastic denoising kernels (score-only) often fail to improve over standard aMC baselines, whereas *second-order* kernels (incorporating Hessian information) and deterministic transitions yield substantially better results. Importantly, our deterministic construction based on a Hutchinson-type estimator remains competitive even without explicit Hessian access, suggesting that deterministic transport can recover much of the benefit of second-order information while relaxing its most demanding requirement.

In the learned regime (e.g., score-matching-like training objectives), the picture changes markedly: the resulting BGs systematically fail across our multi-modal benchmarks, even when the learned scores appear accurate. Our experiments point to inaccuracies in DM log-density estimation as the primary culprit. Specifically, the obtained estimates are mostly mode-blind, as they fail to accurately represent relative mode proportions along the diffusion path in regions where the modes are well separated. As a consequence, such errors directly disrupt sampling and can dominate any gains from improved transitions. In other words, high-quality score estimates are not sufficient to guarantee successful BG construction when the correction step relies on unreliable log-density approximations.

In the spirit of Grenioux et al. [2025], our goal is not to demonstrate scalability but to expose and analyze the fundamental limitations of diffusion-based aMC-BGs in a simple, fully controlled benchmark. The underlying rationale is that methods that do not succeed in these elementary multi-modal settings are unlikely to behave reliably on more complex targets with many modes or ill-conditioned energy landscapes. Accordingly, this work emphasizes failure mechanisms over performance claims, consistent with our largely negative conclusions.

Several extensions remain natural. First, moving beyond Gaussian mixtures to more challenging targets is an important next step. Second, we did not include adaptive SMC and RE-style algorithms in the main comparisons. With larger budgets (e.g., more intermediate distributions or carefully tuned tempering schedules), such methods could potentially narrow the observed gaps. Third, diffusion-based methods may depend on the noise schedule: for completeness we derive in Appendix E.2.3 the formulas for the VE noising scheme, and we leave a systematic numerical study of its impact to future work.

Finally, the promising behavior of deterministic (as opposed to stochastic) transitions motivates two concrete research directions. On the inference side, we will further study the deterministic construction of Section 6.4 to reduce sensitivity to Hessian-trace approximations, including the use of more robust estimators [Liu et al., 2025b]. On the modeling side, the main bottleneck in realistic settings is the mode blindness of current DM log-density estimation techniques; addressing this issue appears necessary for reliable sampling. Beyond building a single BG, if one can design training schemes that mitigate or eliminate mode blindness, the proposed aMC machinery could serve as an inner loop in iterative diffusion-based training procedures tailored for sampling, departing from the one-shot correction perspective, in the spirit of adaptive, data-free training strategies [Gabrié et al., 2022, Phillips et al., 2024, Akhound-Sadegh et al., 2024].

Acknowledgments

We warmly thank Marylou Gabrié, Alain Durmus, and José Miguel Hernández-Lobato for the insightful discussions and reflections that helped shape and refine this work. Their perspectives and feedback have been invaluable throughout the development of the ideas presented here. This work was performed using HPC resources from GENCI–IDRIS (AD011014860R1, AD011014860R2 and AD011015234R1). This work received government funding managed by the French National Research Agency under France 2030, reference ANR-23-IACL-0005.

Part III

Optimal transport methodologies based on diffusion models

Chapter 7

Tree-based Diffusion Schrödinger Bridge with applications to Wasserstein Barycenters

7.1 Introduction

In the last decade, computational Optimal Transport (OT) has shown great success with applications in various fields such as biology [Schiebinger et al., 2019, Bunne et al., 2022], shape correspondence [Su et al., 2015, Feydy et al., 2017, Eisenberger et al., 2020], control theory [Bayraktar et al., 2018, Acciaio et al., 2019] and computer vision [Schmitz et al., 2018, Carion et al., 2020]. While OT commonly seeks at computing the transport plan that minimizes the cost of moving between two distributions, it can naturally be extended to the multi-marginal setting (mOT) when considering several distributions. This extension of OT has notably been studied in quantum chemistry [Cotar et al., 2013], clustering [Cuturi and Doucet, 2014] and statistical inference [Srivastava et al., 2018]. In particular, a popular application in unsupervised learning of mOT with Euclidean cost consists in computing the Wasserstein barycenter of a set of probability distributions [Agueh and Carlier, 2011, Benamou et al., 2015, Álvarez-Esteban et al., 2016, Peyré et al., 2019].

Interior point methods can be used to solve OT and mOT problems but they come with computational challenges [Pele and Werman, 2009]. In order to mitigate these limitations, one often considers an *entropic regularization* of OT, known as Entropic OT (EOT). This regularized formulation can be efficiently solved in discrete state-spaces using the celebrated *Sinkhorn* algorithm [Cuturi, 2013, Knight, 2008, Sinkhorn and Knopp, 1967], which admits a continuous state-space counterpart referred to as the *Iterative Proportional Fitting* (IPF) procedure [Fortet, 1940, Kullback, 1968, Ruschendorf, 1995]. In the case of a quadratic cost, EOT is equivalent to the *static* formulation of the Schrödinger Bridge (SB) problem [Schrödinger, 1932]. Given a reference diffusion with finite time horizon T and two probability measures, solving SB amounts to finding the closest diffusion to the reference (in terms of Kullback–Leibler divergence on path spaces) with the given marginals at times $t = 0$ and $t = T$. This framework naturally arises in stochastic control [Dai Pra, 1991] where one aims at controlling the marginal distribution of a stochastic process at a fixed time. Recently, Bortoli et al. [2021] introduced Diffusion Schrödinger Bridge (DSB), an approximation of a *dynamic* version of the IPF scheme on path spaces, see also Vargas et al. [2021], Chen et al. [2022]. This methodology leverages advances in the field of denoising diffusion models [Song et al., 2021b, Ho et al., 2020] in order to derive a scalable and efficient scheme to solve SB, and thus EOT.

Similarly to OT, mOT admits an entropic regularization (EmOT), which can be solved via a multi-marginal generalization of Sinkhorn/IPF algorithm [Benamou et al., 2015, Marino and Gerolin, 2020]. Recently, Haasler et al. [2021] proposed an extension of the *static* SB problem in *discrete* state-space to any multi-marginal tree-based setting. They notably made the correspondence between this formulation and EmOT, when the cost function writes as the sum of interaction energies onto the given tree structure, and introduced an efficient version of Sinkhorn algorithm to solve it.

Motivations and contributions. In this work, we investigate the *continuous* and *dynamic* counterpart of the tree-based framework from Haasler et al. [2021]. To be more specific, we present an extension of the static SB formulation in continuous state-space to any multi-marginal tree-based setting, referred to as TreeSB. Then, we establish the equivalence between TreeSB and a formulation of EmOT relying on a (quadratic) tree-structured cost function, analogously to Haasler et al. [2021]. Inspired by DSB, we develop TreeDSB, a dynamic counterpart of the multi-marginal IPF (mIPF) to solve it, by operating on path spaces and using score-based diffusion techniques. To bridge gaps in literature, we prove the convergence of mIPF iterations in a *non-compact* setting under mild assumptions, by extending results on IPF convergence [Ruschendorf, 1995]. Finally, we illustrate TreeDSB on examples of Wasserstein barycenters from statistical inference and image processing.

Although our approach can be applied to any tree, we focus on *star-shaped trees*. In this setting, we show that TreeSB reduces to a regularized Wasserstein barycenter problem. Our method comes with several benefits compared to existing works. First, it is out-of-sample, *i.e.*, it does not require re-running the full procedure when given a new data point. Second, our formulation of the Wasserstein barycenter problem obtained from TreeSB allows us to avoid numerical issues of having to choose the regularization too small, see Section 7.5. Finally, to the best of our knowledge, this is the first methodology to extend ideas from diffusion-based models to the computation of Wasserstein barycenters. We notably believe that the idea of iterative refinement, *i.e.*, solving the *dynamic* counterpart of a *static* problem, plays a key role in the efficiency and scalability of the method.

Notation. For any measurable space $(\mathsf{X}, \mathcal{X})$, we denote by $\mathscr{P}(\mathsf{X})$ the space of probability measures defined on $(\mathsf{X}, \mathcal{X})$. Unless specified, $\mathcal{X}$ is defined as the Borel sets on X. For any $\ell \in \mathbb{N}$, let $\mathscr{P}^{(\ell)} = \mathscr{P}((\mathbb{R}^d)^\ell)$; we denote $\mathscr{P}^{(1)}$ by $\mathscr{P}$. Assume that $\mathsf{X} = (\mathbb{R}^d)^\ell$ for some $\ell \in \mathbb{N}$. For any $x \in \mathsf{X}$ and any $m, n \in \{0, \dots, \ell\}$ such that $m \leq n$, let $x_{m:n} = (x_m, x_{m+1}, \dots, x_n)$. Let Leb be the Lebesgue measure. For any non-negative function $f : \mathsf{X} \to \mathbb{R}_+$, such that $\int_{\mathsf{X}} f \mathrm{dLeb} < +\infty$, define $\mathrm{H}(f) = -\int_{\mathsf{X}} f \log f \mathrm{dLeb} \in (-\infty, +\infty]$. For any distribution $\mu \in \mathscr{P}(\mathsf{X})$, we define the entropy of μ as $\mathrm{H}(\mu) = \mathrm{H}(\mathrm{d}\mu/\mathrm{dLeb})$ if $\mu \ll \mathrm{Leb}$ and $\mathrm{H}(\mu) = +\infty$ otherwise. For any two arbitrary measures μ and ν defined on $(\mathsf{X}, \mathcal{X})$, define the Kullback–Leibler divergence between μ and ν as $\mathrm{KL}(\mu|\nu) = \int_{\mathsf{X}} \log(\mathrm{d}\mu/\mathrm{d}\nu)\mathrm{d}\mu - \int_{\mathsf{X}} \mathrm{d}\mu + \int_{\mathsf{X}} \mathrm{d}\nu$ if $\mu \ll \nu$ and $\mathrm{KL}(\mu \mid \nu) = +\infty$ otherwise. For any $T > 0$, we denote by $\mathrm{C}([0, T], \mathbb{R}^d)$ the space of continuous functions from $[0, T]$ to $\mathbb{R}^d$. For any path measure $\mathbb{P} \in \mathscr{P}(\mathrm{C}([0, T], \mathbb{R}^d))$, we denote by $\mathrm{Ext}(\mathbb{P}) \in \mathscr{P}^{(2)}$ the coupling between the *extremal* distributions of $\mathbb{P}$, *i.e.*, $\mathrm{Ext}(\mathbb{P}) = \mathbb{P}_{0,T}$. Note that, for a given coupling $\pi_{0,T} \in \mathscr{P}^{(2)}$, there may exist several path measures $\mathbb{P}$ verifying $\mathrm{Ext}(\mathbb{P}) = \pi_{0,T}$. For any undirected tree $\mathsf{T} = (\mathsf{V}, \mathsf{E})$ with vertices V and edges E, we denote by $\{v, v'\}$ (or $\{v', v\}$) the undirected edge between $v \in \mathsf{V}$ and $v' \in \mathsf{V}$, if it exists. Given $r \in \mathsf{V}$, we denote by $\mathsf{T}_r = (\mathsf{V}, \mathsf{E}_r)$ the directed version of T rooted in r, where the directed edges E_r are uniquely defined from the edges E, see Appendix F.2 for further details. In this case, the edge linking $v \in \mathsf{V}$ to $v' \in \mathsf{V}$ in T_r is denoted by (v, v'). Finally, for any integers $(n, K) \in \mathbb{N} \times \mathbb{N}^*$, we define $n \bmod(K)$ as the the remainder of the Euclidean division of n by K.

7.2 Background and setting

Multi-marginal optimal transport. Let $\ell \in \mathbb{N}^*$. Given a cost function $c : (\mathbb{R}^d)^{\ell+1} \to \mathbb{R}$, a subset $\mathsf{S} \subset \{0, \dots, \ell\}$ and a family of probability measures $\{\mu_i\}_{i \in \mathsf{S}} \in \mathscr{P}^{|S|}$, mOT consists in solving

$$\pi^\star = \arg\min \left\{ \int c(x_{0:\ell}) \mathrm{d}\pi(x_{0:\ell}) : \pi \in \mathscr{P}^{(\ell+1)},\ \pi_i = \mu_i\ , \forall i \in \mathsf{S} \right\}, \tag{mOT}$$

where π_i is the i-th marginal of π, *i.e.*, $\pi_i(\mathsf{A}) = \pi(\mathrm{proj}_i^{-1}(\mathsf{A}))$ for any $\mathsf{A} \in \mathcal{B}(\mathbb{R}^d)$, with $\mathrm{proj}_i : x_{0:\ell} \mapsto x_i$. Given some weights $(w_i)_{i \in \{1,\dots,\ell\}} \in (\mathbb{R}_+)^\ell$, the Wasserstein barycenter between the measures $\{\mu_i\}_{i \in \mathsf{S}}$ is given by $\pi_0^\star$ in (mOT), in the case where $\mathsf{S} = \{1, \dots, \ell\}$ and $c(x_{0:\ell}) = \sum_{i=1}^\ell w_i \|x_0 - x_i\|^2$ [Peyré et al., 2019]. In particular, when $w_i = 1/\ell$, the distribution $\pi_0^\star$ can be regarded as the Fréchet mean [Karcher, 2014] of the measures $\{\mu_i\}_{i \in \mathsf{S}}$ for the Wasserstein distance of order 2. Similarly to OT, (mOT) can be relaxed using the following entropic regularization

$$\pi^\star = \arg\min \left\{ \int c(x_{0:\ell}) \mathrm{d}\pi(x_{0:\ell}) + \varepsilon \mathrm{KL}(\pi|\nu) : \pi \in \mathscr{P}^{(\ell+1)},\ \pi_i = \mu_i\ , \forall i \in \mathsf{S} \right\}, \tag{EmOT}$$

where $\varepsilon > 0$ is a hyperparameter and ν is an arbitrary measure defined on $((\mathbb{R}^d)^{\ell+1}, \mathcal{B}((\mathbb{R}^d)^{\ell+1}))$.

Link with Schrödinger Bridge. We first recall the relationship between Schrödinger Bridge and EOT. Given $T > 0$, $\mathbb{Q}$ a (reference) path measure, *i.e.*, $\mathbb{Q} \in \mathscr{P}(\mathrm{C}([0, T], \mathbb{R}^d))$ and two measures $\mu_0, \mu_1 \in \mathscr{P}(\mathbb{R}^d)$, solving the SB problem amounts to finding the path measure $\mathbb{P}^\star$ defined by

$$\mathbb{P}^\star = \arg\min \{ \mathrm{KL}(\mathbb{P}|\mathbb{Q}) \ :\ \mathbb{P} \in \mathscr{P}(\mathrm{C}([0, T], \mathbb{R}^d)),\ \mathbb{P}_0 = \mu_0,\ \mathbb{P}_T = \mu_1 \}\ . \tag{SB}$$

If $\mathbb{Q}$ is associated with a Stochastic Differential Equation (SDE)[1], of the form $\mathrm{d}\mathbf{X}_t = -a\mathbf{X}_t\mathrm{d}t + \mathrm{d}\mathbf{B}_t$, with $a \geq 0$, then it can be shown, see [Léonard, 2014b, Proposition 1] that $\mathbb{P}^\star_{0,T}$ verifies

$$\mathbb{P}^\star_{0,T} = \arg\min\{\mathrm{KL}(\pi|\mathbb{Q}_{0,T}) \,:\, \pi \in \mathscr{P}^{(2)},\ \pi_0 = \mu_0,\ \pi_1 = \mu_1\}\,. \tag{static-SB}$$

This is called the *static* formulation of SB. It can be shown that solving (static-SB) is equivalent to solving EOT with quadratic cost and regularization $\varepsilon = 2\sinh(aT)/a$ if $a > 0$, $\varepsilon = 2T$ if $a = 0$. Moreover, since $\mathbb{P}^\star = \mathbb{P}^\star_{0,T} \otimes \mathbb{Q}_{|0,T}$, where $\mathbb{Q}_{|0,T}$ is the measure $\mathbb{Q}$ conditioned on initial and terminal conditions, solving the *dynamic* problem (SB) is equivalent to solving (static-SB).

Similarly, (EmOT) can be easily rewritten in a *static* multi-marginal SB fashion

$$\pi^\star = \arg\min\{\mathrm{KL}(\pi|\pi^0) \,:\, \pi \in \mathscr{P}^{(\ell+1)},\ \pi_i = \mu_i\ , \forall i \in \mathsf{S}\}\ , \tag{mSB-like}$$

with $(\mathrm{d}\pi^0/\mathrm{dLeb})(x_{0:\ell}) \propto \exp[-c(x_{0:\ell})/\varepsilon](\mathrm{d}\nu/\mathrm{dLeb})(x_{0:\ell})$, where π^0 is the *reference* measure.

Diffusion Schrödinger Bridge. Recently, Bortoli et al. [2021] introduced Diffusion Schrödinger Bridge (DSB), a numerical scheme to solve (SB). It approximates the iterates of a *dynamic* version of the *Iterative Proportional Fitting* (IPF) scheme [Sinkhorn and Knopp, 1967, Knight, 2008, Peyré et al., 2019, Cuturi and Doucet, 2014], which can be described as follows: consider a sequence of path measures $(\mathbb{P}^n)_{n\in\mathbb{N}}$ such that $\mathbb{P}^0 = \mathbb{Q}$ and for any $n \in \mathbb{N}$

$$\mathbb{P}^{2n+1} = \arg\min\{\mathrm{KL}(\mathbb{P}|\mathbb{P}^{2n}) \,:\, \mathbb{P}_T = \mu_1\}, \qquad \mathbb{P}^{2n+2} = \arg\min\{\mathrm{KL}(\mathbb{P}|\mathbb{P}^{2n+1}) \,:\, \mathbb{P}_0 = \mu_0\}\,. \tag{7.1}$$

This procedure alternatively projects between the measures with fixed initial distribution and the ones with fixed terminal distribution. For the first iteration, we get that $\mathbb{P}^1 = \mu_1 \otimes \mathbb{Q}_{|T}$. Assuming that $\mathbb{Q}$ is given by $\mathrm{d}\mathbf{X}_t = f_t(\mathbf{X}_t)\mathrm{d}t + \mathrm{d}\mathbf{B}_t$, with $f : [0,T]\times\mathbb{R}^d \to \mathbb{R}^d$, then $\mathbb{P}^1$ is associated with the *time-reversal* of this SDE initialized at μ_1. The time-reversal of an SDE has been derived under mild assumptions on the drift and diffusion coefficients [Haussmann and Pardoux, 1986, Cattiaux et al., 2023]. In this case, we have $(\mathbf{Y}_{T-t})_{t\in[0,T]} \sim \mathbb{P}^1$, with $\mathbf{Y}_0 \sim \mu_1$ and

$$\mathrm{d}\mathbf{Y}_t = \{-f_{T-t}(\mathbf{Y}_t) + \nabla \log p_{T-t}(\mathbf{Y}_t)\}\mathrm{d}t + \mathrm{d}\mathbf{B}_t, \tag{7.2}$$

where p_t is the density of $\mathbb{P}^0_t$ w.r.t. the Lebesgue measure. The score $\nabla \log p_t$ is estimated using score matching techniques [Hyvärinen, 2005, Vincent, 2011]. The first iterate of DSB, $\mathbb{P}^1$, corresponds to a *denoising diffusion model* [Ho et al., 2020, Song et al., 2021b]. DSB iterates further and not only parameterizes the backward process but also the forward process. It can therefore be seen as a refinement of diffusion models drawing a bridge between generative modeling and optimal transport.

Tree-based framework. Consider an undirected tree $\mathsf{T} = (\mathsf{V}, \mathsf{E})$, with vertices V and edges E, such that V is identified with $\{0, \dots, \ell\}$. Inspired by Haasler et al. [2021], we restrict our study of (EmOT), to the case where the cost function c is the tree-structured *quadratic* cost derived from T

$$c(x_{0:\ell}) = \textstyle\sum_{\{v,v'\}\in\mathsf{E}} w_{v,v'}\|x_v - x_{v'}\|_2^2\ , \tag{7.3}$$

where $w_{v,v'}$ is a weight on the edge $\{v, v'\}$, which links v to v' (and v' to v). Furthermore, as in Haasler et al. [2021], we choose S, *i.e.*, the set of vertices of T with constrained marginals, to coincide with the *leaves* of T. This framework recovers important applications, from Wasserstein barycenters to Wasserstein propagation, see Solomon et al. [2014, 2015]. We emphasize that it differs from an OT problem defined on the space of graphs [Chen et al., 2016]. Here, each node represents a probability measure (observed or to be inferred) and each edge represents a coupling between two distributions.

We consider an arbitrary vertex $r \in \mathsf{V}$ and choose ν in (EmOT) such that $(\mathrm{d}\nu/\mathrm{dLeb})(x_{0:\ell}) = \varphi_r(x_r)$, where φ_r is a density defined on $\mathbb{R}^d$. Due to the form of ν and c, the reference measure π^0 in (mSB-like) is therefore a *probability* distribution which factorizes along $\mathsf{T}_r = (\mathsf{V}, \mathsf{E}_r)$, the directed version of T rooted in r. We refer to Appendix F.2 for more details on the notion of directed trees. In this setting, (EmOT) is equivalent to the tree-based problem

$$\pi^\star = \arg\min\{\mathrm{KL}(\pi|\pi^0) \,:\, \pi \in \mathscr{P}^{(|\mathsf{V}|)},\ \pi_i = \mu_i\ , \forall i \in \mathsf{S}\}\ , \tag{TreeSB}$$

$$\text{with} \quad \pi^0 = \pi^0_r \bigotimes_{(v,v')\in\mathsf{E}_r} \pi^0_{v'|v}\ , \tag{7.4}$$

[1] We refer to Appendix F.3 for details on solutions of SDEs and associated measures.

where $\pi^0_{v'|v}(\cdot \mid x_v) = \mathrm{N}(x_v, \varepsilon/(2w_{v,v'})\mathrm{I}_d)$ and $\pi^0_r \ll \mathrm{Leb}$ with density φ_r. In a manner akin to Haasler et al. [2021], we thus establish, in *continuous* state-space, the correspondence between (TreeSB), a *static* tree-based version of SB, and a version of EmOT with tree-structured cost (7.3). In our work, we make the following assumption on the constrained marginals $\{\mu_i\}_{i\in\mathsf{S}}$.

A 5. *For any* $i \in \mathsf{S}$, $\mu_i \ll \mathrm{Leb}$ *and* $\mathrm{H}(\mu_i) < \infty$.

In what follows, we define K as the number of leaves of T , denoting $\mathsf{S} = \{i_0, \dots, i_{K-1}\}$, and define the horizon times $T_{v,v'} = \varepsilon/(2w_{v,v'})$ for any $\{v, v'\} \in \mathsf{E}$. For any $i_k \in \mathsf{S}$, we will denote by $\mathsf{T}_k = (\mathsf{V}, \mathsf{E}_k)$ the directed version of T rooted in the leaf i_k. In the next section, we present our *dynamic* method to solve (TreeSB), called *Tree-based Diffusion Schrödinger Bridge*.

7.3 Tree-based Diffusion Schrödinger Bridge

In this section, we present a method to solve (TreeSB) in the case where $r \in \mathsf{S}$, *i.e.*, r is a leaf of T. We refer to Appendix F.5 for the extension to the case where $r \in \mathsf{V}\backslash\mathsf{S}$. Without loss of generality, see Appendix F.5, we assume that $r = i_{K-1}$ and choose $\varphi_r = \mathrm{d}\mu_{i_{K-1}}/\mathrm{dLeb}$, such that $\pi^0_{i_{K-1}} = \mu_{i_{K-1}}$.

Dynamic approach to mIPF. In order to approximate solutions of (TreeSB), we consider the *multi-marginal* extension of the IPF algorithm, denoted by mIPF. Namely, we define a sequence of probability distributions $(\pi^n)_{n\in\mathbb{N}}$ such that for any $n \in \mathbb{N}$

$$\pi^{n+1} = \arg\min\{\mathrm{KL}(\pi|\pi^n) \,:\, \pi \in \mathscr{P}^{(|\mathsf{V}|)},\ \pi_{i_{k_n+1}} = \mu_{i_{k_n+1}}\}\,, \tag{mIPF}$$

where $k_n = (n-1) \bmod(K)$ and $(k_n + 1)$ is identified with $n \bmod(K)$. We define a *mIPF cycle* as a sequence of K consecutive mIPF updates. In particular, each marginal constraint is considered exactly once during one mIPF cycle. In a practical setting, our main aim is to sample from the (mIPF) iterates at the lowest cost. Although these updates can be made explicit, see Marino and Gerolin [2020] for instance, direct sampling is unfeasible in practice when d is large. To overcome this limitation, we suggest to compute these iterates in a *dynamic* fashion with equivalent path measures.

Since π^0 factorizes along T, see (7.4), one can show that the iterates of (mIPF) also factorize along T, see Section 7.4. Since these iterates all have a constrained marginal, we obtain the following decomposition for any $n \in \mathbb{N}$: $\pi^n = \mu_{i_{k_n}} \otimes_{(v,v')\in\mathsf{E}_{k_n}} \pi^n_{v'|v}$ where E_{k_n} denotes the set of edges of the directed tree T_{k_n}. Then, our approach consists in computing *dynamic* iterates, *i.e.*, path measures, along the edges of T that coincide on their extremal times with the *static* iterates $(\pi^n)_{n\in\mathbb{N}}$. Namely, for any $n \in \mathbb{N}$, for any edge $(v, v') \in \mathsf{E}_{k_n}$, we define a path measure $\mathbb{P}^n_{(v,v')} \in \mathscr{P}(\mathrm{C}([0, T_{v,v'}], \mathbb{R}^d))$ such that $\mathrm{Ext}(\mathbb{P}^n_{(v,v')}) = \pi^n_{v,v'}$, where $\mathrm{Ext}(\mathbb{P}^n_{(v,v')})$ stands for the joint distribution of $\mathbb{P}^n_{(v,v')}$ at times 0 and $T_{v,v'}$. In particular, it comes that $\pi^n_{v'|v} = \mathbb{P}^n_{(v,v'),T_{v,v'}|0}$. Using the tree-based form of the (mIPF) iterates, we can thus sample from π^n by (i) following the directed edges of T_{k_n}, (ii) diffusing along them the corresponding path measures $(\mathbb{P}^n_{(v,v')})_{(v,v')\in\mathsf{E}_{k_n}}$ and (iii) picking the samples on the vertices. When T is a *bridge-shaped* tree (2 vertices, 1 edge), it simply reduces to the dynamic reformulation of the IPF scheme. In what follows, we explain how to obtain our *dynamic* sequence.

Definition of the dynamic iterates. We first compute the iterate $\mathbb{P}^0$, corresponding to the dynamic version of π^0 defined (7.4), in Proposition 11. Then, we build the following iterates by recursion on $n \in \mathbb{N}$ and prove their well-posedness in Proposition 12.

Proposition 11. *Let* $\mathsf{T}_{K-1} = (\mathsf{V}, \mathsf{E}_{K-1})$*, the directed tree associated with* $\mathsf{T} = (\mathsf{V}, \mathsf{E})$ *and root* i_{K-1}*. Then, for any* $(v, v') \in \mathsf{E}_{K-1}$*, there exists* $\mathbb{P}^0_{(v,v')} \in \mathscr{P}(\mathrm{C}([0, T_{v,v'}], \mathbb{R}^d))$ *with* $\mathrm{Ext}(\mathbb{P}^0_{(v,v')}) = \pi^0_{(v,v')}$ *and such that* $\mathbb{P}^0_{(v,v')|0}$ *is the distribution of* $(\mathbf{B}_t)_{t\in[0,T_{v,v'}]}$*, recalling that* $T_{v,v'} = \varepsilon/(2w_{v,v'})$.

Before deriving the dynamic counterpart of the (mIPF) iterates, we introduce several definitions. For any path measure $\mathbb{P}$, we denote by $\mathbb{P}^R$ the *time-reversal* of $\mathbb{P}$. For any directed tree and any vertex v of this tree, $p(v)$ refers to the (unique) *parent* of v, and $c(v)$ to the unique *child* of v when it exists, see Appendix F.2 for more details.

Let $n \in \mathbb{N}$. Assume that we have defined the sequence of our dynamic iterates $(\mathbb{P}^m_{(v,v')})_{(v,v')\in\mathsf{E}_{k_m}, m\le n}$ up to stage n.

Consider the path $\mathsf{P}_n = \{(v_j, v_{j+1})\}_{j=1}^J$ in the directed tree T_{k_n} such that $v_1 = i_{k_n}$ and $v_{J+1} = i_{k_n+1}$.

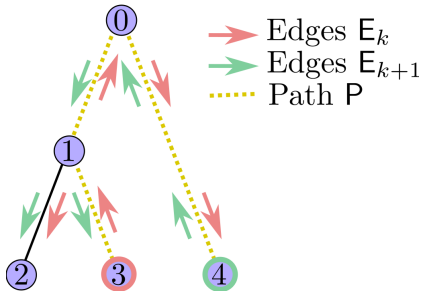


Figure 7.1: Illustration of the change of root in a toy tree with 5 vertices.

In particular, for any $(v, v') \in \mathsf{E}_{k_n+1}$, either $(v', v) \in \mathsf{P}_n$ or $(v, v') \in \mathsf{E}_{k_n} \backslash \mathsf{P}_n$. This is illustrated in Figure 7.1 when $\mathsf{V} = \{0, 1, 2, 3, 4\}$, $S = \{2, 3, 4\}$, $i_k = 3$ and $i_{k+1} = 4$: in this case, $\mathsf{P} = \{(3, 1), (1, 0), (0, 4)\}$ and $(1, 2)$ is the only edge common to E_k and E_{k+1}.

Consider now the directed tree $\mathsf{T}_{k_{n+1}}$. We define the $(n+1)$-th iterate of our dynamic sequence by recursion on the edges of this tree, following the breadth-first order. In this order, $(i_{k_n+1}, c(i_{k_n+1})) = (v_{J+1}, v_J)$ is the first edge considered.

First, we define $\mathbb{P}^{n+1}_{(v_{J+1}, v_J)} = \mu_{i_{k_n+1}} \otimes (\mathbb{P}^n_{(v_J, v_{J+1})})^R_{|0}$. In the case of a *bridge-shaped* tree, this is exactly the $(n+1)$-th update described in DSB. Then, for any $(v, v') \in \mathsf{E}_{k_n+1} \backslash \{(v_{J+1}, v_J)\}$,
(a) either $(v, v') \in \mathsf{E}_{k_n} \backslash \mathsf{P}_n$, and we define $\mathbb{P}^{n+1}_{(v,v')} = \mathbb{P}^{n+1}_{(p(v),v), T_{p(v),v}} \otimes \mathbb{P}^n_{(v,v')|0}$,
(b) or $(v', v) \in \mathsf{P}_n$, and we define $\mathbb{P}^{n+1}_{(v,v')} = \mathbb{P}^{n+1}_{(p(v),v), T_{p(v),v}} \otimes (\mathbb{P}^n_{(v',v)})^R_{|0}$.

Proposition 12. *Consider the sequence of dynamic iterates defined by (a) and (b). Then, for any $n \in \mathbb{N}$ and any $(v, v') \in \mathsf{E}_{k_n}$, $\mathbb{P}^n_{(v,v')} \in \mathscr{P}(\mathrm{C}([0, T_{v,v'}], \mathbb{R}^d))$ and we have $\mathrm{Ext}(\mathbb{P}^n_{(v,v')}) = \pi^n_{(v,v')}$.*

Proposition 12 highlights the equivalence between the (mIPF) iterates and our dynamic iterates. These path measures are defined iteratively, by following the updates (a) and (b) along the edges of T. The key observation here is that the computation of each dynamic iterate reduces to a sequence of updates (b) on a *path* linking two leaves of T. We emphasize that our iterates could be similarly obtained by directly considering a dynamic formulation of (TreeSB) and introducing the formalism of deterministic time branching processes. We leave the study of this problem for future work. We now get into the details of our practical implementation, which relies on score-based methods.

Approximation of the dynamic iterates. The time-reversal operated in the update (b) can be computed explicitly, see Haussmann and Pardoux [1986] for instance. Indeed, assuming that $\mathbb{P}^n_{(v',v)}$ is associated with $\mathrm{d}\mathbf{X}_t = f_{t,v',v}(\mathbf{X}_t)\mathrm{d}t + \mathrm{d}\mathbf{B}_t$ with $\mathbf{X}_0 \sim \pi^n_{v'}$, then, under mild conditions, its time-reversal $(\mathbb{P}^n_{(v',v)})^R$ is associated with $\mathrm{d}\mathbf{Y}_t = \{-f_{T-t,v',v} + \nabla \log p_{v',v,T-t}\}(\mathbf{Y}_t)\mathrm{d}t + \mathrm{d}\mathbf{B}_t$ with $\mathbf{Y}_0 \sim \pi^{n+1}_v$, where $p_{v',v,t}$ is the density of $\mathbb{P}^n_{(v',v),t}$ w.r.t. the Lebesgue measure. The score $\nabla \log p_{v',v,T-t}$ can then be approximated using score-matching techniques [Hyvärinen, 2005, Vincent, 2011] which are now ubiquitous in diffusion models [Song et al., 2021b] and used in DSB Bortoli et al. [2021]. Therefore, at iteration $(n+1)$, the update (b) is similar to the one of DSB *for each edge* on the path joining i_{k_n} and i_{k_n+1}. In practice, we parameterize the drifts $f_{t,v,v'}$ for any $\{v, v'\} \in \mathsf{E}$ with neural networks $f_{t,\theta_{v,v'}}$ and use the *mean-matching* loss introduced by Bortoli et al. [2021]. Note that doing so, we obtain $2|\mathsf{E}|$ neural networks. The whole procedure consisting in computing our dynamic iterates using the DSB framework is called *Tree-based Diffusion Schrödinger Bridge* (TreeDSB) and is summarized in Algorithm 3.

Algorithm 3 TreeDSB (Training)

Input: $\mathsf{T} = (\mathsf{V}, \mathsf{E})$, $\{\mu_i\}_{i \in \mathsf{S}}$, $\{\theta_{v,v'}\}_{\{v,v'\} \in \mathsf{E}}$, $N \in \mathbb{N}$
for $n = 0, \dots, N$ **do**
  Let $k_n = (n-1) \bmod (K)$
  Get path between i_{k_n} and i_{k_n+1}, $\mathsf{P}_n = \{v_j, v_{j+1}\}_{j=1}^J$
  while *not converged* **do**
    for $j = 1, \dots, J$ **do**
      Sample from $\mathbb{P}^n_{v_j, v_{j+1}}$ (Euler-Maruyama)
      Compute *mean matching* loss $\ell(\theta_{v_{j+1}, v_j})$
      $\theta_{v_{j+1}, v_j} \leftarrow$ Gradient Step$(\ell(\theta_{v_{j+1}, v_j}))$
      Update $f_{t, \theta_{v_{j+1}, v_j}}$
    end
  end
end
Output: $\{\theta_{v,v'}\}_{\{v,v'\} \in \mathsf{E}}$

The algorithm is initialized with $f_{t,\theta_{v,v'}} = 0$ for all $\{v, v'\} \in \mathsf{E}$. This corresponds to Brownian motion dynamics when sampling at the first iteration of TreeDSB, see Proposition 11. Note that in Algorithm 3, when we sample from $\mathbb{P}^n_{(v_j, v_{j+1})}$, we update $f_{t,\theta_{v_{j+1},v_j}}$ which will be used to sample from $\mathbb{P}^{n+1}_{(v_{j+1}, v_j)}$ in the next iterations. In order to sample from the dynamics $\mathbb{P}^n_{(v_j, v_{j+1})}$, we consider its Euler–Maruyama discretization, see Appendix F.6 for more details. We describe the different steps of the algorithm in the case of a toy example below, see Figure 7.2 for an illustration.

TreeDSB on a toy tree. We consider a star-shaped tree with three leaves denoted $\{1, 2, 3\}$ and its central node $\{0\}$. Following (7.4), we define π^0 with $r = 3$ and $\varphi_r = (\mathrm{d}\mu_3/\mathrm{dLeb})$. During the first iteration of TreeDSB, T is rooted at vertex 3 and we compute samples from the *forward* path $\mathsf{P}_0 = \{(3, 0), (0, 1)\}$ with Brownian motions, see Proposition 11, in order to learn the *backward* path $\{(1, 0), (0, 3)\}$.

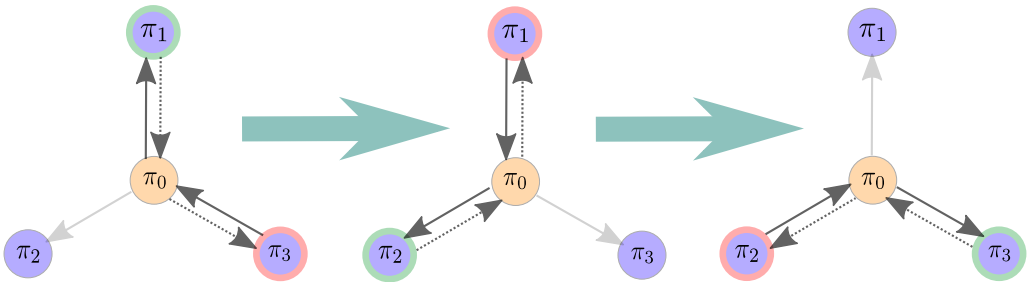


Figure 7.2: Illustration of one mIPF cycle solved by TreeDSB for a toy star-shaped tree. At each iteration, our method learns the *backward* stochastic process (dotted arrows) that goes from the target leaf (green-circled), corresponding to the constrained marginal, to the current root of the tree (red-circled) by using samples from the *forward* stochastic process (solid arrows).

In the next iteration, we re-root the tree T at vertex 1 and consider the *forward* path $\mathsf{P}_1 = \{(1,0),(0,2)\}$, where the edges $(1,0)$ and $(0,2)$ are respectively given by the first iteration and the initialisation. This highlights that *TreeDSB does not require to update the whole tree.* The following iterations are done similarly. At each iteration $n \in \mathbb{N}$, we sample from π^n by first sampling from μ_{k_n} at leaf i_{k_n} and then following the parameterized SDEs on the directed edges of T_{k_n}.

7.4 Theoretical properties of mIPF

In this section, we study some of the theoretical properties of the *static* iterates $(\pi^n)_{n\in\mathbb{N}}$, that are equivalent to our *dynamic* iterates according to Proposition 12. In the case where the cost function c is bounded in (EmOT), results of convergence of (mIPF) exist [Marino and Gerolin, 2020, Carlier, 2022]. However, our setting does not satisfy their assumptions, since our transport cost is quadratic and the measures are defined on $\mathbb{R}^d$. In what follows, we provide the first non-quantitative convergence results for (mIPF) in a *non-compact* setting.

For the rest of the section, we consider a static formulation of the multi-marginal Schrödinger bridge problem which is more general than (TreeSB), defined as

$$\pi^\star = \arg\min\{\mathrm{KL}(\pi|\pi^0) \,:\, \pi \in \mathscr{P}^{(\ell+1)},\ \pi_i = \mu_i\ , \forall i \in \mathsf{S}\}\ , \tag{static-mSB}$$

where $\mathsf{S} \subset \{0,\dots,\ell\}$, $\pi^0 \in \mathscr{P}$, $\{\mu_i\}_{i\in\mathsf{S}} \in \mathscr{P}^{|\mathsf{S}|}$. We consider the following set of assumptions.

A6. *There exists a family of measures $\{\nu_i\}_{i\in\{0,\dots,\ell\}}$ defined on $(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$ such that $\pi^0 \ll \bigotimes_{i=0}^{\ell} \nu_i$ with density $h = \mathrm{d}\pi^0/(\mathrm{d}\bigotimes_{i=0}^{\ell}\nu_i)$ and $\mu_i \ll \nu_i$ with density $r_i = \mathrm{d}\mu_i/\mathrm{d}\nu_i$ for any $i \in \mathsf{S}$.*

A7. $\{\pi \in \mathscr{P}^{(\ell+1)} : \mathrm{KL}(\pi \mid \pi^0) < \infty,\ \pi_i = \mu_i,\ \forall i \in \mathsf{S}\} \neq \emptyset$.

A8. *There exists a family of probability measures $\{\tilde{\mu}_j\}_{j\in\{0,\dots,\ell\}\setminus\mathsf{S}}$ such that $\pi^0 \sim \tilde{\pi}^0$, where $\tilde{\pi}^0 = \bigotimes_{i\in\mathsf{S}} \mu_i \bigotimes_{j\in\{0,\dots,\ell\}\setminus\mathsf{S}} \tilde{\mu}_j$.*

In particular, (static-mSB) recovers (TreeSB) by considering $\nu_i = \mathrm{Leb}$ for any $i \in \{0,\dots,\ell\}$ and $h(x_{0:\ell}) = \varphi_r(x_r)\exp[-c(x_{0:\ell})/\varepsilon]$ in **A6**. We detail in Appendix F.4 how **A7** and **A8** can be met in (TreeSB). Under these assumptions, the multi-marginal Schrödinger Bridge exists.

Proposition 13. *Assume* **A6** *and* **A7**. *Then, there exists a unique solution $\pi^\star$ to* (static-mSB). *In addition, assume* **A8**. *Then, there exists a family $\{\psi_i^\star\}_{i\in\mathsf{S}}$ of measurable functions $\psi_i^\star : \mathbb{R}^d \to \mathbb{R}$ such that*

$$(\mathrm{d}\pi^\star/\mathrm{d}\pi^0) = \exp[\textstyle\bigoplus_{i\in\mathsf{S}} \psi_i^\star] \quad \pi^0\text{-a.s.} \tag{7.5}$$

In order to establish the existence and uniqueness result of Proposition 13, we extend results from Nutz [2021] to the multi-marginal setting. A consequence of Proposition 13 is that the iterates of (mIPF) can be described using potentials.

Corollary 14. *Assume* **A6**, **A7** *and* **A8**. *Let $(\pi^n)_{n\in\mathbb{N}}$ be the sequence given by* (mIPF). *Then, for any $n \in \mathbb{N}^*$ with $k_n = (n-1) \bmod (K)$ and $q_n \in \mathbb{N}$ such that $n = q_n K + k_n + 1$, there exists a family of measurable functions $\{\psi_{i_0}^{q_n+1}, \dots, \psi_{i_{k_n}}^{q_n+1}, \psi_{i_{k_n+1}}^{q_n}, \dots, \psi_{i_{K-1}}^{q_n}\}$ such that*

$$(\mathrm{d}\pi^n/\mathrm{d}\pi^0)(x_{0:\ell}) = \exp[\textstyle\bigoplus_{j=0}^{k_n} \psi_{i_j}^{q_n+1}(x_{i_j}) \bigoplus_{j=k_n+1}^{K-1} \psi_{i_j}^{q_n}(x_{i_j})] \quad \pi^0\text{-a.s.} \tag{7.6}$$

In the tree-based setting, Corollary 14 explains why the (mIPF) iterations preserve the tree-based Markovian nature of π^0. We now prove that the marginal π_i^n converges to μ_i for any $i \in \mathsf{S}$, as n goes to infinity, *i.e.*, we have marginal convergence on the leaves of T.

Proposition 15. *Assume* **A**6 *and* **A**7. *Let* $(\pi^n)_{n\in\mathbb{N}}$ *be the sequence given by* (mIPF). *Then, we have* $\lim_{n\to\infty} \|\pi_i^n - \mu_i\|_{\mathrm{TV}} = 0$ *for any* $i \in \mathsf{S}$.

The previous result does not ensure the convergence of $(\pi^n)_{n\in\mathbb{N}}$ to the solution to (static-mSB). In particular, Proposition 15 does not provide the convergence of the marginals on the nodes $v \in \mathsf{V}\backslash\mathsf{S}$, which is key to compute regularized Wasserstein barycenters with TreeDSB. Relying on additional assumptions, we now derive the convergence of (mIPF).

A 9. $\bigoplus_{i\in\mathsf{S}} \mathrm{L}^1(\mu_i) \subset \mathrm{L}^1(\pi^\star)$ *is closed.*

A 10. *There exist* $\bar{c} \in (0,\infty)$ *such that* $\exp(\psi_{i_k}^n - \psi_{i_k}^{n+1}) \leq \bar{c}$, *for any* $n \in \mathbb{N}$, *any* $k \in \{0,\dots,K-2\}$.

These assumptions can be seen as multi-marginal extensions of the ones of Ruschendorf [1995], see Appendix F.4 for a discussion and examples.

Proposition 16. *Assume* **A**6, **A**7, **A**8, **A**9 *and* **A**10. *Let* $(\pi^n)_{n\in\mathbb{N}}$ *be the sequence given by* (mIPF). *Then, we have* $\lim_{n\to\infty} \|\pi^n - \pi^\star\|_{\mathrm{TV}} = 0$, *where* $\pi^\star$ *is given in Proposition 13.*

To the best of our knowledge, Proposition 16 is the first convergence result of (mIPF) without assuming that the space is compact or that the cost is bounded. We highlight that traditional techniques to prove the convergence of IPF cannot be easily extended to the multi-marginal setting as pointed by Carlier [2022]. In the case of bounded cost, quantitative results exist [Marino and Gerolin, 2020, Carlier, 2022]. We leave the study of such results in the *unbounded* cost setting for future work.

7.5 Application to Wasserstein barycenters

Although Algorithm 3 can be applied to trees T with fixed marginals on the leaves, one case of particular interest is star-shaped trees, *i.e.*, trees with a central node, denoted by index 0, and such that $\mathsf{S} = \{1,\dots,\ell\}$ (see Figure 7.2 for an illustration with $\ell = 3$). In this section, we draw a link between (TreeSB) and regularized Wasserstein barycenters. We recall the definition of the Wasserstein distance of order 2 with ε-entropic regularization between μ and ν [Peyré et al., 2019, Chapter 4]

$$W_{2,\varepsilon}^2(\mu,\nu) = \inf\{\textstyle\int \|x_1 - x_0\|^2 \mathrm{d}\pi(x_0,x_1) - \varepsilon \mathrm{H}(\pi) : \pi \in \mathscr{P}^{(2)}, \pi_0 = \mu, \pi_1 = \nu\}\,. \tag{7.7}$$

In this work, we consider the $(\ell\varepsilon, (\ell-1)\varepsilon)$-doubly-regularized Wasserstein-2 barycenter problem [Chizat, 2025] defined as follows

$$\mu_\varepsilon^\star = \arg\min\{\textstyle\sum_{i=1}^\ell w_i W_{2,\varepsilon/w_i}^2(\mu,\mu_i) + (\ell-1)\varepsilon \mathrm{H}(\mu) : \mu \in \mathscr{P}\}\,, \tag{regWB}$$

where $(w_i)_{i\in\{1,\dots,\ell\}} \in (0,+\infty)^\ell$. The following proposition shows the equivalence between the barycenter problem (regWB) and the multi-marginal Schrödinger bridge problem (TreeSB) over T. In particular, it allows us to use TreeDSB to estimate the solution $\mu_\varepsilon^\star$ of (regWB).

Proposition 17. *Let* $\varepsilon > 0$. *Assume* ***A**5. Also assume that* T *is a star-shaped tree with central node indexed by* 0, *and that the reference measure of* (TreeSB) *defined in* (7.4) *verifies* $r = i_{K-1}$ *and* $\varphi_r = \mathrm{d}\mu_{i_{K-1}}/\mathrm{dLeb} > 0$. *Under* **A**7, (regWB) *has a unique solution* $\pi_0^\star$, *where* $\pi^\star$ *solves* (TreeSB).

The proof of this result is postponed to Appendix F.4. More generally, we show in Appendix F.4 that, for any tree T, (TreeSB) is equivalent to a regularized version of the Wasserstein propagation problem [Solomon et al., 2014, 2015]. Moreover, we present in Appendix F.5 an extension of Proposition 17 in the case where the chosen root r is not a leaf of T. We finally emphasize that the formulation of (regWB) leads to a *minimization* of the entropy of the barycenter. In particular, this allows us to choose ε reasonably large in TreeDSB, which is a stability advantage compared to other regularized methods which do not consider this further regularization.

7.6 Related work

Diffusion Schrödinger Bridge. Schrödinger Bridges [Schrödinger, 1932] have been extensively studied using tools from stochastic control and probability theory [Léonard, 2014b, Dai Pra, 1991, Chen et al., 2021b]. More recently, algorithms were proposed to efficiently approximate such bridges in the context of machine learning.

In particular, Bortoli et al. [2021] proposed DSB while Vargas et al. [2021], Chen et al. [2022] developed related algorithms. In Chen et al. [2023b], the authors study a multi-marginal version of DSB in a linear tree-based setting, where the set of observed nodes is the whole set of vertices. However, contrary to our setting, Chen et al. [2023b] introduced a momentum variable. This allows for smoother trajectories which are desirable for single-cell trajectories applications and correspond to some spline interpolation in the space of probability measures [Chen et al., 2018b]. A general framework for tree-based static Schrödinger Bridges on discrete state-spaces was given in Haasler et al. [2021]. In this work, we extend their formulation to a dynamic and continuous setting, see Appendix F.4 for more a thorough comparison.

Wasserstein barycenters. The notion of Wasserstein barycenter was first introduced in Rabin et al. [2012] and then later studied in Agueh and Carlier [2011]. The algorithms to solve this problem can be split into two families: the in-sample based approaches and the parametric ones. In-sample approaches require access to all the measures μ_i which are assumed to be empirical measures [Cuturi and Doucet, 2014, Benamou et al., 2015, Solomon et al., 2015]. Related to this class of algorithms is the semi-discrete approach, which aims at computing a Wasserstein barycenter between continuous distribution but rely on a discretization of the barycenter [Claici et al., 2018, Staib et al., 2017, Mi et al., 2020]. Most recent approaches do not rely on a discrete representation of the barycenter, but instead parameterize it using neural networks. These approaches can be further split into two categories. First, *measure-based optimization* approaches parameterize the measures using a neural network. This is the case of Cohen et al. [2020], where the barycenter is given by a generative model, which is then optimized . Fan et al. [2021] introduce an optimization procedure which relies on a *min-max-min* problem using the framework of Makkuva et al. [2020]. More recently, Korotin et al. [2022] considered a fixed point-based algorithm introduced in Álvarez-Esteban et al. [2016] to update a generative model parametrizing the barycenter. On the one hand, *potential-based methods* rely on a dual formulation of the barycenter. Korotin et al. [2021] parameterized the dual potentials using Input Convex Neural Network and considered regularizing losses imposing conjugacy and congruency. On the other hand, Li et al. [2020] consider a dual version of the *regularized* Wasserstein barycenter problem contrary to other works. Our approach applied to start-shaped trees also approximates a *regularized* Wasserstein barycenter. However, contrary to Li et al. [2020], we do not consider a parameterization of the potentials in the *static* setting but instead, parameterize the drift of an associated *dynamic* formulation using Schrödinger bridges. To the best of our knowledge TreeDSB is the first approach leveraging DSB-like algorithms to compute Wasserstein barycenters.

7.7 Experiments

In our experiments[2], we illustrate the performance of TreeDSB to compute entropic regularized Wasserstein barycenters for various tasks . We choose to compare our method with state-of-the-art regularized algorithms: fast free-support Wasserstein barycenter (fsWB) [Cuturi and Doucet, 2014] , and continuous regularized Wasserstein barycenter (crWB) [Li et al., 2020]. In all of our settings, we consider a star-shaped tree with K leaves and edge weights that are equal to $1/K$, resulting in a sequential training procedure over $2K$ neural networks. The initial diffusion is always a Brownian motion parameterized as explained in Proposition 11. Hence, the time horizon on each edge is defined by $T = K\varepsilon/2$. The order of the leaves is randomly shuffled between the mIPF cycles. We consider 50 steps for the time discretization on $[0, T]$. We refer to Appendix F.7 for details on the choice of the schedule, the architecture of the neural networks and the settings of our experiments.

Synthetic two dimensional datasets. We first illustrate TreeDSB in a synthetic two dimensional setting. We consider three different datasets *Swiss-roll* (vertex 0, starting node r), *Circle* (vertex 2) and *Moons* (vertex 3) and compute their Wasserstein barycenter (vertex 1) by running TreeDSB for 50 mIPF cycles with $\varepsilon = 0.1$. In Figure 7.3, we show the estimated densities of the datasets on the leaves of the tree (we emphasize that the distributions plotted on each leaf are generated from the central barycenter measure). In Figure 7.4, we observe the consistency between the barycenters generated from the different leaves. In Appendix F.7, we present additional results for this setting.

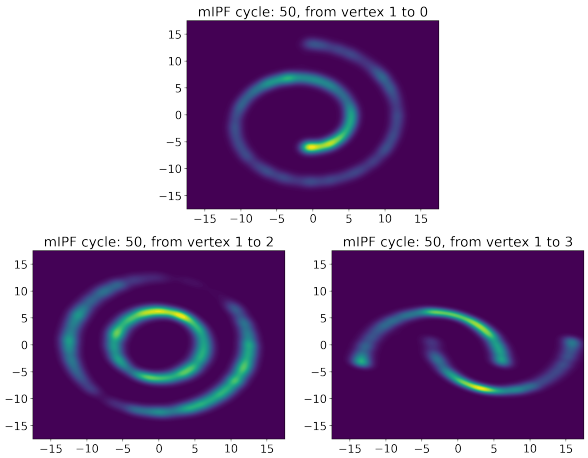


Figure 7.3: Estimated densities on the leaves.

[2]Code available at https://github.com/maxencenoble/tree-diffusion-schrodinger-bridge.

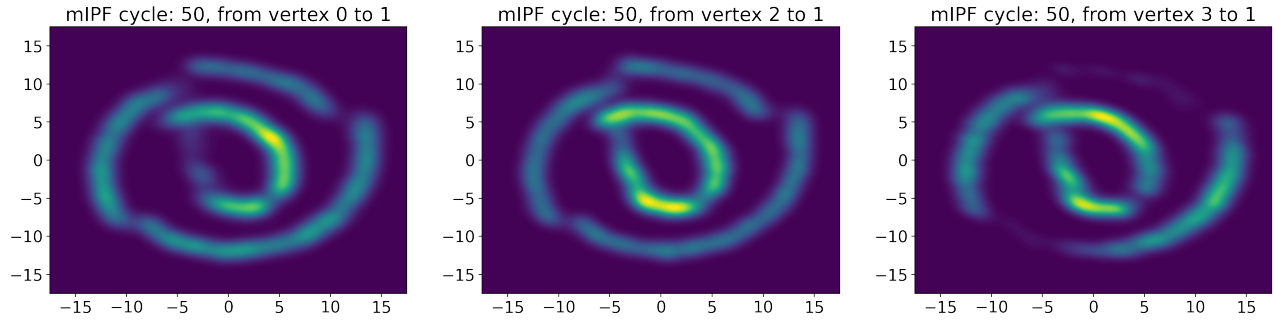


Figure 7.4: From left to right: barycenter estimated from the leaves *Swiss-roll*, *Circle* and *Moons*.

Method	$d = 2$	$d = 16$	$d = 64$	$d = 128$	$d = 256$
fsWB [Cuturi and Doucet, 2014]	$0.06_{\pm 0.01}$	$2.86_{\pm 0.06}$	$11.12_{\pm 0.06}$	$14.47_{\pm 0.07}$	$17.41_{\pm 0.05}$
crWB [Li et al., 2020]	$\mathbf{0.02_{\pm 0.01}}$	$\mathbf{1.52_{\pm 0.11}}$	$11.41_{\pm 0.73}$	$5.75_{\pm 0.02}$	$18.27_{\pm 0.54}$
Tree DSB	$0.63_{\pm 0.26}$	$\mathbf{1.07_{\pm 0.58}}$	$\mathbf{1.39_{\pm 0.07}}$	$\mathbf{1.92_{\pm 0.02}}$	$\mathbf{2.62_{\pm 0.07}}$

Table 7.1: Gaussian setting: comparison with the regularized methods crWB and fsWB.

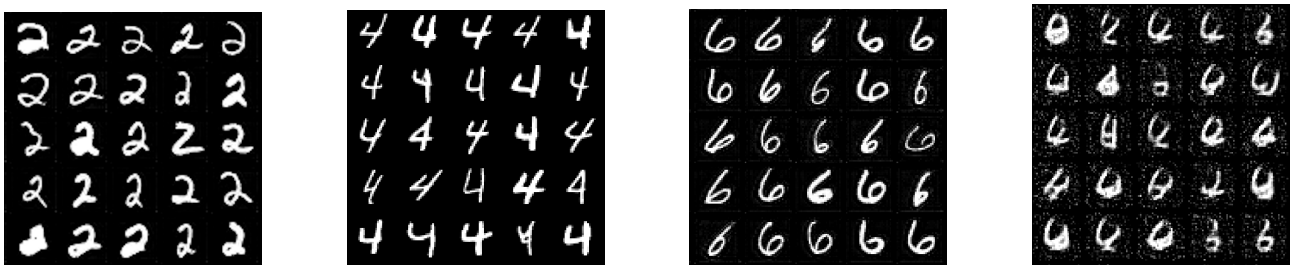


Figure 7.5: Samples from the estimated MNIST 2-4-6 marginals and from their Wasserstein barycenter.

Synthetic Gaussian datasets. Next, we consider three independent Gaussian distributions with zero mean and random non-diagonal covariance matrices whose conditional number is less than 10, following Fan et al. [2021]. In this case, the non-regularized barycenter can be exactly computed. To evaluate the performance of the algorithms, we use the Bures-Wasserstein Unexplained Variance Percentage (UVP), following [Korotin et al., 2021, Section 5]. Given a target distribution $\mu^\star \in \mathscr{P}$ and some approximation $\mu \in \mathscr{P}$, we define

$$\mathrm{BW}_2^2\text{-UVP}(\mu, \mu^\star) = 100 \cdot 2\,\mathrm{BW}_2^2(\mu, \mu^\star)/\,\mathrm{Var}(\mu^\star)\% \,, \tag{7.8}$$

where $\mathrm{BW}_2^2(\mu, \mu^\star) = W_2^2(\mathrm{N}(\mathbb{E}[\mu], \mathrm{Cov}(\mu)), \mathrm{N}(\mathbb{E}[\mu^\star], \mathrm{Cov}(\mu^\star))$.

In this setting, we choose $\mu^\star$ to be the non-regularized barycenter and assess the dependency w.r.t. the dimension of the algorithms using the BW_2^2-UVP metric. In Table 7.1, we compare ourselves with the two regularized methods Li et al. [2020] (L_2-reg. equal to 10^{-4}) and Cuturi and Doucet [2014]. We run TreeDSB for 10 mIPF cycles with $\varepsilon = 0.1$. Bold numbers represent the best values up to statistical significance. While Li et al. [2020] and Cuturi and Doucet [2014] enjoy better performance in low dimensions ($d = 2$), TreeDSB outperforms these methods as the dimension increases.

MNIST Wasserstein barycenter. We then turn to an image experiment using MNIST dataset [LeCun, 1998]. Here, an image is not considered as a 2D-dimensional distribution as in Cuturi and Doucet [2014] and Li et al. [2020], but as a sample from a high-dimensional probability measure ($d = 784$). We aim at computing a Wasserstein barycenter between the digits 2,4 and 6. To do so, we run TreeDSB for 10 mIPF cycles with r that corresponds to the digit 6 and $\varepsilon = 0.5$. In Figure 7.5, we display samples from the estimated marginals on the leaves, to assess the reconstruction of the digits 2, 4 and 6, and samples from the barycenter, obtained by diffusing from the leaf corresponding to the digit 6. Our results prove the scalability of TreeDSB to the high-dimensional setting, compared to state-of-the-art regularized methods. Additional results on MNIST dataset are given in Appendix F.7.

Subset posterior aggregation. Finally, we evaluate TreeDSB in the context of *Bayesian fusion* [Srivastava et al., 2018], also called posterior aggregation. Given a Bayesian model and a dataset partitioned into several shards, this task aims at recovering the full data posterior distribution from the posterior distributions computed on each shard.

Method	Without het.	With het.
fsWB [Cuturi and Doucet, 2014]	$12.95_{\pm 0.35}$	$14.43_{\pm 0.51}$
crWB [Li et al., 2020]	$20.66_{\pm 0.71}$	$23.06_{\pm 0.12}$
Tree DSB	$\mathbf{8.69_{\pm 0.12}}$	$\mathbf{8.90_{\pm 0.68}}$

Table 7.2: Bayesian fusion setting: comparison with the regularized methods crWB and fsWB.

In particular, it has been proved that the barycenter of the subdataset posteriors is close to the full data posterior under mild assumptions [Srivastava et al., 2018]. Here, we consider a logistic regression model applied to the `wine` dataset[3] ($d = 42$) and proceed as follows. We first split this dataset into 3 subsets, with or without heterogeneity, and estimate the posterior parameters on each shard. Then, we draw samples from the obtained logistic distributions to define μ_1, μ_2, μ_3. Then, we compute the Wasserstein barycenter of these measures, and compare it to the posterior computed on the full dataset. As in the synthetic Gaussian experiment, we run TreeDSB for 10 mIPF cycles $\varepsilon = 0.1$ and we compare ourselves with Li et al. [2020] (L_2-reg. equal to 10^{-4}) and Cuturi and Doucet [2014]. We evaluate the methods using the BW_2^2-UVP metric, where $\mu^\star$ is the estimated full data posterior, and report the results in Table 7.2. In both settings, we observe that our method outperforms existing regularized methods to compute Wasserstein barycenters.

Limitations. One of the main limitation of entropic regularized OT approach is that their behavior is usually badly conditioned as $\varepsilon \to 0$. In our setting, we observe that if ε, or equivalently T, is too low then the algorithm becomes less stable as the training of the models slows down. In the future, we plan to mitigate this issue by incorporating fixed point techniques like the one used in Korotin et al. [2022]. Finally, since our algorithm is based on DSB [Bortoli et al., 2021], it suffers from the same limitations. In particular, training different neural networks iteratively incurs some bias in the SDE which is harmful for large number of mIPF iterations.

7.8 Discussion

In this paper, we introduced Tree-based Diffusion Schrödinger Bridge (TreeDSB) a scalable scheme to approximate solutions of entropic-regularized multi-marginal Optimal Transport (mOT) problems. Our methodology leverages tools from the diffusion model literature and extends Diffusion Schrödinger Bridge [Bortoli et al., 2021]. In particular, it approximates the iterates of the multi-marginal Iterative Proportional Fitting (mIPF) algorithm, for which we prove its convergence under mild assumptions. We illustrate the efficiency of TreeDSB for image processing and Bayesian fusion, using the link between mOT and Wasserstein barycenters. In future work, we would like to study quantitative convergence bounds for mIPF in the *unbounded* cost setting. Another line of work would be to scale TreeDSB to higher dimensional problems building on recent developments in the diffusion model and flow matching community [Lipman et al., 2023, Peluchetti, 2023, Shi et al., 2023].

Acknowledgments

We thank James Thornton for the DSB codebase[4] and useful discussions. AD acknowledges support from the Lagrange Mathematics and Computing Research Center. AD and MN would like to thank the Isaac Newton Institute for Mathematical Sciences for support and hospitality during the programme *The mathematical and statistical foundation of future data-driven engineering* when work on this paper was undertaken. MN acknowledges funding from the grant SCAI (ANR-19-CHIA-0002).

[3] https://archive.ics.uci.edu/ml/datasets/wine
[4] https://github.com/JTT94/diffusion_schrodinger_bridge

Chapter 8

Twisted Schrödinger Bridge Matching : a unified continuous- and discrete-time framework

8.1 Introduction

The task of learning a transport map between two arbitrary probability distributions μ (the *source*) and ν (the *target*) is ubiquitous in modern machine learning, with prominent applications in generative modeling. In recent years, this research area has attracted significant attention through the development of efficient *bridge matching* algorithms [Albergo et al., 2023], which unify within a single framework widely used diffusion models [Song et al., 2021b] and flow matching models [Lipman et al., 2023], typically considering μ as a Gaussian distribution. These learning procedures operate dynamically, *i.e.*, the transport from μ to ν (and from ν to μ) is achieved through the simulation of a Markovian stochastic process defined on a time interval $[0, T]$, with $T > 0$.

Concretely, bridge matching methods proceed in two successive stages. First, they define a smooth density path $(p_t)_{t\in[0,T]}$ interpolating between $\mu = p_0$ and $\nu = p_T$. Second, they learn forward and backward dynamics whose marginals match the prescribed density path. When no constraint is imposed on the transport, the first stage is straightforward. Given access to a coupling $\pi_{0,T}$ between μ and ν, typically through training samples $(X_0, X_T) \sim \pi_{0,T}$, one can define a *deterministic* interpolant $(X_t)_{t\in[0,T]}$ as a linear combination of X_0 and X_T, with boundary conditions enforcing the correct marginals at $t = 0$ and $t = T$, and set $p_t = \mathcal{L}(X_t)$. Based on this interpolant, one may add a centered Gaussian noise term with a time-dependent variance that vanishes at $t = 0$ and $t = T$. In this case, $(X_t)_{t\in[0,T]}$ becomes a *stochastic* interpolant, and $p_t = \mathcal{L}(X_t)$ corresponds to a Gaussian convolution of $\pi_{0,T}$. In both settings, the conditional distribution of X_t given (X_0, X_T), denoted $p_{t|0,T}$, is tractable and can be sampled efficiently. Moreover, the second stage reduces to a tractable mean-square regression objective. This combination explains the success of bridge matching methods in large-scale settings.

Importantly, the bridge matching framework remains valid when the transport between μ and ν is constrained to be the Optimal Transport (OT) map associated with the quadratic Euclidean cost. For instance, Tong et al. [2023] show that if $\pi_{0,T}$ is the OT plan between μ and ν in the Kantorovich sense, and $p_{t|0,T}$ is chosen as a Dirac mass at the linear interpolation between X_0 and X_T, then the dynamics learned via bridge matching recover the dynamical OT formulation [Benamou and Brenier, 2000], *i.e.*, the induced flow coincides with the OT map. A similar result holds for entropy-regularized OT (EOT). Indeed, Tong et al. [2024] prove that if $\pi_{0,T}$ is the EOT plan between μ and ν, and $p_{t|0,T}$ corresponds to the Brownian bridge between X_0 and X_T, then the resulting dynamics solve the Schrödinger Bridge (SB) problem [Schrödinger, 1932], which is known to be the dynamical counterpart of the EOT problem, see, e.g., Bortoli et al. [2021].

In its most general form, the SB problem is formulated as a variational problem over path measures on $[0, T]$, aiming at finding $\mathbb{P}$ closest to a reference measure $\mathbb{Q}$, under the marginal constraints $\mathbb{P}_0 = \mu$ and $\mathbb{P}_T = \nu$. When connected to EOT, this problem is referred to as the *standard* or *canonical* SB problem, in which $\mathbb{Q}$ simply corresponds to Brownian motion. In this case, the SB problem admits a formulation as a Stochastic Optimal Control (SOC) problem with affine feedback control, enabling a rich modeling of population dynamics.

A fundamental result states that the SB solution $\mathbb{P}^\star$ is the unique path measure satisfying the endpoint marginal constraints, which is Markov and belongs to the reciprocal class of $\mathbb{Q}$, *i.e.*, $\mathbb{P}^\star_{|0,T} = \mathbb{Q}_{|0,T}$ [Léonard, 2014b]. This characterization has motivated the development of fixed-point algorithms to solve the standard SB problem, that alternate KL projections onto subsets of these constraints, using tools from generative modeling as computational backbones. Two main classes of methods can be distinguished. The first implements the Iterative Proportional Fitting (IPF) scheme [Fortet, 1940], enforcing marginal constraints, with DSB [Bortoli et al., 2021] as a representative method based on score matching. The second implements the Iterative Markovian Fitting (IMF) scheme, enforcing Markovian and reciprocal constraints (through Markovian and reciprocal projections, respectively) with DSBM [Shi et al., 2023] as a representative method relying on bridge matching. Compared to DSB, DSBM exhibits superior numerical performance, which has motivated substantial interest in its large-scale development [Bortoli et al., 2024b, Peluchetti, 2025].

In practice, however, the canonical SB formulation may be too restrictive for general trajectory inference problems. In many applications, additional information about the transport at intermediate times is available and should be incorporated into the model, see, e.g., Tamir et al. [2023]. A typical example arises when snapshot data are observed at a small number of intermediate time points, which is insufficient to treat them as prescribed marginals and therefore precludes the use of multi-marginal SB formulations [Chen et al., 2023b]. Incorporating such intermediate information has motivated recent work by Liu et al. [2024], who introduce a generalized version of the SB problem. This formulation remains a SOC problem with affine feedback control, but includes an additional time-dependent potential acting as a state cost. While the feedback term regularizes the trajectories, the potential may act either as a reward, attracting trajectories toward certain regions, or as a penalty, repelling them, thereby inducing a trade-off between the two effects. To solve this generalized SB problem, Liu et al. [2024] propose Generalized Schrödinger Bridge Matching (GSBM), an iterative bridge matching algorithm extending DSBM. Despite its strong empirical performance, it remains unclear how the proposed updates relate to the IMF scheme, which in principle applies to general SB-type problems. This work aims at filling this gap.

Contributions. In this paper, we focus on the SOC problem introduced by Liu et al. [2024] and make the following contributions:

- We first reformulate this problem through the concept of *Twisted Schrödinger Bridge* (TSB), an instance of the dynamical SB problem whose reference process is a twisted Brownian motion, *i.e.*, a Feynman-Kac transform of Brownian motion. Building on this formulation, we derive a faithful extension of the IMF scheme to this setting, which differs from the update rules proposed in Liu et al. [2024]. In particular, we introduce a novel Markovian projection loss that explicitly incorporates the gradient of the state cost, and we reformulate the Reciprocal projection as a tractable variational problem with respect to a twisted Brownian bridge. This computational derivation of the IMF algorithm corresponds to the continuous-time version of *Twisted Schrödinger Bridge Matching* (TSBM).
- Second, we introduce a discrete-time formulation of the TSB problem, *i.e.*, defined on a finite number of time steps over $[0, T]$, extending the discrete formulation proposed by Gushchin et al. [2024] for the canonical SB problem. This formulation is motivated by practical considerations, as implementations operate in discrete time, and allows us to reduce the bias induced by continuous-to-discrete translations. We derive an ideal iterative scheme in the spirit of IMF, alternating discrete-time Markovian and Reciprocal projections.
- Finally, we propose a practical implementation of the discrete-time scheme, corresponding to the discrete-time version of TSBM. We notably introduce a tractable discrete-time bridge matching loss for the Markovian projection loss. Departing from continuous-time approaches, we introduce an original methodology to compute the reciprocal projection by reformulating it as a sampling problem, for which we design an advanced Monte Carlo scheme. This strategy avoids the use of parameterized families at this stage, thereby reducing approximation error in principle.

Notation. For any measurable space $(\mathsf{X}, \mathcal{X})$, we denote by $\mathscr{P}(\mathsf{X})$ the space of probability measures defined on $(\mathsf{X}, \mathcal{X})$. Unless specified, $\mathcal{X}$ is defined as the Borel sets on X. We denote by $\mathbf{C}_T = \mathrm{C}([0, T], \mathbb{R}^d)$ the space of continuous functions from $[0, T]$ to $\mathbb{R}^d$, hence $\mathscr{P}(\mathbf{C}_T)$ is the set of continuous-time stochastic processes (*i.e.*, path measures) on $[0, T]$, and by $\mathrm{C}^{0,1}([0, T] \times \mathbb{R}^d, \mathbb{R})$ the space of potentials from $[0, T] \times \mathbb{R}^d$ to $\mathbb{R}$ that are continuous with respect to their time input and continuously differentiable with respect to their position input. We use the following notation for mixture of bridges: for any joint distribution $\pi_{0,T} \in \mathscr{P}(\mathbb{R}^d \times \mathbb{R}^d)$ and any path measure $\mathbb{Q} \in \mathscr{P}(\mathbf{C}_T)$, $\mathbb{P} = \pi_{0,T}\mathbb{Q}_{|0,T}$ is the path measure defined by $\mathbb{P}(\cdot) = \int_{\mathbb{R}^d \times \mathbb{R}^d} \mathbb{Q}_{|0,T}(\cdot|x_0, x_T)\mathrm{d}\pi_{0,T}(x_0, x_T)$.

We denote by $\mathcal{M}_T \subset \mathscr{P}(\mathbf{C}_T)$ the set of Markovian continuous-time stochastic processes. For any $N \geq 0$, we use the notation $\mathscr{P}^{(N+2)} = \mathscr{P}((\mathbb{R}^d)^{N+2})$, referring to set of discrete-time stochastic processes defined over $(N+2)$ timesteps (N intermediary timesteps taken between 0 and T). Assume that $\mathsf{X} = (\mathbb{R}^d)^{N+2}$. For any $x \in \mathsf{X}$ and any $m, n \in \{0, \dots, N+1\}^2$ such that $m \leq n$, let $x_{m:n} = (x_m, x_{m+1}, \dots, x_n)$. Let Leb be the Lebesgue measure. For any non-negative function $f : \mathsf{X} \to \mathbb{R}_+$, such that $\int_{\mathsf{X}} f \mathrm{dLeb} < +\infty$, we define $\mathrm{H}(f) = -\int_{\mathsf{X}} f \log f \mathrm{dLeb} \in (-\infty, +\infty]$. For any distribution $\mu \in \mathscr{P}(\mathsf{X})$, we define the entropy of μ as $\mathrm{H}(\mu) = \mathrm{H}(\mathrm{d}\mu/\mathrm{dLeb})$ if $\mu \ll \mathrm{Leb}$ and $\mathrm{H}(\mu) = +\infty$ otherwise. For any two arbitrary measures μ and ν defined on $(\mathsf{X}, \mathcal{X})$, we define the Kullback–Leibler divergence between μ and ν as $\mathrm{KL}(\mu|\nu) = \int_{\mathsf{X}} \log(\mathrm{d}\mu/\mathrm{d}\nu)\mathrm{d}\mu - \int_{\mathsf{X}} \mathrm{d}\mu + \int_{\mathsf{X}} \mathrm{d}\nu$ if $\mu \ll \nu$ and $\mathrm{KL}(\mu \mid \nu) = +\infty$ otherwise. Finally, we denote $\mathrm{N}(\mathbf{m}, \Sigma)$ the Gaussian distribution with mean $\mathbf{m}$ and covariance matrix Σ.

Throughout the whole paper, we consider a horizon time $T > 0$ and two probability distributions $\mu \in \mathscr{P}(\mathbb{R}^d)$ and $\nu \in \mathscr{P}(\mathbb{R}^d)$, to be the constrained marginals of the SB problem, respectively at times 0 and T. We also consider a scaling hyperparameter $\sigma > 0$ and denote by $\mathbb{Q}$ the scaled Brownian motion $(\sigma B_t)_{t \in [0,T]}$.

8.2 Background

Schrödinger Bridge problem. In its most general formulation, the Schrödinger Bridge (SB) problem consists in finding a path measure $\mathbb{P}^\star \in \mathscr{P}(\mathbf{C}_T)$ solving the constrained variational problem

$$\arg\min\{\mathrm{KL}(\mathbb{P}\|\mathbb{Q}^{\mathrm{ref}}) \ : \ \mathbb{P} \in \mathscr{P}(\mathbf{C}_T), \mathbb{P}_0 = \mu, \ \mathbb{P}_T = \nu\} \ . \tag{8.1}$$

Here, $\mathbb{Q}^{\mathrm{ref}} \in \mathcal{M}$ denotes a reference path measure. By applying the KL chain rule, one readily shows that this problem is equivalent to finding $\mathbb{P}^\star = \mathbb{P}^\star_{0,T}\mathbb{Q}^{\mathrm{ref}}_{|0,T}$, where $\mathbb{P}^\star_{0,T}$ solves the *static* SB problem

$$\arg\min\{\mathrm{KL}(\pi\|\mathbb{Q}^{\mathrm{ref}}_{0,T}) \ : \ \pi \in \mathscr{P}^{(2)}, \pi_0 = \mu, \ \pi_1 = \nu\} \ . \tag{8.2}$$

Under mild assumptions on $\mathbb{Q}^{\mathrm{ref}}$, μ, and ν, Léonard [2014b] established that $\mathbb{P}^\star$ is the unique Markov path measure satisfying $\mathbb{P}^\star = \mathbb{P}^\star_{0,T}\mathbb{Q}^{\mathrm{ref}}_{|0,T}$ together with the marginal constraints $\mathbb{P}^\star_0 = \mu$ and $\mathbb{P}^\star_T = \nu$. This characterization naturally motivates the Iterative Markovian Fitting (IMF) scheme [Shi et al., 2023], a fixed-point procedure inspired by iterative Bregman projections [Bregman, 1967], which we now briefly recall.

Iterative Markovian Fitting. Let $\mathbb{P} \in \mathscr{P}(\mathbf{C}_T)$. The *Reciprocal projection* of $\mathbb{P}$ with respect to $\mathbb{Q}^{\mathrm{ref}}$, denoted $\mathrm{proj}_{\mathcal{R}(\mathbb{Q}^{\mathrm{ref}})}(\mathbb{P}) \in \mathscr{P}(\mathbf{C}_T)$, is defined as the solution of

$$\arg\min\{\mathrm{KL}(\mathbb{P}\|\Pi) \ : \ \Pi \in \mathcal{R}(\mathbb{Q}^{\mathrm{ref}})\} \ , \tag{8.3}$$

where $\mathcal{R}(\mathbb{Q}^{\mathrm{ref}})$ denotes the reciprocal class of $\mathbb{Q}^{\mathrm{ref}}$, *i.e.*, the set of path measures of the form $\Pi = \Pi_{0,T}\mathbb{Q}^{\mathrm{ref}}_{|0,T}$. Importantly, this projection admits a closed-form expression via the KL chain rule:

$$\mathrm{proj}_{\mathcal{R}(\mathbb{Q}^{\mathrm{ref}})}(\mathbb{P}) = \mathbb{P}_{0,T}\mathbb{Q}^{\mathrm{ref}}_{|0,T} \ . \tag{8.4}$$

Similarly, the *Markovian projection* of $\mathbb{P}$, denoted $\mathrm{proj}_{\mathcal{M}_T}(\mathbb{P}) \in \mathscr{P}(\mathbf{C}_T)$, is defined as the solution of

$$\arg\min\{\mathrm{KL}(\mathbb{P}\|\mathbb{M}) \ : \ \mathbb{M} \in \mathcal{M}_T\} \ , \tag{8.5}$$

where $\mathcal{M}_T$ denotes the set of Markov path measures. Unlike the Reciprocal projection, this minimization does not admit an explicit closed-form solution in general.

The ideal IMF scheme alternates between these two KL projections, thereby enforcing both the Markov and reciprocal constraints satisfied by $\mathbb{P}^\star$. Starting from an arbitrary $\mathbb{P}^{-1} \in \mathscr{P}(\mathbf{C}_T)$ with correct marginals $\mathbb{P}^{-1}_0 = \mu$ and $\mathbb{P}^{-1}_T = \nu$, one defines recursively for $n \in \mathbb{N}$

$$\mathbb{P}^{2n} = \mathrm{proj}_{\mathcal{R}(\mathbb{Q}^{\mathrm{ref}})}(\mathbb{P}^{2n-1}) \ , \quad \mathbb{P}^{2n+1} = \mathrm{proj}_{\mathcal{M}_T}(\mathbb{P}^{2n}) \ . \tag{8.6}$$

By construction, the marginal constraints are preserved at each iteration. Recently, Silveri et al. [2025] provided evidence of exponential convergence of IMF toward $\mathbb{P}^\star$ under weak assumptions.

Application to the canonical SB problem. Historically, from its original formulation [Schrödinger, 1932] to modern developments [Bortoli et al., 2021], the SB problem (8.1) has primarily been studied with Brownian motion as the reference measure, *i.e.*, $\mathbb{Q}^{\text{ref}} = \mathbb{Q}$. In this setting, the conditional measure $\mathbb{Q}^{\text{ref}}_{|0,T}$ reduces to the Brownian bridge, which is fully tractable and efficiently simulable. The static SB problem (8.2) then becomes an entropy-regularized optimal transport problem with quadratic transport cost, defined by

$$\arg\min\{\textstyle\int_{\mathbb{R}^d\times\mathbb{R}^d} \|x_1 - x_0\|^2 \,\mathrm{d}\pi(x_0, x_1) - 2\sigma^2 T\,\mathrm{H}(\pi) \;:\; \pi \in \mathscr{P}^{(2)}, \pi_0 = \mu,\ \pi_1 = \nu\}\,. \tag{8.7}$$

Moreover, Léonard [2014b] showed in this case that the dynamical formulation (8.1) is equivalent to the affine stochastic optimal control (SOC) problem

$$\arg\min\{\textstyle\int_0^T \frac{1}{2}\mathbb{E}[\|u_t(\mathbf{X}^u_t)\|^2]\mathrm{d}t \;:\; u \in \mathbf{A}_T(\nu)\,,\quad \mathrm{d}\mathbf{X}^u_t = u_t(\mathbf{X}^u_t)\mathrm{d}t + \sigma\mathrm{d}\mathbf{B}_t\,,\ \mathbf{X}^u_0 \sim \mu\}\,, \tag{8.8}$$

where $\mathbf{A}_T(\nu) = \{u \in \mathbf{A}_T : \mathbf{X}^u_T \sim \nu\}$ is the subset of admissible control functions $u \in \mathbf{A}_T$ (see details in Appendix G.1) enforcing the terminal marginal constraint. Then, the optimal path measure $\mathbb{P}^\star$ is induced by $\mathbf{X}^{u^\star}$, where $u^\star$ solves (8.8) [Léonard, 2014b, Proposition 2.10]. In the limit $\sigma \to 0$, the entropy term in (8.7) vanishes, recovering classical quadratic OT, while (8.8) converges to the Benamou-Brenier problem [Benamou and Brenier, 2000], its dynamical counterpart.

Recently, Shi et al. [2023], Peluchetti [2023] concurrently proposed computational implementations of IMF as machine learning procedures; we refer to them collectively as DSBM [Shi et al., 2023]. In this framework, the Reciprocal projection remains explicit, while the Markovian projection (8.5) is restricted to Markov path measures induced by affine-controlled SDEs associated with $\mathbb{Q}$, with controls v^θ_t parameterized by $\theta \in \Theta$. For any $\Pi \in \mathcal{R}(\mathbb{Q})$, they show using Doob's h-transform [Rogers and Williams, 2000] that computing $\mathrm{proj}_{\mathcal{M}_T}(\Pi)$ within the restricted parametric Markov paths reduces to finding $\theta^\star_f$, the solution of the regression problem

$$\arg\min\{\textstyle\int_0^T \mathbb{E}_{\Pi_{t,T}}[\|\sigma^2\nabla_{\mathbf{X}_t}\log\mathbb{Q}_{T|t}(\mathbf{X}_T|\mathbf{X}_t) - v^{\theta_f}_t(\mathbf{X}_t)\|^2]\mathrm{d}t \;:\; \theta_f \in \Theta\}\,, \tag{8.9}$$

and taking the path measure induced by $\mathrm{d}\mathbf{X}_t = v^{\theta^\star_f}_t(\mathbf{X}_t)\mathrm{d}t + \sigma\mathrm{d}\mathbf{B}_t$, with $\mathbf{X}_0 \sim \mu$. This objective is fully tractable: (i) samples from $\Pi_{t,T}$ can be obtained from samples of $\Pi_{0,T}$ via the Brownian bridge, since $\Pi \in \mathcal{R}(\mathbb{Q})$, and (ii) the transition kernels of $\mathbb{Q}$ are explicit. Hence, IMF reduces to an iterative bridge matching procedure with Brownian Bridge as stochastic interpolant [Albergo et al., 2023]. They also consider the equivalent backward formulation

$$\arg\min\{\textstyle\int_0^T \mathbb{E}_{\Pi_{0,t}}[\|\sigma^2\nabla_{\mathbf{X}_t}\log\mathbb{Q}_{t|0}(\mathbf{X}_t|\mathbf{X}_0) - v^{\theta_b}_t(\mathbf{X}_t)\|^2]\mathrm{d}t \;:\; \theta_b \in \Theta\}\,, \tag{8.10}$$

defining $\mathrm{proj}_{\mathcal{M}_T}(\Pi)$ as the time-reversal of the path measure induced by $\mathrm{d}\mathbf{Y}_t = v^{\theta^\star_b}_{T-t}(\mathbf{Y}_t)\mathrm{d}t + \sigma\mathrm{d}\tilde{\mathbf{B}}_t$, with $\mathbf{Y}_0 \sim \nu$, where $\theta^\star_b$ solves (8.10). To mitigate potential bias arising from imperfect marginal matching in practice, Shi et al. [2023] alternate between forward loss (8.9) and backward loss (8.10) for Markovian projection steps, requiring two neural networks to parameterize the drifts in both directions.

8.3 Continuous-time Twisted Schrödinger Bridge

In this section, we introduce a particular instance of the SB problem (8.1), which we call the *Twisted Schrödinger Bridge*. We prove that it is strictly equivalent to the SOC formulation introduced by Liu et al. [2024], which we restate below. Building on this SB formulation, we make the associated IMF scheme explicit. This yields a novel iterative bridge matching procedure, implementable in practice, whose updates differ from those of GSBM. The proof of the propositions stated in this section are postponed to Appendix G.1.

8.3.1 Introduction to the Twisted Schrödinger Bridge

Let $V \in \mathrm{C}^{0,1}([0,T]\times\mathbb{R}^d, \mathbb{R})$ be an arbitrary time-dependent potential. We study the instance of (8.1) with reference measure $\mathbb{Q}^{\text{ref}} = \mathbb{Q}^V$, where $\mathbb{Q}^V$ is absolutely continuous with respect to $\mathbb{Q}$, with Radon–Nikodym derivative

$$\mathrm{d}\mathbb{Q}^V(x_{[0:T]}) = \tfrac{1}{Z_V}\exp\Big(-\textstyle\int_0^T \frac{V_t(x_t)}{\sigma^2}\mathrm{d}t\Big)\,\mathrm{d}\mathbb{Q}(x_{[0:T]})\,, \tag{8.11}$$

with normalizing constant $Z_V = \int \exp\Big(-\int_0^T \frac{V_t(x_t)}{\sigma^2}\mathrm{d}t\Big)\,\mathrm{d}\mathbb{Q}(x_{[0:T]})$. Such transforms of Brownian motion are referred to under various names in the probability literature, including Feynman–Kac penalization [Roynette

and Yor, 2009], general h-transform [Léonard, 2011], exponential tilting [Chetrite and Touchette, 2015], and exponential reweighting [Nüsken and Richter, 2021]. In this work, we say that $\mathbb{Q}^V$ is a *twisted* version of $\mathbb{Q}$, and we adopt the same terminology for the associated bridges, *i.e.*, $\mathbb{Q}^V_{|0,T}$ is a twisted version of $\mathbb{Q}_{|0,T}$. We refer to the resulting SB problem as the Twisted SB (TSB) problem.

Assuming that V is such that $Z_V < \infty$, one can show that $\mathbb{Q}^V \in \mathcal{M}_T$, with an associated SDE that can be written as an affine-controlled Brownian motion, although the corresponding drift is generally intractable for arbitrary V, see Appendix G.1. Therefore, using Girsanov's formula, solving the TSB problem is equivalent to solving the affine SOC problem introduced by Liu et al. [2024], namely

$$\arg\min\{\int_0^T \mathbb{E}[\tfrac{1}{2}\|u_t(\mathbf{X}^u_t)\|^2 + V_t(\mathbf{X}^u_t)]\mathrm{d}t \,:\, u \in \mathbf{A}_T(\nu)\,, \quad \mathrm{d}\mathbf{X}^u_t = u_t(\mathbf{X}^u_t)\mathrm{d}t + \sigma\mathrm{d}\mathbf{B}_t\,,\ \mathbf{X}^u_0 \sim \mu\}\,, \tag{8.12}$$

and, as in the canonical setting, the optimal TSB path measure $\mathbb{P}^\star$ is induced by $\mathbf{X}^{u^\star}$, where $u^\star$ solves (8.12). From a physical standpoint, (8.12) can be interpreted as a trajectory inference problem with marginal constraints at endpoints, where V represents a penalty to be minimized along the dynamics in addition to the quadratic control cost. In particular, the canonical setting is recovered when $V \equiv 0$, and the formulation can be solved in closed form when V is quadratic [Teter et al., 2024].

Having established the equivalence between the TSB problem and (8.12), a natural strategy to solve (8.12) is to derive the IMF scheme associated with the TSB formulation, extending the algorithm originally introduced for the canonical SB problem. To this end, we now explain how the reciprocal and Markovian projections can be handled computationally in this setting; we denote this practical translation of IMF by *Twisted Schrödinger Bridge Matching* (TSBM).

8.3.2 Computation of the Reciprocal projection

Let $\mathbb{P} \in \mathcal{M}_T$, and assume that we have access to samples from $\mathbb{P}_{0,T}$[1]. In principle, computing the Reciprocal projection presents no difficulty, since it is explicit and given by (8.4). However, a key difference arises compared to the canonical setting. Although we show in Appendix G.1 that the twisted bridge $\mathbb{Q}^V_{|0,T}$ is Markov and corresponds to a diffusion process, its drift is not tractable in general. This prevents efficient sampling from $\mathbb{Q}^V_{|0,T}$, in contrast to the Brownian bridge $\mathbb{Q}_{|0,T}$.

To overcome this limitation, we propose to approximate the twisted bridge $\mathbb{Q}^V_{|0,T}$ by solving a reverse Kullback–Leibler problem on path space

$$\arg\min_{\psi\in\Psi} \mathbb{E}_{\mathbb{P}_{0,T}}[\mathrm{KL}(\mathbb{Q}^\psi_{|0,T}\|\mathbb{Q}^V_{|0,T})]\,. \tag{8.13}$$

Here, $\{\mathbb{Q}^\psi_{|0,T}\}_{\psi\in\Psi}$ denotes a parametric family of Markov bridges defined as *affine-controlled versions of the Brownian motion*, with drift parameterized by $\psi \in \Psi$. This variational class is naturally motivated by the structural properties of $\mathbb{Q}^V_{|0,T}$, which is itself an affine-controlled Brownian motion; see Appendix G.1 for a detailed characterization. Importantly, this setting enables the derivation of a tractable and practically implementable optimization objective, as formalized in the following proposition.

Proposition 18. *Assume that the variational family $\{\mathbb{Q}^\psi_{|0,T}\}_{\psi\in\Psi}$ in (8.13) describes the pinned diffusion processes associated, for almost surely any $(x_0, x_T) \sim \mathbb{P}_{0,T}$, to forward-time and backward-time parametric SDEs of the form*

$$\mathrm{d}\mathbf{X}^{0,T}_t = v^{f,\psi}_t(\mathbf{X}^{0,T}_t|x_0,x_T)\mathrm{d}t + \sigma\mathrm{d}\mathbf{B}_t\,,\ \mathbf{X}^{0,T}_0 = x_0,\ \mathbf{X}^{0,T}_T = x_T\,, \tag{8.14}$$

$$\mathrm{d}\mathbf{Y}^{0,T}_t = v^{b,\psi}_{T-t}(\mathbf{Y}^{0,T}_t|x_0,x_T)\mathrm{d}t + \sigma\mathrm{d}\tilde{\mathbf{B}}_t\,,\ \mathbf{Y}^{0,T}_0 = x_T,\ \mathbf{Y}^{0,T}_T = x_0\,, \tag{8.15}$$

with $v^{f,\psi}, v^{b,\psi} \in \mathbf{A}_T$. Then, the minimizer of (8.13), denoted by $\psi^\star$, is equivalently the minimizer of the two following losses

$$\mathcal{L}^f_{\mathcal{R}}(\psi) = \int_0^T \mathbb{E}_{\Pi^\psi_{0,t,T}}\left[\frac{1}{2}\|v^{f,\psi}_t(\mathbf{X}_t|\mathbf{X}_0,\mathbf{X}_T) - \sigma^2\nabla_{\mathbf{X}_t}\log\mathbb{Q}_{T|t}(\mathbf{X}_T|\mathbf{X}_t)\|^2 + V_t(\mathbf{X}_t)\right]\mathrm{d}t\,, \tag{8.16}$$

$$\mathcal{L}^b_{\mathcal{R}}(\psi) = \int_0^T \mathbb{E}_{\Pi^\psi_{0,t,T}}\left[\frac{1}{2}\|v^{b,\psi}_t(\mathbf{X}_t|\mathbf{X}_0,\mathbf{X}_T) - \sigma^2\nabla_{\mathbf{X}_t}\log\mathbb{Q}_{t|0}(\mathbf{X}_t|\mathbf{X}_0)\|^2 + V_t(\mathbf{X}_t)\right]\mathrm{d}t\,, \tag{8.17}$$

where $\Pi^\psi_{0,t,T} = \mathbb{Q}^\psi_{t|0,T}\mathbb{P}_{0,T}$.

[1] In practice, this can be achieved by simulating the SDE associated with $\mathbb{P}$ in either the forward or backward direction, as in standard bridge matching approaches.

In particular, (8.16) and (8.17) can be interpreted as conditional SOC problems, respectively defined in forward and backward time direction. For practical implementation, each variational bridge $\mathbb{Q}^{\psi}_{|0,T}$ should satisfy three requirements: (1) it should be easy to sample from marginally, (2) it should support the reparameterization trick for efficient backpropagation, and (3) it should admit a tractable expression for the parameterized drifts $v^{f,\psi}$ and $v^{b,\psi}$. A natural choice satisfying these conditions is to set $\mathbb{Q}^{\psi}_{|0,T}$ as a stochastic interpolant [Albergo et al., 2023]. In this case, $\mathbb{Q}^{\psi}_{t|0,T}$ is Gaussian, enabling both efficient sampling and reparameterization, and the associated forward and backward drifts are tractable. We refer to Appendix G.1 for implementation details. To enrich the expressivity of the variational family, one may also consider mixtures of stochastic interpolants, as proposed by Du et al. [2024], also detailed in Appendix G.1.

8.3.3 Computation of the Markovian projection

In Proposition 19, we show that the Markovian projection step in the IMF scheme for the TSB problem reduces to solving, in either forward or backward direction, a mean-square regression problem that involves the gradient of the penalty term ∇V.

Proposition 19. *Let $\Pi \in \mathcal{R}(\mathbb{Q}^V)$. Define $\theta_f^\star$ and $\theta_b^\star$ as the minimizers over Θ of the respective following objectives*

$$\mathcal{L}^f_{\mathcal{M}_T}(\theta_f) = \int_0^T (T-t)^{-1} \int_t^T \mathbb{E}_{\Pi_{t,s,T}} \left[\left\| \sigma^2 \nabla_{\mathbf{X}_t} \log \mathbb{Q}_{T|t}(\mathbf{X}_T|\mathbf{X}_t) - (T-s)\nabla V_s(\mathbf{X}_s) - v_t^{\theta_f}(\mathbf{X}_t) \right\|^2 \right] \mathrm{d}s\mathrm{d}t \,, \tag{8.18}$$

$$\mathcal{L}^b_{\mathcal{M}_T}(\theta_b) = \int_0^T t^{-1} \int_0^t \mathbb{E}_{\Pi_{0,s,t}} \left[\left\| \sigma^2 \nabla_{\mathbf{X}_t} \log \mathbb{Q}_{t|0}(\mathbf{X}_t|\mathbf{X}_0) - s\nabla V_s(\mathbf{X}_s) - v_t^{\theta_b}(\mathbf{X}_t) \right\|^2 \right] \mathrm{d}s\mathrm{d}t \,. \tag{8.19}$$

Then, $\mathrm{proj}_{\mathcal{M}_T}(\Pi)$ *can equivalently be approximated by :*

- *the path measure induced by* $\mathrm{d}\mathbf{X}_t = v_t^{\theta_f^\star}(\mathbf{X}_t)\mathrm{d}t + \sigma \mathrm{d}\mathbf{B}_t$*, with* $\mathbf{X}_0 \sim \mu$*,*
- *the time-reversal of the path measure induced by* $\mathrm{d}\mathbf{Y}_t = v_{T-t}^{\theta_b^\star}(\mathbf{Y}_t)\mathrm{d}t + \sigma \mathrm{d}\tilde{\mathbf{B}}_t$*, with* $\mathbf{Y}_0 \sim \nu$*.*

Remarkably, TSBM objectives (8.18) and (8.19) *exactly* recover DSBM objectives (8.9) and (8.10) by taking $V \equiv 0$. Informally, in both the forward and backward settings, the optimal velocity field v_t can be interpreted as a perturbation of the standard control learned in the canonical setting. This perturbation arises from the contribution of the gradient cost ∇V, evaluated along the future portion of forward and backward trajectories at times $s > t$. Its magnitude is typically smaller when s is close to the terminal endpoints, and more pronounced for times nearer to the current time t. Moreover, the derived objectives are completely independent of hyperparameter σ, due to the simplification when exactly computing the terms $\nabla_{\mathbf{X}_t} \log \mathbb{Q}_{T|t}$ and $\nabla_{\mathbf{X}_t} \log \mathbb{Q}_{t|0}$; hence, the regression target remains unchanged in the deterministic setting by letting $\sigma = 0$.

Remark on practical implementation. Given $\Pi \in \mathcal{R}(\mathbb{Q}^V)$, the joint distributions $\Pi_{t,s,T}$ and $\Pi_{0,s,t}$ appearing in the expectations are not directly simulable (unlike in DSBM), precisely because the twisted bridge $\mathbb{Q}^V_{|0,T}$ is not simulable, as discussed in Section 8.3.2. For implementation purposes, we therefore substitute Π by its variational approximation $\Pi^{\psi^\star} = \mathbb{Q}^{\psi^\star}_{|0,T}\Pi_{0,T}$, where $\psi^\star$ denotes the minimizer of the variational problem (8.13) with coupling $\Pi_{0,T}$, obtained at the preceding Reciprocal projection step using parametric stochastic interpolants. The expectations are then replaced by Monte Carlo estimators. More precisely, to evaluate the forward Markovian objective (8.18), we set $\psi^\star$ to be the minimizer of forward variational objective (8.16), and perform, in parallel for $i \in \{1, \dots, M\}$, the three following steps:

1. sample $(x_0^i, x_T^i) \sim \Pi_{0,T}$,
2. sample $t^i \sim \mathrm{U}(0,T)$ and $x_t^i \sim \mathbb{Q}^{\psi^\star}_{t^i|0,T}(\cdot|x_0^i, x_T^i)$, which is computationally comparable to DSBM,
3. for $j \in \{1, \dots, J\}$ with $J \geq 1$, sample in parallel $s^{i,j} \sim \mathrm{U}(t^i, T)$ and $x_s^{i,j} \sim \mathbb{Q}^{\psi^\star}_{s^{i,j}|t^i,T}(\cdot|x_t^i, x_T^i)$ by integrating the parametric forward pinned SDE (8.14) over $[t^i, s^{i,j}]$ with initial condition $\mathbf{X}_{t^i} = x_t^i$. This step induces a non-negligible additional cost compared to the original DSBM implementation.

The forward loss $\mathcal{L}^f_{\mathcal{M}_T}(\theta_f)$ is then approximated by

$$\hat{\mathcal{L}}^f_{\mathcal{M}_T}(\theta_f) = \frac{1}{MJ}\sum_{i=1}^{M}\sum_{j=1}^{J}\left\|\sigma^2\nabla_{x_t^i}\log\mathbb{Q}_{T|t^i}(x_T^i|x_t^i) - (T-s^{i,j})\nabla V_{s^{i,j}}(x_t^{i,j}) - v_{t^i}^{\theta_f}(x_t^i)\right\|^2 .$$

Symmetrically, to evaluate the backward Markovian objective (8.19), we may consider the minimizer of backward variational objective (8.17), associated to parametric backward SDEs (8.15), yielding an analogous Monte Carlo estimator.

Although the additional sampling cost induced by the potential V is non-negligible when $\sigma > 0$ (as it may require simulating an SDE over a time horizon of length up to T), the procedure becomes inexpensive in the deterministic regime ($\sigma = 0$). In that case, the forward and backward parametric pinned ODEs admit closed-form solutions; see Appendix G.1 for further details.

Equivalent expression under denoiser-type parameterization. In practice, we may adopt denoiser-type parameterizations, inherited from the canonical diffusion setting, defined by

$$v_t^{\theta_f}(x) = \frac{\tilde{x}_T^{\theta_f}(t,x) - x}{T-t}, \quad v_t^{\theta_b}(x) = \frac{\tilde{x}_0^{\theta_b}(t,x) - x}{t},$$

where $\tilde{x}_T^{\theta_f}$ and $\tilde{x}_0^{\theta_b}$ are predictors of the terminal state x_T and the initial state x_0, respectively, given (t,x). Under this parameterization, the Markovian objectives admit the simplified forms

$$\mathcal{L}^f_{\mathcal{M}_T}(\theta_f) = \int_0^T (T-t)^{-1}\int_t^T \mathbb{E}_{\Pi_{t,s,T}}\left[\left\|-(T-s)\nabla V_s(\mathbf{X}_s) + \frac{1}{T-t}\left(\mathbf{X}_T - \tilde{x}_T^{\theta_f}(t,\mathbf{X}_t)\right)\right\|^2\right] \mathrm{d}s\mathrm{d}t , \tag{8.20}$$

$$\mathcal{L}^b_{\mathcal{M}_T}(\theta_b) = \int_0^T t^{-1}\int_0^t \mathbb{E}_{\Pi_{0,s,t}}\left[\left\|-s\nabla V_s(\mathbf{X}_s) + \frac{1}{t}\left(\mathbf{X}_0 - \tilde{x}_0^{\theta_b}(t,\mathbf{X}_t)\right)\right\|^2\right] \mathrm{d}s\mathrm{d}t . \tag{8.21}$$

8.3.4 Comparison between TSBM and GSBM

To solve (8.12), Liu et al. [2024] propose an iterative scheme (GSBM) with two alternating optimization steps close to the IMF scheme, but the corresponding connection is not explicitly established.

Although Liu et al. [2024] also consider a variational approximation of the twisted bridge based on stochastic interpolants to be used in their bridge matching loss, the related loss derived in [Liu et al., 2024, Proposition 2] does not match the TSM objective (8.16). In particular, their derivation relies on SOC arguments, and the link with the twisted bridge is not made explicit. To reduce computational cost, Liu et al. [2024] further propose a spline parameterization of the stochastic interpolants instead of neural networks (restated in Appendix G.1), leveraging saved control-point trajectories when sampling from the coupling $\mathbb{P}_{0,T}$. We note that this lightweight parameterization can also be adopted within the TSBM variational framework.

The main discrepancy between TSBM and GSBM lies in the Markovian projection loss. Indeed, GSBM relies on forward and backward regression problems respectively given by

$$\arg\min\{\textstyle\int_0^T \mathbb{E}_{\Pi^{\psi^\star}_{0,t,T}}\left[\left\|v_t^{f,\psi^\star}(\mathbf{X}_t|\mathbf{X}_0,\mathbf{X}_T) - v_t^{\theta_f}(\mathbf{X}_t)\right\|^2\right]\mathrm{d}t \,:\, \theta_f\in\Theta\} \quad \text{[Liu et al., 2024, Equation 5]} , \tag{8.22}$$

$$\arg\min\{\textstyle\int_0^T \mathbb{E}_{\Pi^{\psi^\star}_{0,t,T}}\left[\left\|v_t^{b,\psi^\star}(\mathbf{X}_t|\mathbf{X}_0,\mathbf{X}_T) - v_t^{\theta_b}(\mathbf{X}_t)\right\|^2\right]\mathrm{d}t \,:\, \theta_b\in\Theta\} , \tag{8.23}$$

where $v^{f,\psi^\star}$ and $v^{b,\psi^\star}$ respectively solve forward and backward formulations of the variational problem used to approximate the preceding Reciprocal projection step. As in our practical version of the Markovian objective, the expectation is taken with respect to $\Pi^{\psi^\star}$, obtained by combining the available coupling with the variational bridge rather than the intractable twisted bridge. However, the regression target is fundamentally different here, since GSBM objectives given in (8.22) and (8.23) rely on the conditional vector field of the variational interpolant itself, thereby inheriting approximation error from variational optimization.

8.4 Discrete-time Twisted Schrödinger Bridge

In practice, numerical methods for solving SB problems rely on translating continuous-time objectives to a discrete-time setting, which can introduce unavoidable bias. To mitigate this issue, we reformulate the SB problem directly in discrete time. This strategy was recently proposed by Gushchin et al. [2024] in the canonical SB case; here, we extend it to the twisted setting, which is the focus of this paper. In Section 8.4.1, we introduce the resulting discrete-time formulation and derive an ideal IMF scheme aligned with the continuous-time procedure reviewed in Section 8.2. We then translate this ideal scheme into a practical algorithm, namely the discrete-time version of TSBM, with the Markovian projection described in Section 8.4.2 and the reciprocal projection in Section 8.4.3. Proofs of the propositions stated in this section are deferred to Appendix G.2.

Throughout this section, we consider a time-discretization grid $\{t_n\}_{n=0}^{N+1}$ of $[0,T]$ with $(N+2)$ increasing time points, such that $t_0 = 0 < t_1 < \ldots < t_{N+1} = T$, and we define the step sizes $\delta_n = t_n - t_{n-1}$ for $n \in \{1,\ldots,N+1\}$.

8.4.1 Setting and ideal IMF scheme

Discrete-time twisted reference measure. Let $q \in \mathscr{P}^{(N+2)}$ denote the joint distribution given by the finite-dimensional projection of $\mathbb{Q}$ at times $\{t_n\}_{n=0}^{N+1}$. In particular, q is Markov and can be written as

$$q(x_{0:N+1}) = \delta_0(x_0) \prod_{n=1}^{N+1} q_{n|n-1}(x_n|x_{n-1}) \;, \quad \text{where} \quad q_{n|n-1}(\cdot|x_{n-1}) = \mathrm{N}(x_{n-1}, \sigma^2(t_n - t_{n-1})\mathrm{I}_d) \;.$$

To define a discrete-time analogue of the SB problem (8.1), we introduce a twisted version of q that will serve as reference. To this end, we replace the continuous-time potential V by a collection of N potentials $\{g_n\}_{n=1}^{N}$ satisfying $g_n \in \mathrm{C}^1(\mathbb{R}^d, (0,\infty))$, and we define the *reference* distribution $q^g \in \mathscr{P}^{(N+2)}$, absolutely continuous with respect to q, by

$$q^g(x_{0:N+1}) = \frac{1}{Z_g} \left(\prod_{n=1}^{N} g_n(x_n) \right) q(x_{0:N+1}) \;, \tag{8.24}$$

with normalizing constant $Z_g = \int_{(\mathbb{R}^d)^{N+2}} \prod_{n=1}^{N} g_n(x_n) \mathrm{d}q(x_{0:N+1})$. Thus, q^g is a Feynman–Kac model with prior q; in line with the continuous-time setting, we refer to q^g as a twisted version of q. Such twisted measures are ubiquitous in hidden Markov modeling and often simulated by Sequential Monte Carlo (SMC), see e.g., Heng et al. [2020]. Intuitively, each potential g_n provides guidance on top of the prior transition kernel $q_{n|n-1}$, with the uninformative limit case given by $g_n \equiv 1$. In Appendix G.2, we show that, as in the continuous-time setting, both q^g and its associated bridge $q^g_{|0,N+1}$ are Markov. Moreover, their transition kernels are explicit but typically intractable, and can be written as Doob's h-transforms of the corresponding kernels of q and $q_{|0,N+1}$, respectively, induced by the potentials $\{g_n\}_{n=1}^{N}$.

Discrete-time Twisted Schrödinger Bridge problem. Based on (8.24), we define the discrete-time TSB problem as

$$\arg\min\{\mathrm{KL}(p\|q^g) \;:\; p \in \mathscr{P}^{(N+2)}, p_0 = \mu,\; p_{N+1} = \nu\} \;. \tag{8.25}$$

By the KL chain rule, the solution of (8.25), denoted $p^\star$, is the finite-dimensional projection of the continuous-time solution $\mathbb{P}^N$ at times $\{t_n\}_{n=0}^{N+1}$, where $\mathbb{P}^g$ solves (8.1) with reference measure $\mathbb{Q}^{\mathrm{ref}} = \mathbb{Q}^N$, where

$$\mathrm{d}\mathbb{Q}^N(x_{[0:T]}) \propto \prod_{n=1}^{N} g_n(x_{t_n}) \mathrm{d}\mathbb{Q}(x_{[0:T]}) \;.$$

This observation allows us to make explicit the correspondence between the continuous-time formulation (8.1) and the discrete-time formulation (8.25).

Proposition 20 (Informal version). *Let $V \in \mathrm{C}^{0,1}([0,T]\times\mathbb{R}^d, \mathbb{R})$ be such that $Z_V < \infty$. Define the potentials $g_n = \exp(-\delta_n V_{t_n}/\sigma^2)$ for any $n \in \{1,\ldots,N\}$. Then, in the asymptotic regime $\delta_n \to 0$ (equivalently $N \to \infty$), $\mathbb{P}^N$ converges to $\mathbb{P}^\star$.*

We now restate [Gushchin et al., 2024, Theorem 3.1], which was originally formulated for the canonical setting but remains valid here under the same assumptions. In particular, this result provides a discrete-time characterization of $p^\star$, consistent with the continuous-time formulation established in Léonard [2014b].

Proposition 21 (From Gushchin et al. [2024]). *Let $p \in \mathscr{P}^{(N+2)}$ such that $p \in \mathcal{M}_N$, $p = p_{0,N+1} q^g_{1:N|0,N+1}$, $p_0 = \mu$, and $p_{N+1} = \nu$. Then, p is a solution to (8.25).*

This result motivates the design of a discrete-time IMF scheme, which we detail now.

Discrete-time ideal IMF scheme. Let $p \in \mathscr{P}^{(N+2)}$. We define the reciprocal projection of p with respect to q^g, denoted $\mathrm{proj}_{\mathcal{R}(q^g)}(p) \in \mathscr{P}^{(N+2)}$, as the solution of

$$\arg\min\{\mathrm{KL}(p\|r) \,:\, r \in \mathcal{R}(q^g)\}\,, \tag{8.26}$$

where $\mathcal{R}(q^g)$ denotes the class of distributions $r \in \mathscr{P}^{(N+2)}$ that are mixtures of q^g-bridges, *i.e.*, $r = r_{0,N+1} q^g_{1:N|0,N+1}$. In analogy with the continuous-time setting, we refer to this class as the set of reciprocal distributions. Following [Gushchin et al., 2024, Proposition 3.3], this projection admits the explicit form

$$\mathrm{proj}_{\mathcal{R}(q^g)}(p) = p_{0,N+1} q^g_{1:N|0,N+1}\,. \tag{8.27}$$

We also define the Markovian projection of p, denoted $\mathrm{proj}_{\mathcal{M}_N}(p) \in \mathscr{P}^{(N+2)}$, as the solution of

$$\arg\min\{\mathrm{KL}(p\|m) \,:\, m \in \mathcal{M}_N\}\,, \tag{8.28}$$

where $\mathcal{M}_N$ denotes the set of Markov distributions $m \in \mathscr{P}^{(N+2)}$ admitting the same factorization as q, *i.e.*,

$$m(x_0, x_1, \dots, x_{N+1}) = m_0(x_0) \prod_{n=1}^{N+1} m_{n|n-1}(x_n|x_{n-1})\,.$$

In general, this minimization does not yield a closed-form solution. However, when $p \in \mathcal{R}(q^g)$, we can expand the KL objective explicitly, as stated in the following lemma, which yields a closed-form characterization of the minimizer.

Lemma 22. *Let $p \in \mathcal{R}(q^g)$ and $m \in \mathcal{M}_N$. Then, the following decomposition holds*

$$\begin{aligned}\mathrm{KL}(p\|m) = \mathrm{KL}(p_0\|m_0) &+ \mathbb{E}_{p_{0,N}}[\mathrm{KL}(p_{N+1|0}(\cdot|X_0)\|m_{N+1|N}(\cdot|X_N))] \\ &+ \sum_{n=1}^{N} \mathbb{E}_{p_{n-1,N+1}}[\mathrm{KL}(q^g_{n|n-1,N+1}(\cdot|X_{n-1}, X_{N+1})\|m_{n|n-1}(\cdot|X_{n-1}))]\,.\end{aligned}$$

Therefore, we have $\mathrm{proj}_{\mathcal{M}_N}(p) = p_0 \prod_{n=1}^{N+1} m^\star_{n|n-1}$, where

$$\begin{aligned} m^\star_{n|n-1}(x_n|x_{n-1}) &= \mathbb{E}_{p_{N+1|n-1}}[q^g_{n|n-1,N+1}(x_n|x_{n-1}, X_{N+1})|X_{n-1} = x_{n-1}] && \text{if } n \in \{1,\dots,N\}\,, \\ m^\star_{N+1|N}(x_{N+1}|x_N) &= \mathbb{E}_{p_{0|N}}[p_{N+1|0}(x_{N+1}|X_0)|X_N = x_n] && \text{else.}\end{aligned}$$

We refer to this expression as the forward Markovian projection of p.

Although this decomposition makes the Markovian projection explicit, it remains intractable for general p. We emphasize that such an explicit characterization of the Markovian projection, exploiting the reference measure, is not stated in Gushchin et al. [2024], although related computations appear in their theoretical development. Using the equivalent backward factorization

$$m(x_0, x_1, \dots, x_{N+1}) = m_{N+1}(x_{N+1}) \prod_{n=1}^{N+1} m_{n-1|n}(x_{n-1}|x_n)\,,$$

we derive a symmetric result.

Lemma 23. *Let $p \in \mathcal{R}(q^g)$ and $m \in \mathcal{M}_N$. Then, the following decomposition holds*

$$\begin{aligned}\mathrm{KL}(p\|m) = \mathrm{KL}(p_{N+1}\|m_{N+1}) &+ \mathbb{E}_{p_{1,N+1}}[\mathrm{KL}(p_{0|N+1}(\cdot|X_{N+1})\|m_{0|1}(\cdot|X_1))] \\ &+ \sum_{n=2}^{N+1} \mathbb{E}_{p_{0,n}}[\mathrm{KL}(q^g_{n-1|0,n}(\cdot|X_0, X_n)\|m_{n-1|n}(\cdot|X_n))]\,.\end{aligned}$$

Therefore, we have $\mathrm{proj}_{\mathcal{M}_N}(p) = p_{N+1} \prod_{n=1}^{N+1} m^\star_{n-1|n}$, *where*

$$\begin{aligned} m^\star_{n-1|n}(x_{n-1}|x_n) &= \mathbb{E}_{p_{0|n}}[q^g_{n-1|0,n}(x_{n-1}|X_0, x_n)|X_n = x_n] && \text{if } n \in \{2, \dots, N+1\} , \\ m^\star_{0|1}(x_0|x_1) &= \mathbb{E}_{p_{N+1|1}}[p_{0|N+1}(x_0|X_{N+1})|X_1 = x_1] && \text{else.} \end{aligned}$$

We refer to this expression as the backward Markovian projection of p.

In particular, the KL objective in Lemma 23 is equivalent to the discrete-time DDPM variational objective derived in [Ho et al., 2020, Equation 5] in the case of uninformative potentials, highlighting a tight connection between bridge matching procedures and diffusion-model training objectives beyond the continuous-time setting.

We can now define the ideal discrete-time IMF procedure for solving (8.25) by alternating between the Markovian and Reciprocal projections defined above. Starting from an arbitrary $p^{-1} \in \mathscr{P}^{(N+2)}$ satisfying $p^{-1}_0 = \mu$ and $p^{-1}_{N+1} = \nu$, we define recursively for $n \in \mathbb{N}$

$$p^{2n} = \mathrm{proj}_{\mathcal{R}(q^g)}(p^{2n-1}) , \quad p^{2n+1} = \mathrm{proj}_{\mathcal{M}_N}(p^{2n}) . \tag{8.29}$$

8.4.2 Computation of the Markovian projection

In this subsection, we focus on the practical implementation of the Markovian projection (8.28), restricted to reciprocal distributions $p \in \mathcal{R}(q^g)$, which is precisely the setting encountered in the IMF scheme (8.29). Although explicit forward and backward Markovian factorizations for this class were derived in Lemma 22 and Lemma 23, these expressions remain intractable in general.

As in the continuous-time setting, we restrict the Markovian projection to a prescribed parametric family indexed by $\theta \in \Theta$, and reformulate the KL objectives of Lemma 22 and Lemma 23 as tractable training losses. More specifically, we consider Markov distributions with Gaussian transition kernels, equipped with a denoiser-based parameterization consistent with standard diffusion-model practice and applicable for arbitrary N.

Before introducing this parameterization, we recall the general expression of the discrete-time Brownian bridge marginals: for any triplet $(i, j, k) \in \{0, \dots, N+1\}$ such that $j < i < k$,

$$q_{i|j,k}(x_i|x_j, x_k) = \mathrm{N}\left(\alpha^{j,k}_i x_j + \beta^{j,k}_i x_k, \sigma^2(\gamma^{j,k}_i)^2 \mathrm{I}_d\right) , \tag{8.30}$$

$$\text{where } \alpha^{j,k}_i = \frac{t_k - t_i}{t_k - t_j} , \ \beta^{j,k}_i = \frac{t_i - t_j}{t_k - t_j} , \ \gamma^{j,k}_i = \sqrt{\frac{(t_i - t_j)(t_k - t_i)}{t_k - t_j}} . \tag{8.31}$$

Denoiser-like Markovian parameterization. Inspecting the KL objectives in Lemma 22 and Lemma 23, a natural strategy would be to parameterize Markov kernels using the twisted bridge $q^g_{|0,N+1}$, mirroring the DDPM parameterization [Ho et al., 2020] in the canonical case where $g_n \equiv 1$. However, since $q^g_{|0,N+1}$ is intractable for general potentials, this approach is not directly applicable. We therefore retain the original DDPM parameterization based on the Brownian bridge $q_{|0,N+1}$. For the forward Markovian projection, we consider

$$\begin{aligned} m_{n|n-1}(x_n|x_{n-1}) &= q_{n|n-1,N+1}(x_n|x_{n-1}, \tilde{x}^{\theta_f}_{N+1}(t_{n-1}, x_{n-1})) && \text{if } n \in \{1, \dots, N\} \\ m_{N+1|N}(x_{N+1}|x_N) &= \mathrm{N}(x_{N+1}; \tilde{x}^{\theta_f}_{N+1}(t_N, x_N), \sigma^2_{N+1}\mathrm{I}_d)\text{else} , \end{aligned} \tag{8.32}$$

where $\tilde{x}^{\theta_f}_{N+1}$ is intended to predict x_{N+1}, and $\sigma_{N+1} > 0$ is left as a hyperparameter. Symmetrically, for the backward projection,

$$\begin{aligned} m_{n-1|n}(x_{n-1}|x_n) &= q_{n-1|0,n}(x_{n-1}|\tilde{x}^{\theta_b}_0(t_n, x_n), x_n) && \text{if } n \in \{2, \dots, N+1\} \\ m_{0|1}(x_0|x_1) &= \mathrm{N}(x_0; \tilde{x}^{\theta_b}_0(t_1, x_1), \sigma^2_0 \mathrm{I}_d) , \end{aligned} \tag{8.33}$$

where $\tilde{x}^{\theta_b}_0$ is intended to predict x_0, and $\sigma_0 > 0$ is left as a hyperparameter. This variational family yields computable Markovian projection objectives for any $N \geq 1$.

Proposition 24. *Let $p \in \mathcal{R}(q^g)$. Define $\theta_f^\star$ and $\theta_b^\star$ as the minimizers over Θ of the respective following objectives*

$$\mathcal{L}^f_{\mathcal{M}_N}(\theta_f) = \sum_{n=1}^{N} \frac{1}{N-n+1} \sum_{k=n}^{N} \mathbb{E}_{p_{n-1,k,N+1}} \left[\left\| A_n^f \sigma^2 (T - t_k) \nabla \log g_k(X_k) + B_n^f \left(X_{N+1} - \tilde{x}^{\theta_f}_{N+1}(t_{n-1}, X_{n-1}) \right) \right\|^2 \right] + \frac{\sigma^2}{\sigma^2_{N+1}} \mathbb{E}_{p_{N,N+1}} \left[\left\| X_{N+1} - \tilde{x}^{\theta_f}_{N+1}(t_N, X_N) \right\|^2 \right] , \tag{8.34}$$

$$\mathcal{L}^b_{\mathcal{M}_N}(\theta_b) = \sum_{n=2}^{N+1} \frac{1}{n-1} \sum_{k=1}^{n-1} \mathbb{E}_{p_{0,k,n}} \left[\left\| A_n^b \sigma^2 t_k \nabla \log g_k(X_k) + B_n^b \left(X_0 - \tilde{x}^{\theta_b}_0(t_n, X_n) \right) \right\|^2 \right] + \frac{\sigma^2}{\sigma_0^2} \mathbb{E}_{p_{0,1}} \left[\left\| X_0 - \tilde{x}^{\theta_b}_0(t_1, X_1) \right\|^2 \right] , \tag{8.35}$$

where $A_n^f = \gamma_n^{n-1,N+1} \times \frac{N-n+1}{T-t_n}$, $B_n^f = \frac{\beta_n^{n-1,N+1}}{\gamma_n^{n-1,N+1}}$, $A_n^b = \gamma_{n-1}^{0,n} \times \frac{n-1}{t_{n-1}}$ and $B_n^b = \frac{\alpha_{n-1}^{0,n}}{\gamma_{n-1}^{0,n}}$. Then, the Markovian projection of p restricted to the denoiser-based family, denoted $m^\star$, is equivalently defined by :

- *$m_0^\star = \mu$ and forward kernels (8.32) parameterized by $\tilde{x}^{\theta_f^\star}_{N+1}$,*
- *$m_{N+1}^\star = \nu$ and backward kernels (8.33) parameterized by $\tilde{x}^{\theta_b^\star}_0$.*

In practice, $\tilde{x}^{\theta_f}_0$ and $\tilde{x}^{\theta_b}_{N+1}$ are implemented as neural networks. In the case $g_n \equiv 1$, we recover exactly the classical DDPM objective for the backward loss (8.35), as expected.

Link with continuous-time setting. When setting $g_n = \exp(-\delta_n V_{t_n}/\sigma^2)$ to match the continuous-time formulation (see Proposition 20) and taking $\delta_n = o(1)$, the following approximations hold

$$A_n^f \sigma^2 (T - t_k) \nabla \log g_k(X_k) \approx -\sqrt{\delta_n}(T - t_k) \nabla V_{t_k}(X_k) , \quad B_n^f \approx \frac{\sqrt{\delta_n}}{T - t_n} ,$$
$$A_n^b \sigma^2 t_k \nabla \log g_k(X_k) \approx -\sqrt{\delta_n} t_k \nabla V_{t_k}(X_k) , \quad B_n^b \approx \frac{\sqrt{\delta_n}}{t_n} .$$

Hence, it is clear that the denoiser-like TSBM objectives (8.20) and (8.21) can respectively be seen as the time-continuous limit of objectives (8.34) and (8.35). Note also, that in this case, the regression targets become independent of σ (with suitable choices of σ_0 and σ_{N+1}), thereby connecting to the deterministic regime.

Remaining computational challenge. As in continuous time, evaluating the forward and backward objectives reduces to sampling from triplets $p_{n-1,k,N+1}$ for $k \in \{n, \dots, N\}$ or $p_{0,k,n}$ for $k \in \{1, \dots, n-1\}$. Since $p \in \mathcal{R}(q^g)$ and samples from $p_{0,N+1}$ are available through the IMF iterations, the remaining task is sampling from the twisted bridge $q^g_{n-1,k|0,N+1}$ for $k \geq n$ or $q^g_{k,n|0,N+1}$ for $k \leq n-1$. While particle-based methods such as Sequential Monte Carlo are natural candidates for Feynman-Kac models [Heng et al., 2020], they become impractical when N or d is large. We therefore propose a novel MCMC-based approach, more scalable, detailed in the next section.

8.4.3 Computation of the Reciprocal projection

For the discrete-time version of TSBM, we replace the Reciprocal projection step by a sampling procedure, rather than a variational optimization stage as in the continuous-time setting. This makes the procedure learning-free and removes any dependence on variational approximation error. Concretely, to evaluate the Markovian objectives from Section 8.4.2, it suffices to sample from the following twisted bridges: $q^g_{n-1,k|0,N+1}$ for $k \geq n$ (forward case), or $q^g_{k,n|0,N+1}$ for $k \leq n-1$ (backward case), each conditioned on a pair (x_0, x_{N+1}) produced by the previous IMF Markovian projection. In the remainder of this subsection, we fix such a pair (x_0, x_{N+1}). Although we present the method for a single conditioning pair, it parallelizes directly over multiple pairs and thus accommodates large batch sizes.

Inspired by the use of control points in the continuous-time variational approximation of Liu et al. [2024], we introduce a two-step sampling methodology that leverages the structure of the twisted bridge:

1. Sample from $(x_{m_1}, \dots, x_{m_C}) \sim q^g_{m_1:m_C|0,N+1}(\cdot|x_0, x_{N+1})$, where $\{m_j\}_{j=1}^C \subset \{1, \dots, N\}$ is a prescribed set of $C \geq 1$ control indices satisfying $0 < m_1 < \cdots < m_C < N+1$, with $C \ll N$.

2. Using these samples, sample from $q^g_{n-1,k|m_0:m_{C+1}}(\cdot|x_0, x_{m_1}, \dots, x_{m_C}, x_{N+1})$ with $k \geq n$ (forward), or from $q^g_{k,n|m_0:m_{C+1}}(\cdot|x_0, x_{m_1}, \dots, x_{m_C}, x_{N+1})$ with $k \leq n-1$ (backward).

We now detail these two steps.

First sampling step. Fix control indices $\{m_j\}_{j=1}^C$, and set $m_0 = 0$, $m_{C+1} = N+1$, and $\mathrm{M}_C = \{m_j\}_{j=0}^{C+1}$. Since the twisted bridge $q^g_{|0,N+1}$ is Markov, we obtain the decomposition[2]

$$q^g_{m_1:m_C|0,N+1} \propto \left(\prod_{i=0}^{C} \int \Big(\prod_{j=m_i+1}^{m_{i+1}-1} g_j\Big) \mathrm{d}q_{m_i+1:m_{i+1}-1|m_i,m_{i+1}}\right) \left(\prod_{i=1}^{C} g_{m_i}\right) q_{m_1:m_C|0,N+1} \,. \tag{8.36}$$

This distribution is not tractable, due to the integral terms, and therefore cannot be sampled from directly. Following the approach of Moufad et al. [2025] for posterior diffusion sampling, we introduce a surrogate by (i) assuming conditional mutual independence of the marginals of $q_{m_i+1:m_{i+1}-1|m_i,m_{i+1}}$, which allows us to separate the expectations in (8.36), and (ii) replacing each marginal integral $\int g_j \mathrm{d}q_{j|m_i,m_{i+1}}$ by its Laplace approximation. The resulting surrogate reads

$$\begin{aligned}\tilde{q}^g_{m_1:m_C|0,N+1}(x_{m_1:m_C}|x_0, x_{N+1}) \propto &\prod_{i=0}^{C} \prod_{j=m_i+1}^{m_{i+1}-1} g_j(\mathbb{E}[X_j|x_{m_i}, x_{m_{i+1}}]) \\ &\times \left(\prod_{i=1}^{C} g_{m_i}(x_{m_i})\right) q_{m_1:m_C|0,N+1}(x_{m_1:m_C}|x_0, x_{N+1})\end{aligned} \tag{8.37}$$

where the expectations are taken under the discrete-time Brownian bridge marginals given in (8.30), and therefore, are tractable. In particular, one expects $\tilde{q}^g_{m_1:m_C|0,N+1}$ to become closer to $q^g_{m_1:m_C|0,N+1}$ as C increases. Moreover, for any $i \in \{1, \dots, C\}$, we have

$$\begin{aligned}\tilde{q}^g(x_{m_i}|x_{m\in \mathrm{M}_C\setminus\{m_i\}}) \propto &\left(\prod_{j=m_{i-1}+1}^{m_i-1} g_j(\mathbb{E}[X_j|x_{m_{i-1}}, x_{m_i}])\right) \left(\prod_{j=m_i+1}^{m_{i+1}-1} g_j(\mathbb{E}[X_j|x_{m_i}, x_{m_{i+1}}])\right) \\ &\times g_{m_i}(x_{m_i}) q_{m_i|m_{i-1},m_{i+1}}(x_{m_i}|x_{m_{i-1}}, x_{m_{i+1}}) \,,\end{aligned} \tag{8.38}$$

with associated conditional score

$$\begin{aligned}x_{m_i} \mapsto &\nabla \log g_{m_i}(x_{m_i}) + \nabla \log q_{m_i|m_{i-1},m_{i+1}}(x_{m_i}|x_{m_{i-1}}, x_{m_{i+1}}) \\ &+ \sum_{j=m_{i-1}+1}^{m_i-1} \frac{t_j - t_{m_{i-1}}}{t_{m_i} - t_{m_{i-1}}} \nabla \log g_j(\mathbb{E}[X_j|x_{m_{i-1}}, x_{m_i}]) \\ &+ \sum_{j=m_i+1}^{m_{i+1}-1} \frac{t_{m_{i+1}} - t_j}{t_{m_{i+1}} - t_{m_i}} \nabla \log g_j(\mathbb{E}[X_j|x_{m_i}, x_{m_{i+1}}]) \,.\end{aligned} \tag{8.39}$$

In particular, both (8.38) and (8.39) are tractable and depend only on the neighboring control points $x_{m_{i-1}}$ and $x_{m_{i+1}}$.

We therefore propose sampling from $\tilde{q}^g_{m_1:m_C|0,N+1}$, instead of $q^g_{m_1:m_C|0,N+1}$, using a Gibbs sampler that cycles over $\{m_j\}_{j=1}^C$. Each Gibbs update requires sampling from (8.38), which we perform using a first-order MCMC method exploiting both the unnormalized density and the score (8.39), such as MALA [Roberts and Tweedie, 1996], which is inexpensive in practice. To initialize the Gibbs chain, we suggest adopting the warm-start strategy of Liu et al. [2024], and initialize $x_{m_1}, \dots, x_{m_C}$ with values stored during the simulation of the Markovian projection that produced the pair (x_0, x_{N+1}).

Second sampling step. Assume access to samples $(x_{m_1:m_C})$, ideally drawn from $q^g_{m_1:m_C|0,N+1}(\cdot|x_0, x_{N+1})$. We explain how to sample from $q^g_{k,n|m_0:m_{C+1}}(\cdot|x_{m_0:m_{C+1}})$ in the backward case for some $k \leq n-1$; the forward case follows the same principles.

[2] We omit the conditioning points to keep the expression light.

Let $m(n)^-$ denote the closest control index m_j such that $m_j < n$, and let $m(n)^+$ denote the closest control index m_j such that $m_j > n$; we adopt the same notation for k. Since $q^g_{k,n|m_0:m_{C+1}}$ is not tractable, we again introduce a Laplace-based surrogate, analogous to the first step:

$$\begin{aligned}\tilde{q}^g_{k,n|m_0:m_{C+1}}(x_k, x_n|x_{m_0:m_{C+1}}) \propto &\left(\prod_{j=m(k)^-+1}^{k-1} g_j(\mathbb{E}[X_j|x_{m(k)^-}, x_k])\right)\left(\prod_{j=k+1}^{m(k)^+-1} g_j(\mathbb{E}[X_j|x_k, x_{m(k)^+}])\right)\\ &\times\left(\prod_{j=m(n)^-+1}^{n-1} g_j(\mathbb{E}[X_j|x_{m(n)^-}, x_n])\right)\left(\prod_{j=n+1}^{m(n)^+-1} g_j(\mathbb{E}[X_j|x_n, x_{m(n)^+}])\right)\\ &\times g_k(x_k)g_n(x_n)q_{k,n|m_0:m_{C+1}}(x_k, x_n|x_{m_0:m_{C+1}})\ . \end{aligned} \tag{8.40}$$

As in the first step, we can sample from this surrogate using a Gibbs sampler alternating over indices k and n, with each update implemented via a first-order MCMC scheme.

8.5 Conclusion & Perspectives

In this work, we provide a rigorous reformulation of the trajectory inference problem with endpoint marginal constraints introduced by Liu et al. [2024] within the framework of the Schrödinger Bridge (SB). We show that this problem can be recast as a SB problem whose reference path measure is no longer standard Brownian motion, as in the canonical case, but rather a Feynman–Kac–twisted version of it. This twisted reference measure is characterized by a time-dependent potential function evaluated along the stochastic dynamics, with the classical SB setting recovered as the special case of a zero potential. We refer to this generalized formulation as the Twisted Schrödinger Bridge (TSB) problem.

Building on this formulation, we translate the ideal Iterative Markovian Fitting (IMF) scheme into a concrete computational procedure, which departs from the Generalized Schrödinger Bridge Matching (GSBM) method of Liu et al. [2024]. The resulting algorithm, which we term Twisted Schrödinger Bridge Matching (TSBM), involves two main modifications. First, we derive a slightly different variational objective for the reciprocal projection step. Second, we introduce a potential-informed Markovian projection loss, applicable in both forward and backward time directions, thereby extending the DSBM losses of Shi et al. [2023] to the twisted setting.

To avoid the intrinsic continuous-to-discrete translation issues arising in practical implementations, we further introduce a discrete-time formulation of the TSB problem. This formulation operates on joint distributions rather than path measures and is obtained by considering a finite number of discretization steps. While the discrete-time SB formulation has been studied in the canonical case by Gushchin et al. [2024], we extend this framework to the twisted setting. After deriving the corresponding ideal IMF scheme, we again translate it into a computational procedure. In particular, we obtain a tractable objective for the Markovian projection step, establish its connection with the continuous-time limit, and design an advanced sampling strategy for the reciprocal projection step.

Future work will focus on implementing and empirically evaluating the proposed continuous- and discrete-time computational schemes in trajectory inference tasks, a prominent examples being single-cell RNA sequencing. In the continuous-time setting, a key question is whether and to what extent the TSBM framework reduces the bias observed in the GSBM scheme. In the discrete-time setting, important issues include understanding how the denoiser-like objective depends on the number of discretization steps (where fewer steps may be preferable in principle) as well as analyzing the sensitivity of the proposed reciprocal projection sampling procedure to its hyperparameters.

Appendices

Appendix A

Appendix of Chapter 2

Organisation of the supplementary

This appendix is organized as follows. Appendix A.1 summarizes general facts that will be useful for proofs. Appendix A.2 and Appendix A.3 provide additional details on Riemannian metrics and technicalities on self-concordance, respectively. The c-BHMC algorithm is presented in Appendix A.4 with some results whose proofs are given in Appendix A.5. Appendix A.6 presents general facts about numerical integration and specific facts on the integrators used in n-BHMC. Appendix A.7 dispenses the proof of the result on implicit integrators of n-BHMC, stated in Section 2.4.1. Appendix A.8 presents the modified version of n-BHMC, incorporating a step-size condition, for which we state the reversibility with respect to the target distribution in Section 2.4.2. Proof of this last result is given in Appendix A.9. A detailed comparison between n-BHMC and CRHMC [Kook et al., 2022] is given in Appendix A.10. Finally, Appendix A.11 provides more details on the experiments of Section 2.6.

Notation. For any Riemannian manifold $(\mathsf{M}, \mathfrak{g})$ and any $z = (x, p) \in \mathrm{T}^\star\mathsf{M}$, we denote by $\|\cdot\|_z$ the norm (2.17) defined on $\mathrm{T}^\star\mathsf{M}$ by $\|(x', p')\|_z = \|x'\|_{\mathfrak{g}(x)} + \|p'\|_{\mathfrak{g}(x)^{-1}}$ for any $(x', p') \in \mathrm{T}^\star\mathsf{M}$. For any open subset $\mathsf{U} \subset \mathbb{R}^d$ and any $k \in \mathbb{N}$, we denote by $\mathrm{C}^k(\mathsf{U}, \mathbb{R}^{d'})$ the set of functions $f : \mathsf{U} \to \mathbb{R}^{d'}$ such that f is k times continuously differentiable.

A.1 Useful facts and lemmas

We recall here some basic knowledge on linear algebra and probability, and state useful inequalities for our proofs.

Linear algebra reminders. For any matrices $\mathrm{A}, \mathrm{B} \in \mathbb{R}^{d\times d}$, we write $\mathrm{A} \preceq \mathrm{B}$ if for any $x \in \mathbb{R}^d$, $x^\top(\mathrm{B}-\mathrm{A})x \geq 0$. Any positive-definite matrix $\mathrm{A} \in \mathbb{R}^{d\times d}$ induces a scalar product $\langle\cdot,\cdot\rangle_\mathrm{A}$ on $\mathbb{R}^d$, defined by $\langle x, y\rangle_\mathrm{A} = \langle x, \mathrm{A}y\rangle$. This scalar product induces the norm $\|\cdot\|_\mathrm{A}$ on $\mathbb{R}^d$, defined by $\|x\|_\mathrm{A} = \sqrt{\langle x, x\rangle_\mathrm{A}} = \|\mathrm{A}^{1/2}x\|_2$. For any positive-definite matrices $\mathrm{A}, \mathrm{B} \in \mathbb{R}^{d\times d}$ and for any $\alpha \geq 0$, $\mathrm{A} \preceq \alpha\mathrm{B}$ is equivalent to $\|\cdot\|_\mathrm{A} \leq \sqrt{\alpha}\|\cdot\|_\mathrm{B}$. The canonical norm of $\mathbb{R}^d$ induces the norm $\|\cdot\|_2$ on $\mathbb{R}^{d\times d}$, defined by $\|\mathrm{A}\|_2 = \sup_{\|u\|_2=1}\|\mathrm{A}u\|_2$. In particular, for any matrices $\mathrm{A}, \mathrm{B} \in \mathbb{R}^{d\times d}$ and any vector $x \in \mathbb{R}^d$, $\|\mathrm{A}x\|_2 \leq \|\mathrm{A}\|_2\|x\|_2$ and $\|\mathrm{AB}\|_2 \leq \|\mathrm{A}\|_2\|\mathrm{B}\|_2$. Moreover, if A is non-negative-definite, then $\|\mathrm{A}\|_2 = \lambda(\mathrm{A})$, where $\lambda(\mathrm{A})$ is the largest eigenvalue of A, and $\|\mathrm{A}^\alpha\|_2 = \lambda(\mathrm{A})^\alpha$ for any $\alpha > 0$.

Probability reminders. In this section, we consider a smooth manifold $\mathsf{M} \subset \mathbb{R}^d$. We first recall the definition of reversibility [Douc et al., 2018, Definition 1.5.1] before stating general results on reversibility.

Definition 25. *Let* $\mathrm{Q} : \mathrm{T}^\star\mathsf{M} \times \mathcal{B}(\mathrm{T}^\star\mathsf{M}) \to [0, 1]$ *be a transition probability kernel and let* $\bar{\pi}$ *be a probability distribution on* $\mathrm{T}^\star\mathsf{M}$. *Then,*

(a) Q *is said to be reversible with respect to* $\bar{\pi}$ *if for any* $f \in \mathrm{C}(\mathrm{T}^\star\mathsf{M} \times \mathrm{T}^\star\mathsf{M}, \mathbb{R})$ *with compact support*

$$\int_{\mathrm{T}^\star\mathsf{M}\times\mathrm{T}^\star\mathsf{M}} f(z, z')\mathrm{Q}(z, \mathrm{d}z')\bar{\pi}(\mathrm{d}z) = \int_{\mathrm{T}^\star\mathsf{M}\times\mathrm{T}^\star\mathsf{M}} f(z', z)\mathrm{Q}(z, \mathrm{d}z')\bar{\pi}(\mathrm{d}z)\,. \tag{A.1}$$

(b) Q *is said to be reversible up to momentum reversal with respect to* $\bar{\pi}$ *if for any* $f \in \mathrm{C}(\mathrm{T}^\star\mathsf{M} \times \mathrm{T}^\star\mathsf{M}, \mathbb{R})$ *with compact support*

$$\int_{\mathrm{T}^\star\mathsf{M}\times\mathrm{T}^\star\mathsf{M}} f(z,z')\mathrm{Q}(z,\mathrm{d}z')\bar{\pi}(\mathrm{d}z) = \int_{\mathrm{T}^\star\mathsf{M}\times\mathrm{T}^\star\mathsf{M}} f(s(z'),s(z))\mathrm{Q}(z,\mathrm{d}z')\bar{\pi}(\mathrm{d}z)\ , \tag{A.2}$$

where we recall that for any $(x,p) \in \mathbb{R}^d \times \mathbb{R}^d$, $s(x,p) = (x,-p)$.

Lemma 26. *Let* $\mathrm{Q} : \mathrm{T}^\star\mathsf{M} \times \mathcal{B}(\mathrm{T}^\star\mathsf{M}) \to [0,1]$ *be a transition probability kernel and let* $\bar{\pi}$ *be a probability distribution on* $\mathrm{T}^\star\mathsf{M}$. *Assume that* $s_\#\bar{\pi} = \bar{\pi}$ *and that* Q *is reversible up to momentum reversal with respect to* $\bar{\pi}$. *Then* $\bar{\pi}$ *is an invariant measure for* Q, *i.e., for any* $f \in \mathrm{C}(\mathrm{T}^\star\mathsf{M}, \mathbb{R})$ *with compact support*

$$\int_{\mathrm{T}^\star\mathsf{M}\times\mathrm{T}^\star\mathsf{M}} f(z')\mathrm{Q}(z,\mathrm{d}z')\bar{\pi}(\mathrm{d}z) = \int_{\mathrm{T}^\star\mathsf{M}\times\mathrm{T}^\star\mathsf{M}} f(z)\bar{\pi}(\mathrm{d}z)\ .$$

Proof. Let $f \in \mathrm{C}(\mathrm{T}^\star\mathsf{M}, \mathbb{R})$ with compact support. We have

$$\begin{aligned}
&\int_{\mathrm{T}^\star\mathsf{M}\times\mathrm{T}^\star\mathsf{M}} f(z')\mathrm{Q}(z,\mathrm{d}z')\bar{\pi}(\mathrm{d}z) \\
&\quad = \int_{\mathrm{T}^\star\mathsf{M}\times\mathrm{T}^\star\mathsf{M}} f(s(z))\mathrm{Q}(z,\mathrm{d}z')\bar{\pi}(\mathrm{d}z) && \text{(Definition 25)} \\
&\quad = \int_{\mathrm{T}^\star\mathsf{M}\times\mathrm{T}^\star\mathsf{M}} f(z)\mathrm{Q}(s(z),\mathrm{d}z')(s_\#\bar{\pi})(\mathrm{d}z) = \int_{\mathrm{T}^\star\mathsf{M}} f(z)\bar{\pi}(\mathrm{d}z) && \text{(momentum reversal on } z)\ ,
\end{aligned}$$

which concludes the proof. □

Lemma 27. *Let* $\mathrm{Q} : \mathrm{T}^\star\mathsf{M} \times \mathcal{B}(\mathrm{T}^\star\mathsf{M}) \to [0,1]$ *be a transition probability kernel and let* $\bar{\pi}$ *be a probability distribution on* $\mathrm{T}^\star\mathsf{M}$. *Assume that* $s_\#\mathrm{Q} = \mathrm{Q}$ *and that* Q *is reversible with respect to* $\bar{\pi}$. *Then,* Q *is reversible up to momentum reversal with respect to* $\bar{\pi}$.

Proof. Let $f \in \mathrm{C}(\mathrm{T}^\star\mathsf{M}, \mathbb{R})$ with compact support. We have

$$\begin{aligned}
&\int_{\mathrm{T}^\star\mathsf{M}\times\mathrm{T}^\star\mathsf{M}} f(z,z')\mathrm{Q}(z,\mathrm{d}z')\bar{\pi}(\mathrm{d}z) \\
&\quad = \int_{\mathrm{T}^\star\mathsf{M}\times\mathrm{T}^\star\mathsf{M}} f(z,s(z'))(s_\#\mathrm{Q})(z,\mathrm{d}z')\bar{\pi}(\mathrm{d}z) && \text{(momentum reversal on } z') \\
&\quad = \int_{\mathrm{T}^\star\mathsf{M}\times\mathrm{T}^\star\mathsf{M}} f(z',s(z))\mathrm{Q}(z,\mathrm{d}z')\bar{\pi}(\mathrm{d}z) && \text{(Definition 25)} \\
&\quad = \int_{\mathrm{T}^\star\mathsf{M}\times\mathrm{T}^\star\mathsf{M}} f(s(z'),s(z))(s_\#\mathrm{Q})(z,\mathrm{d}z')\bar{\pi}(\mathrm{d}z) && \text{(momentum reversal on } z') \\
&\quad = \int_{\mathrm{T}^\star\mathsf{M}\times\mathrm{T}^\star\mathsf{M}} f(s(z'),s(z))\mathrm{Q}(z,\mathrm{d}z')\bar{\pi}(\mathrm{d}z)\ ,
\end{aligned}$$

which concludes the proof. □

Useful inequalities. The following inequalities hold:

(a) For any $u \in [0,1/5]$, $(1-u)^{-2} \leq 1+3u$ and $(1-u)^{-3} \leq 1+5u$.

(b) For any $u \in [0,1/2]$, $(1-u)^{-1} \leq 1+2u$.

(c) For any $u \in [0,1]$, $|(1-u)^2 - 1| \leq 3u$.

(d) For any $u \geq 0$, we have $1-(1+u)^{-1} \leq u$ and $(1-u)^2 - 1 \geq -2u$.

A.2 Details on Riemannian metrics

Let M be a smooth d-dimensional manifold, endowed with a metric $\mathfrak{g}$. We recall that the *Riemannian volume element* corresponding to $(\mathsf{M},\mathfrak{g})$ is given in local coordinates by $\mathrm{dvol}_\mathsf{M}(x) = \sqrt{\det(\mathfrak{g})}\mathrm{d}x$, where $\mathrm{d}x$ is a dual coframe, [Lee, 2006, Lemma 3.2.]. For any $x \in \mathsf{M}$, we respectively denote by $\mathrm{T}_x\mathsf{M}$ and $\mathrm{T}_x^\star\mathsf{M}$, the tangent space at x and its dual space, *i.e.*, the cotangent space at x. Note that $\mathrm{T}_x\mathsf{M}$ and $\mathrm{T}_x^\star\mathsf{M}$ are space vectors, and $\mathrm{T}_x\mathsf{M}$ is endowed with the scalar product $\langle\cdot,\cdot\rangle_{\mathfrak{g}(x)}$ by definition of the Riemannian metric. For clarity sake, we denote by v (resp. p) an element of the tangent (resp. cotangent) space. We recall that the tangent bundle $\mathrm{T}\mathsf{M}$ and the cotangent bundle $\mathrm{T}^\star\mathsf{M}$ are respectively defined by $\mathrm{T}\mathsf{M} = \sqcup_{x\in\mathsf{M}}\{x\} \cup \mathrm{T}_x\mathsf{M}$ and $\mathrm{T}^\star\mathsf{M} = \sqcup_{x\in\mathsf{M}}\{x\} \cup \mathrm{T}_x^\star\mathsf{M}$. These two sets are $2d$-dimensional manifolds.

Metric on the tangent bundle TM. Sasaki [1958] originally introduced on TM a Riemannian metric $\hat{\mathfrak{g}}$, which, among other properties, preserves the Euclidean metric induced by $\mathfrak{g}$ on each tangent space. This metric is defined by

$$\hat{\mathfrak{g}} = \begin{pmatrix} \mathfrak{g}_{ij} + v^k v^l \Gamma^k_{is} \Gamma^l_{tj} \mathfrak{g}^{st} & -\mathfrak{g}^{js} v^k \Gamma^k_{is} \\ -\mathfrak{g}^{is} v^k \Gamma^k_{js} & \mathfrak{g}_{ij} \end{pmatrix} ,$$

where $\mathfrak{g}_{ij}$ and $\mathfrak{g}^{ij}$ respectively refer to $\mathfrak{g}$ and $\mathfrak{g}^{-1}$, and Γ corresponds to the Christoffel symbol. Since $(\mathsf{TM}, \hat{\mathfrak{g}})$ is a Riemannian manifold, the volume form on TM satisfies for any $(x, v) \in \mathsf{TM}$

$$\mathrm{dvol}_{\mathsf{TM}}(x, v) = \sqrt{\det(\hat{\mathfrak{g}}(x, v))} \mathrm{d}x\mathrm{d}v = \det(\mathfrak{g}(x)) \mathrm{d}x\mathrm{d}v \, .$$

Metric on the cotangent bundle $\mathsf{T}^\star\mathsf{M}$. Elaborating on the metric defined by Sasaki [1958], Mok [1977] showed that for any $x \in \mathsf{M}$, $\mathsf{T}^\star_x\mathsf{M}$ is naturally endowed with the scalar product $\langle \cdot, \cdot \rangle_{\mathfrak{g}(x)^{-1}}$, and proposed a Riemannian metric $\mathfrak{g}^\star$ on $\mathsf{T}^\star\mathsf{M}$, which notably preserves this result on each cotangent space. This metric is closely related to $\hat{\mathfrak{g}}$ and is defined by

$$\mathfrak{g}^\star = \begin{pmatrix} \mathfrak{g}_{ij} + p^k p^l \Gamma^k_{is} \Gamma^l_{tj} \mathfrak{g}^{st} & -\mathfrak{g}^{js} p^k \Gamma^k_{is} \\ -\mathfrak{g}^{is} p^k \Gamma^k_{js} & \mathfrak{g}^{ij} \end{pmatrix} .$$

Since $(\mathsf{T}^\star\mathsf{M}, \mathfrak{g}^\star)$ is a Riemannian manifold, the volume form on $\mathsf{T}^\star\mathsf{M}$ satisfies for any $(x, p) \in \mathsf{T}^\star\mathsf{M}$

$$\mathrm{dvol}_{\mathsf{T}^\star\mathsf{M}}(x, p) = \sqrt{\det(\mathfrak{g}^\star(x, p))} \mathrm{d}x\mathrm{d}p = \mathrm{d}x\mathrm{d}p \, .$$

In contrast to the tangent bundle, the volume form on the cotangent bundle does not depend on $\mathfrak{g}$, which motivates to augment on $\mathsf{T}^\star\mathsf{M}$ (instead of TM) any target measure π defined on M.

A.3 Properties of self-concordance

We first recall the definition of self-concordance and derive some of its properties in Lemmas 29 and 30.

Definition 28 (Nesterov and Nemirovskii [1994])**.** *Let U be a non-empty open convex domain in $\mathbb{R}^d$. A function $\phi : \mathsf{U} \to \mathbb{R}$ is said to be a ν-self-concordant barrier (with $\nu \geq 1$) on U if it satisfies:*

(a) $\phi \in \mathrm{C}^3(\mathsf{U}, \mathbb{R})$ and ϕ is convex,

(b) $\phi(x) \to +\infty$ as $x \to \partial\mathsf{U}$,

(c) $|D^3\phi(x)[h, h, h]| \leq 2\|h\|^3_{\mathfrak{g}(x)}$, for any $x \in \mathsf{M}$, $h \in \mathbb{R}^d$,

(d) $|D\phi(x)[h]|^2 \leq \nu\|h\|^2_{\mathfrak{g}(x)}$, for any $x \in \mathsf{M}$, $h \in \mathbb{R}^d$,

where $\mathfrak{g}(x) = D^2\phi(x)$.

Self-concordance can be thought as a certain kind of regularity. Indeed, if ϕ is a ν-self-concordant barrier on a convex body U, then $D^2\phi$ is 2-Lipschitz continuous on U with respect to the local norm induced by $\mathfrak{g}$ (see Property (c)), and more restrictively, ϕ is ν-Lipschitz continuous with respect to the same norm (see Property (d)).

Lemma 29. *Let $\phi : \mathsf{U} \to \mathbb{R}$ be a ν-self-concordant barrier with $\mathfrak{g} = D^2\phi$. Assume that U is bounded. Then, $\|\nabla\phi(x)\|_2 \to +\infty$ and $\|\mathfrak{g}(x)\|_2 \to +\infty$ as $x \to \partial\mathsf{U}$.*

Proof. Since ϕ is convex, we have, for any $(x, y) \in \mathsf{U}^2$

$$\begin{aligned} \phi(x) &\leq \phi(y) + \nabla\phi(x)(x - y) \, , \\ |\phi(x)| &\leq |\phi(y)| + \|\nabla\phi(x)\|_2 \|x - y\|_2 \\ &\leq |\phi(y)| + \|\nabla\phi(x)\|_2 \, \mathrm{diam}(\mathsf{U}) \, , \end{aligned}$$

where we used Cauchy-Schwartz inequality in the second line. Using Property (b) of ϕ, we obtain that $\|\nabla\phi(x)\|_2 \to +\infty$ as $x \to \partial\mathsf{U}$. By combining this result with Property (d) of ϕ, we obtain that $\|\mathfrak{g}(x)\|_2 \to +\infty$ as $x \to \partial\mathsf{U}$. □

According to Nesterov [2003] (see Lemma 4.1.2), Property (c) of self-concordance is equivalent to

$$|D^3\phi(x)[h_1,h_2,h_3]| \leq 2\prod_{i=1}^3 \|h_i\|_{\mathfrak{g}(x)} \,, \tag{A.3}$$

for any $x \in \mathsf{M}$ and any $(h_1,h_2,h_3) \in \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}^d$.

Let $u : \mathbb{R}^d \to \mathbb{R}^d$ be a linear operator. For any $x \in \mathbb{R}^d$, we define the operator norms $\|u\|_{\mathfrak{g}(x)}$ and $\|u\|_{\mathfrak{g}(x)^{-1}}$ by

$$\begin{aligned} \|u\|_{\mathfrak{g}(x)} &= \sup\{u(\mathfrak{g}(x)h) \,:\, \|h\|_{\mathfrak{g}(x)} = 1\} \,, \\ \|u\|_{\mathfrak{g}(x)^{-1}} &= \sup\{u(\mathfrak{g}(x)^{-1}h) \,:\, \|h\|_{\mathfrak{g}(x)^{-1}} = 1\} \,. \end{aligned} \tag{A.4}$$

In addition, a set $\mathsf{V} \subset \mathbb{R}^d$ contains no straight line if for any $\mathsf{D}_{x_0} = \{tx_0 \,:\, t \in \mathbb{R}\}$ with $x_0 \neq 0$, we have $\mathsf{V}^{\mathrm{c}} \cap \mathsf{D}_{x_0} \neq \emptyset$.

Lemma 30. *Let $\phi : \mathsf{U} \to \mathbb{R}$ be a ν-self-concordant barrier with $\mathfrak{g} = D^2\phi$. Assume that U contains no straight line. For any $x \in \mathsf{U}$ and any $r > 0$, we denote by $\mathsf{W}^0(x,r)$ the open Dikin ellipsoid of ϕ at x, given by $\mathsf{W}^0(x,r) = \{y \in \mathbb{R}^d \mid \|y-x\|_{\mathfrak{g}(x)} < r\}$. Then, the following properties hold:*

(a) For any $x \in \mathsf{U}$ and any $(h_1,h_2) \in \mathbb{R}^d \times \mathbb{R}^d$

$$\|D^3\phi(x)[h_1,h_2]\|_{\mathfrak{g}(x)^{-1}} \leq 2\|h_1\|_{\mathfrak{g}(x)}\|h_2\|_{\mathfrak{g}(x)} \,. \tag{A.5}$$

(b) For any $x \in \mathsf{U}$, $\mathsf{W}^0(x,1) \subset \mathsf{U}$, and for any $y \in \mathsf{W}^0(x,1)$

$$(1-\|y-x\|_{\mathfrak{g}(x)})^2\mathfrak{g}(x) \preceq \mathfrak{g}(y) \preceq (1-\|y-x\|_{\mathfrak{g}(x)})^{-2}\mathfrak{g}(x) \,. \tag{A.6}$$

(c) For any $x \in \mathsf{U}$, any $y \in \mathsf{W}^0(x,1)$ and any $h \in \mathbb{R}^d$, the following hold

$$\begin{aligned} \|h\|_{\mathfrak{g}(x)} &\leq (1-\|y-x\|_{\mathfrak{g}(x)})^{-1}\|h\|_{\mathfrak{g}(y)} \,, \\ \|h\|_{\mathfrak{g}(y)} &\leq (1-\|y-x\|_{\mathfrak{g}(x)})^{-1}\|h\|_{\mathfrak{g}(x)} \,, \\ \|h\|_{\mathfrak{g}(x)^{-1}} &\leq (1-\|y-x\|_{\mathfrak{g}(x)})^{-1}\|h\|_{\mathfrak{g}(y)^{-1}} \,, \\ \|h\|_{\mathfrak{g}(y)^{-1}} &\leq (1-\|y-x\|_{\mathfrak{g}(x)})^{-1}\|h\|_{\mathfrak{g}(x)^{-1}} \,. \end{aligned}$$

Proof. The result is a direct consequence of [Nesterov, 2003, Theorem 4.1.3, Theorem 4.1.5, Theorem 4.1.6]. □

We now introduce α-regularity, which slightly strengthens self-concordance, by ensuring that $D^3\phi$ is $\alpha(\alpha+1)$-Lipschitz continuous with respect to the local norm induced by $\mathfrak{g}$ (see (A.7)). We state below the definition of α-regularity as well as some of its properties in Lemma 32.

Definition 31 (Nesterov and Nemirovski [1998])**.** *Let $\alpha \geq 1$ and U be non-empty open convex domain in $\mathbb{R}^d$. A self-concordant function ϕ on U is said α-regular, if $\phi \in \mathrm{C}^4(\mathsf{U},\mathbb{R})$ and for any $x \in \mathsf{U}$ and any $h \in \mathbb{R}^d$*

$$|D^4\phi(x)[h,h,h,h]| \leq \alpha(\alpha+1)\|h\|^2_{\mathfrak{g}(x)}\|h\|^2_{\mathsf{U},x} \,,$$

where $\|h\|_{\mathsf{U},x} := \inf\{t^{-1} \mid t > 0, x \pm th \in \mathsf{U}\}$.

Lemma 32. *Let $\phi : \mathsf{U} \to \mathbb{R}$ be a α-regular function with $\mathfrak{g} = D^2\phi$. Assume that U contains no straight line.*

(a) For any $x \in \mathsf{U}$ and any $h \in \mathbb{R}^d$

$$|D^4\phi(x)[h,h,h,h]| \leq \alpha(\alpha+1)\|h\|^4_{\mathfrak{g}(x)} \,. \tag{A.7}$$

(b) For any $x \in \mathsf{U}$, any $(h_1,h_2,h_3,h_4) \in \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}^d$

$$|D^4\phi(x)[h_1,h_2,h_3,h_4]| \leq \alpha(\alpha+1)\prod_{i=1}^4 \|h_i\|_{\mathfrak{g}(x)} \,. \tag{A.8}$$

(c) For any $x \in \mathsf{U}$, any $y \in \mathsf{W}^0(x,1)$ and any $(h_1,h_2) \in \mathbb{R}^d \times \mathbb{R}^d$

$$\|D^3\phi(x)[h_1,h_2] - D^3\phi(y)[h_1,h_2]\|_{\mathfrak{g}(x)^{-1}} \leq \alpha(\alpha+1)/3\|h_1\|_{\mathfrak{g}(x)}\|h_2\|_{\mathfrak{g}(x)}((1-\|y-x\|_{\mathfrak{g}(x)})^{-3}-1) \,. \tag{A.9}$$

Proof. We first prove (A.7). First remark that the result is true for $h = 0$. Consider now $h \neq 0$. Let $\varepsilon > 0$. We define $t_\varepsilon = (1+\varepsilon)^{-1}\|h\|^{-1}_{\mathfrak{g}(x)} > 0$, $y^+_\varepsilon = x + t_\varepsilon h$ and $y^-_\varepsilon = x - t_\varepsilon h$. We have $\|x - y^+_\varepsilon\|_{\mathfrak{g}(x)} = \|x - y^-_\varepsilon\|_{\mathfrak{g}(x)} = 1/(1+\varepsilon) < 1$. Hence, $y^-_\varepsilon \in \mathsf{U}$ and $y^+_\varepsilon \in \mathsf{U}$ by Lemma 30. Therefore, $\|h\|_{\mathsf{U},x} \leq t_\varepsilon^{-1}$ and

$$|D^4\phi(x)[h,h,h,h]| \leq \alpha(\alpha+1)\|h\|^4_{\mathfrak{g}(x)}(1+\varepsilon)^2 \, .$$

We conclude upon taking $\varepsilon \to 0$. Then, (A.8) is an immediate consequence of (A.7) using [Nesterov and Nemirovskii, 1994, Proposition 9.1.1]. We now prove (A.9). Using Lemma 30, we have

$$\begin{aligned}
&\|D^3\phi(x)[h_1,h_2] - D^3\phi(y)[h_1,h_2]\|_{\mathfrak{g}(x)^{-1}} \\
&\quad \leq \textstyle\int_0^1 \|D^4\phi(x + t(y-x))[h_1,h_2,y-x]\|_{\mathfrak{g}(x)^{-1}}\mathrm{d}t \\
&\quad \leq \textstyle\int_0^1 (1 - t\|y-x\|_{\mathfrak{g}(x)})^{-1}\|D^4\phi(x + t(y-x))[h_1,h_2,y-x]\|_{\mathfrak{g}(x+t(y-x))^{-1}}\mathrm{d}t \\
&\quad \leq \textstyle\int_0^1 \alpha(\alpha+1)/(1 - t\|y-x\|_{\mathfrak{g}(x)})\|h_1\|_{\mathfrak{g}(x+t(y-x))}\|h_2\|_{\mathfrak{g}(x+t(y-x))}\|y-x\|_{\mathfrak{g}(x+t(y-x))}\mathrm{d}t \\
&\quad \leq \alpha(\alpha+1)\|h_1\|_{\mathfrak{g}(x)}\|h_2\|_{\mathfrak{g}(x)} \textstyle\int_0^1 \|y-x\|_{\mathfrak{g}(x)}/(1 - t\|y-x\|_{\mathfrak{g}(x)})^4\mathrm{d}t \\
&\quad \leq \alpha(\alpha+1)/3\|h_1\|_{\mathfrak{g}(x)}\|h_2\|_{\mathfrak{g}(x)}((1 - \|y-x\|_{\mathfrak{g}(x)})^{-3} - 1) \, ,
\end{aligned}$$

which concludes the proof. □

We now prove the equivalence between the Euclidean norm and the norms induced by $\mathfrak{g}$ and $\mathfrak{g}^{-1}$.

Lemma 33. *Let $(\mathsf{U},\mathfrak{g})$ be non-empty Riemannian manifold in $\mathbb{R}^d$. For any $x_0 \in \mathsf{U}$, any $(x,p) \in \mathsf{U} \times \mathbb{R}^d$, we have*

$$\|\mathfrak{g}(x_0)^{-1}\|_2^{-1/2}\|x\|_2 \leq \|x\|_{\mathfrak{g}(x_0)} \leq \|\mathfrak{g}(x_0)\|_2^{1/2}\|x\|_2 \, , \tag{A.10}$$

$$\|\mathfrak{g}(x_0)\|_2^{-1/2}\|p\|_2 \leq \|p\|_{\mathfrak{g}(x_0)^{-1}} \leq \|\mathfrak{g}(x_0)^{-1}\|_2^{1/2}\|p\|_2 \, . \tag{A.11}$$

In addition, let

$$C_{x_0} = \max(\|\mathfrak{g}(x_0)\|_2^{1/2}, \|\mathfrak{g}(x_0)^{-1}\|_2^{1/2}) > 0 \, .$$

Then, we have

$$1/C_{x_0}(\|x\|_2 + \|p\|_2) \leq \|x\|_{\mathfrak{g}(x_0)} + \|p\|_{\mathfrak{g}(x_0)^{-1}} \leq C_{x_0}(\|x\|_2 + \|p\|_2) \, .$$

A.4 C-BHMC: Algorithm and Results

In this section, we first state general results on the Hamiltonian dynamics (2.7) under our main assumptions. It allows us to introduce the *continuous* version of BHMC (c-BHMC), for which we derive the reversibility with respect to π. Full proofs of this section are provided in Appendix A.5.

Results on the Hamiltonian dynamics. Using Cauchy-Lipschitz's theorem, we obtain in Proposition 34 the existence and uniqueness of the Hamiltonian dynamics (2.7), starting from any $z_0 \in \mathrm{T}^\star\mathsf{M}$.

Proposition 34. *Assume* **A**0, **A**1. *Let $z_0 \in \mathrm{T}^\star\mathsf{M}$. Then (2.7) admits a unique maximal solution (J_{z_0}, z), where (i) $J_{z_0} \subset \mathbb{R}$ is an open neighbourhood of 0, (ii) $z \in \mathrm{C}^1(J_{z_0}, \mathrm{T}^\star\mathsf{M})$, (iii) $z(0) = z_0$ and (iv) for any $t \in J_{z_0}$, $H(z(t)) = H(z_0)$.*

In contrast to the Euclidean case (*i.e.*, $\mathfrak{g}(x) = \mathrm{I}_d$), the Hamiltonian dynamics (2.7) relies on a metric derived from a self-concordant barrier, and thus is defined up to an explosion time. In the next result, we characterize the behaviour of the solutions when this explosion time is finite.

Proposition 35. *Assume* **A**0, **A**1. *Let $z_0 \in \mathrm{T}^\star\mathsf{M}$, (J_{z_0}, z) as in Proposition 34. Assume $T_{z_0} = \sup J_{z_0} < +\infty$, then we have*
(a) $\lim_{t \to T_{z_0}} \|p(t)\|_2 = +\infty$.
(b) There exists $x_\mathsf{M} \in \partial\mathsf{M}$ such that $\lim_{t \to T_{z_0}} x(t) = x_\mathsf{M}$.

In the rest of this section, we define the *Hamiltonian flow* as the map $\mathrm{T} : \mathsf{D} \to \mathrm{T}^\star\mathsf{M}$, where $\mathsf{D} = \{(h, z_0) : z_0 \in \mathrm{T}^\star\mathsf{M}, h < \sup J_{z_0}\}$, such that $\mathrm{T}_h(z_0) = z(h)$ for any $(h, z_0) \in \mathsf{D}$, z being uniquely defined from z_0 in Proposition 34.

Introduction to c-BHMC. In the case where $\mathsf{D} = \mathrm{T}^\star\mathsf{M} \times \mathbb{R}$, the definition of the Hamiltonian dynamics ensures that T_h is involutive (up to momentum reversal) for any $h > 0$, which is key to derive reversibility in BHMC. However, in our manifold setting, there is one subtlety that needs to be checked, namely that the Hamiltonian flow does not leave the cotangent bundle in finite time. As detailed in Proposition 35, there is indeed no guarantee that this flow is defined for all times, *i.e.*, that we have $\mathsf{D} = \mathrm{T}^\star\mathsf{M} \times \mathbb{R}$, due to the ill-conditioned behavior of the dynamics near the boundary of the manifold. Therefore, given a time horizon $h > 0$, we restrict the cotangent bundle to the points where the Hamiltonian flow is defined up to time h. Hence, we define, for any $h > 0$, the sets

$$\mathsf{E}_h = \{z_0 \in \mathrm{T}^\star\mathsf{M} \,:\, h < \sup J_{z_0}\} \,, \quad \bar{\mathsf{E}}_h = \{z_0 \in \mathsf{E}_h \,:\, h < \sup J_{(s\circ \mathrm{T}_h)(z_0)}\} \,. \tag{A.12}$$

Note that elements of $\bar{\mathsf{E}}_h$ correspond to points z_0 for which:

(a) we are able to compute *exactly* the Hamiltonian dynamics (2.7) starting from z_0 on the time interval $[0, h]$, by considering $\{\mathrm{T}_t(z_0)\}_{t\in[0,h]}$ (since $\bar{\mathsf{E}}_h \subset \mathsf{E}_h$),

(b) we are also able to compute *exactly* the **reversed** dynamics - up to momentum reversal - starting from $(s \circ \mathrm{T}_h)(z_0)$ on the time interval $[0, h]$.

In Appendix A.5, we prove that T_h is symplectic (up to momentum reversal) on $\bar{\mathsf{E}}_h$ for any $h > 0$, which thus ensures that the Hamiltonian dynamics is *reversible* with respect to $\bar{\pi}$, see Lemma 37.

Let $h > 0$. Our algorithm c-BHMC, whose pseudo-code is provided in Algorithm 2, then proceeds as follows at stage $n \geq 1$. Assume that the current state is $(X_{n-1}, P_{n-1}) \in \mathrm{T}^\star\mathsf{M}$. First define a partial refreshment of the momentum $\tilde{P}_n \leftarrow \sqrt{1-\beta}P_{n-1} + \sqrt{\beta}G_n$, where $G_n \sim \mathrm{N}_x(0, \mathrm{I}_d)$ is independent from P_{n-1}. Then, verify that $(X_{n-1}, \tilde{P}_n) \in \bar{\mathsf{E}}_h$, which ensures that the Hamiltonian flow is involutive on the time interval $[0, h]$, starting at this state. In the same spirit as in Algorithm 1, we refer to this step as an "involution checking step", which is highlighted in yellow in Algorithm 2, although the involution property directly derives from the Hamiltonian dynamics. If this step is valid, define $(X_n, \bar{P}_n) = \mathrm{T}_h(X_{n-1}, \tilde{P}_n)$; otherwise, set $(X_n, \bar{P}_n) \leftarrow s(X_{n-1}, \tilde{P}_n)$. Finally, update the momentum as in the beginning of the iteration to obtain the new state (X_n, P_n).

Algorithm 2: c-BHMC with Momentum Refresh

Input: $(X_0, P_0) \in \mathrm{T}^\star\mathsf{M}$, $\beta \in (0, 1]$, $N \in \mathbb{N}$, $h > 0$
Output: $(X_n, P_n)_{n\in[N]}$
for $n = 1, ..., N$ **do**
  Step 1: $G_n \sim \mathrm{N}_x(0, \mathrm{I}_d)$, $\tilde{P}_n \leftarrow \sqrt{1-\beta}P_{n-1} + \sqrt{\beta}G_n$
  Step 2: solving continuous ODE (2.7)
  if $(X_{n-1}, \tilde{P}_n) \in \bar{\mathsf{E}}_h$ **then** $(X_n, \bar{P}_n) \leftarrow \mathrm{T}_h(X_{n-1}, \tilde{P}_n)$
  else $(X_n, \bar{P}_n) \leftarrow s(X_{n-1}, \tilde{P}_n)$
  Step 3: $G'_n \sim \mathrm{N}_x(0, \mathrm{I}_d)$, $P_n \leftarrow \sqrt{1-\beta}\bar{P}_n + \sqrt{\beta}G'_n$
end

Denote by $\mathrm{Q}_c : \mathrm{T}^\star\mathsf{M} \times \mathcal{B}(\mathrm{T}^\star\mathsf{M}) \to [0, 1]$, the transition kernel of the (homogeneous) Markov chain $(X_n, P_n)_{n\in[N]}$ obtained with Algorithm 2. Using the properties of the Hamiltonian dynamics, we get the following result.

Theorem 36. *Assume* **A**0, **A**1. *Then,* Q_c *is reversible up to momentum reversal, i.e., we have for any* $f \in \mathrm{C}(\mathrm{T}^\star\mathsf{M} \times \mathrm{T}^\star\mathsf{M}, \mathbb{R})$ *with compact support*

$$\int_{\mathrm{T}^\star\mathsf{M}\times\mathrm{T}^\star\mathsf{M}} f(z, z')\mathrm{d}\bar{\pi}(z)\mathrm{Q}_c(z, \mathrm{d}z') = \int_{\mathrm{T}^\star\mathsf{M}\times\mathrm{T}^\star\mathsf{M}} f(s(z'), s(z))\mathrm{d}\bar{\pi}(z)\mathrm{Q}_c(z, \mathrm{d}z') \,.$$

In particular, $\bar{\pi}$ *is an invariant measure for* Q_c.

A.5 Proofs of Appendix A.4

A.5.1 Existence, uniqueness and explosion time of Hamiltonian dynamics

We prove below Proposition 34 and Proposition 35.

Proof of Proposition 34. Assume **A**0, **A**1. In particular, M contains no straight line, and Lemmas 30 and 32 apply here. We rewrite the ODE (2.7) as a Cauchy problem defined on Banach space $\mathsf{E} = (\mathbb{R}^d \times \mathbb{R}^d, \|\cdot\|_{\mathsf{E}})$

where $\|(a,b)\|_{\mathsf{E}} = \|a\|_2 + \|b\|_2$ for any pair $(a,b) \in \mathbb{R}^d \times \mathbb{R}^d$. We define $\Omega = \mathsf{M} \times \mathbb{R}^d$, which is an open set in E. A solution (J, z) to this Cauchy problem is defined for all $t \in J$ by

$$\dot{z}(t) = F(z(t)) \ , \ z(0) = z_0 \ , \quad \text{where} \quad F(x,p) = (\partial_p H(x,p), -\partial_x H(x,p)) \ ,$$

such that J is an open interval of $\mathbb{R}$ with $0 \in J$, and $z = (x,p) : J \to \Omega$ is differentiable on J. We now prove two results on F to establish existence and uniqueness of a maximal solution (J,z) for this Cauchy problem:

(a) F is continuously differentiable on Ω.

(b) F is locally Lipschitz-continuous on Ω with respect to $\|\cdot\|_{\mathsf{E}}$.

Since $\nabla V, D\mathfrak{g} \in \mathrm{C}^1(\mathsf{M}, \mathbb{R})$, we directly obtain Item (a). We now prove Item (b). Let $\bar{z} = (\bar{x}, \bar{p}) \in \Omega$. We endow E with the norm $\|\cdot\|_{\bar{z}}$. Since M is an open subset of $\mathbb{R}^d$, there exists $0 < r \leq 1/(11 C_{\bar{x}})$, where we recall that $C_{\bar{x}}$ is defined in Lemma 33, such that $\mathrm{B}_{\|\cdot\|_2}(\bar{x}, r) \subset \mathsf{M}$. We consider such r, and we define $\mathsf{B} = \mathrm{B}_{\|\cdot\|_{\mathsf{E}}}(\bar{z}, r)$ and $\mathsf{B}_{\mathfrak{g}} = \mathrm{B}_{\|\cdot\|_{\bar{z}}}(\bar{z}, \bar{r})$, where $\bar{r} = C_{\bar{x}} r \leq 1/11$. In particular, we have by Lemma 30-(b) that $\mathsf{B}_{\mathfrak{g}} \subset \mathrm{B}_{\|\cdot\|_{\mathfrak{g}(\bar{x})}}(\bar{x}, \bar{r}) \times \mathrm{B}_{\|\cdot\|_{\mathfrak{g}(\bar{x})^{-1}}}(\bar{p}, \bar{r}) \subset \Omega$ since $\bar{r} < 1$. Moreover, using Lemma 33, we have $\mathsf{B} \subset \mathsf{B}_{\mathfrak{g}} \subset \Omega$.

Since $V \in \mathrm{C}^2(\mathsf{M}, \mathbb{R})$, ∇V is Lipschitz-continuous on $\mathsf{B}_{\mathfrak{g}}$, *i.e.*, there exists $C_V > 0$ such that for any $(z, z') \in \mathsf{B}_{\mathfrak{g}} \times \mathsf{B}_{\mathfrak{g}}$,

$$\|\nabla V(x) - \nabla V(x')\|_{\mathfrak{g}(\bar{x})^{-1}} \leq C_V \|x - x'\|_{\mathfrak{g}(\bar{x})} \ .$$

We first show that F is Lipschitz-continuous on $\mathsf{B}_{\mathfrak{g}}$ with respect to $\|\cdot\|_{\bar{z}}$. Consider $(z, z') \in \mathsf{B}_{\mathfrak{g}} \times \mathsf{B}_{\mathfrak{g}}$ denoted by $z = (x,p)$ and $z' = (x', p')$. Note that $(x, x') \in W^0(\bar{x}, 1) \times W^0(\bar{x}, 1)$, where $W^0(\bar{x}, 1)$ is defined in Lemma 30. We first bound $\|F^{(1)}(z) - F^{(1)}(z')\|_{\mathfrak{g}(\bar{x})} = \|\partial_p H(z) - \partial_p H(z')\|_{\mathfrak{g}(\bar{x})}$. We have

$$\partial_p H(z) - \partial_p H(z') = \mathfrak{g}(x)^{-1} p - \mathfrak{g}(x')^{-1} p' = \mathfrak{g}(x)^{-1}(p - p') + (\mathfrak{g}(x)^{-1} - \mathfrak{g}(x')^{-1}) p' \ ,$$

then,

$$\begin{aligned} \|\partial_p H(z) - \partial_p H(z')\|_{\mathfrak{g}(\bar{x})} \leq{}& \|\mathfrak{g}(\bar{x})^{1/2} \mathfrak{g}(x)^{-1} \mathfrak{g}(\bar{x})^{1/2}\|_2 \|p - p'\|_{\mathfrak{g}(\bar{x})^{-1}} \\ &+ \|\mathfrak{g}(\bar{x})^{1/2} (\mathfrak{g}(x)^{-1} - \mathfrak{g}(x')^{-1}) \mathfrak{g}(\bar{x})^{1/2}\|_2 \|p'\|_{\mathfrak{g}(\bar{x})^{-1}} \ . \end{aligned} \tag{A.13}$$

Using Lemma 30-(b), we have

$$(1 - \|\bar{x} - x\|_{\mathfrak{g}(x)})^2 \mathfrak{g}(\bar{x})^{-1} \preceq \mathfrak{g}(x)^{-1} \preceq (1 - \|\bar{x} - x\|_{\mathfrak{g}(x)})^{-2} \mathfrak{g}(\bar{x})^{-1} \ ,$$

and thus,

$$(1 - \|\bar{x} - x\|_{\mathfrak{g}(x)})^2 \mathrm{I}_d \preceq \mathfrak{g}(\bar{x})^{1/2} \mathfrak{g}(x)^{-1} \mathfrak{g}(\bar{x})^{1/2} \preceq (1 - \|\bar{x} - x\|_{\mathfrak{g}(x)})^{-2} \mathrm{I}_d \tag{A.14}$$

$$(1 - \bar{r})^2 \mathrm{I}_d \prec \mathfrak{g}(\bar{x})^{1/2} \mathfrak{g}(x)^{-1} \mathfrak{g}(\bar{x})^{1/2} \prec (1 - \bar{r})^{-2} \mathrm{I}_d \ . \tag{A.15}$$

Therefore, we have

$$\|\mathfrak{g}(\bar{x})^{1/2} \mathfrak{g}(x)^{-1} \mathfrak{g}(\bar{x})^{1/2}\|_2 \leq (1 - \bar{r})^{-2} \ . \tag{A.16}$$

In addition, we have

$$\begin{aligned} \|x - x'\|_{\mathfrak{g}(x)} &\leq (1 - \|\bar{x} - x'\|_{\mathfrak{g}(\bar{x})})^{-1} \|x - x\|_{\mathfrak{g}(\bar{x})} \\ &< (1 - \bar{r})^{-1} 2\bar{r} \\ &< 1/5, \end{aligned}$$

where we used Lemma 30-(c) in the first inequality and $\bar{r} \leq 1/11$ in the last inequality. Therefore, $x' \in \mathsf{W}^0(x, 1)$ and we have using Lemma 30-(b)

$$\begin{aligned} \{(1 - \|x' - x\|_{\mathfrak{g}(x)})^2 - 1\} \mathfrak{g}(x')^{-1} &\preceq \mathfrak{g}(x)^{-1} - \mathfrak{g}(x')^{-1} \\ &\preceq \{(1 - \|x' - x\|_{\mathfrak{g}(x)})^{-2} - 1\} \mathfrak{g}(x')^{-1} \\ \{(1 - \|x' - x\|_{\mathfrak{g}(x)})^2 - 1\} (1 - \bar{r})^2 \mathrm{I}_d &\preceq \mathfrak{g}(\bar{x})^{1/2} (\mathfrak{g}(x)^{-1} - \mathfrak{g}(x')^{-1}) \mathfrak{g}(\bar{x})^{1/2} \\ &\preceq \{(1 - \|x' - x\|_{\mathfrak{g}(x)})^{-2} - 1\} (1 - \bar{r})^{-2} \mathrm{I}_d \ . \end{aligned}$$

Then,

$$\begin{aligned}
&\|\mathfrak{g}(\bar{x})^{1/2}(\mathfrak{g}(x)^{-1}-\mathfrak{g}(x')^{-1})\mathfrak{g}(\bar{x})^{1/2}\|_2 \\
&\quad\le \max(|(1-\|x'-x\|_{\mathfrak{g}(x)})^2-1|(1-\bar{r})^2, \{(1-\|x'-x\|_{\mathfrak{g}(x)})^{-2}-1\}(1-\bar{r})^{-2}) \\
&\quad\le (1-\bar{r})^{-2}\max(|(1-\|x'-x\|_{\mathfrak{g}(x)})^2-1|, (1-\|x'-x\|_{\mathfrak{g}(x)})^{-2}-1)\ .
\end{aligned} \tag{A.17}$$

Using Inequalities (a) and (c) with $u=\|x'-x\|_{\mathfrak{g}(x)}$, where $u\le 1/5$ by (A.5.1), we obtain

(a) $|(1-\|x'-x\|_{\mathfrak{g}(x)})^2-1|\le 3\|x'-x\|_{\mathfrak{g}(x)} < 3(1-\bar{r})^{-1}\|x-x'\|_{\mathfrak{g}(\bar{x})}$,

(b) $(1-\|x'-x\|_{\mathfrak{g}(x)})^{-2}-1\le 3\|x'-x\|_{\mathfrak{g}(x)} < 3(1-\bar{r})^{-1}\|x-x'\|_{\mathfrak{g}(\bar{x})}$.

Moreover, we have

$$\|p'\|_{\mathfrak{g}(\bar{x})^{-1}} = \|p'-\bar{p}+\bar{p}\|_{\mathfrak{g}(\bar{x})^{-1}} \le \bar{r}+\|\bar{p}\|_{\mathfrak{g}(\bar{x})^{-1}}\ .$$

Combining (A.13), (A.16) and (A.17), it comes that

$$\|F^{(1)}(z)-F^{(1)}(z')\|_{\mathfrak{g}(\bar{x})} \le (1-\bar{r})^{-2}\|p-p'\|_{\mathfrak{g}(\bar{x})^{-1}} + 3(1-\bar{r})^{-3}(\bar{r}+\|\bar{p}\|_{\mathfrak{g}(\bar{x})^{-1}})\|x-x'\|_{\mathfrak{g}(\bar{x})}\ .$$

We define $A_{\bar{x},1}=(1-\bar{r})^{-2}+3(1-\bar{r})^{-3}(\bar{r}+\|\bar{p}\|_{\mathfrak{g}(\bar{x})^{-1}})$ such that $\|F^{(1)}(z)-F^{(1)}(z')\|_{\mathfrak{g}(\bar{x})}\le A_{\bar{x},1}\|z-z'\|_{\bar{z}}$ and we aim to bound $\|F^{(2)}(z)-F^{(2)}(x')\|_{\mathfrak{g}(\bar{x})^{-1}} = \|\partial_x H(z)-\partial_x H(z')\|_{\mathfrak{g}(\bar{x})^{-1}}$. We have

$$\begin{aligned}
\partial_x H(z)-\partial_x H(z') =& \nabla V(x)-\nabla V(x') \\
&+\tfrac{1}{2}D\mathfrak{g}(x')[\mathfrak{g}(x')^{-1}p',\mathfrak{g}(x')^{-1}p'] - \tfrac{1}{2}D\mathfrak{g}(x)[\mathfrak{g}(x)^{-1}p,\mathfrak{g}(x)^{-1}p] \\
&+\tfrac{1}{2}\mathfrak{g}(x)^{-1}:D\mathfrak{g}(x) - \tfrac{1}{2}\mathfrak{g}(x')^{-1}:D\mathfrak{g}(x')
\end{aligned} \tag{A.18}$$

We have

$$\begin{aligned}
&\|D\mathfrak{g}(x')[\mathfrak{g}(x')^{-1}p',\mathfrak{g}(x')^{-1}p'] - D\mathfrak{g}(x)[\mathfrak{g}(x)^{-1}p,\mathfrak{g}(x)^{-1}p]\|_{\mathfrak{g}(\bar{x})^{-1}} \\
&\quad\le \|D\mathfrak{g}(x)[\mathfrak{g}(x)^{-1}p,\mathfrak{g}(x)^{-1}p] - D\mathfrak{g}(x)[\mathfrak{g}(x')^{-1}p',\mathfrak{g}(x')^{-1}p']\|_{\mathfrak{g}(\bar{x})^{-1}} \\
&\qquad+\|D\mathfrak{g}(x)[\mathfrak{g}(x')^{-1}p',\mathfrak{g}(x')^{-1}p'] - D\mathfrak{g}(x')[\mathfrak{g}(x')^{-1}p',\mathfrak{g}(x')^{-1}p']\|_{\mathfrak{g}(\bar{x})^{-1}}
\end{aligned}$$

Note that we have $\mathfrak{g}(x)^{-1}p=F^{(1)}(z)$ (resp. $\mathfrak{g}(x')^{-1}p'=F^{(1)}(z')$). Using Lemma 30-(c), the first upper bound in (A.5.1) is bounded by

$$\begin{aligned}
&(1-\bar{r})^{-1}\|D\mathfrak{g}(x)[F^{(1)}(z),F^{(1)}(z)] - D\mathfrak{g}(x)[F^{(1)}(z'),F^{(1)}(z')]\|_{\mathfrak{g}(x)^{-1}} \\
&\quad\le 2(1-\bar{r})^{-1}\|D\mathfrak{g}(x)[F^{(1)}(z')-F^{(1)}(z),F^{(1)}(z')]\|_{\mathfrak{g}(x)^{-1}} \\
&\qquad+(1-\bar{r})^{-1}\|D\mathfrak{g}(x)[F^{(1)}(z')-F^{(1)}(z),F^{(1)}(z')-F^{(1)}(z)]\|_{\mathfrak{g}(x)^{-1}} \\
&\quad\le 4(1-\bar{r})^{-1}\|F^{(1)}(z')-F^{(1)}(z)\|_{\mathfrak{g}(x)}\|F^{(1)}(z')\|_{\mathfrak{g}(x)} \\
&\qquad+2(1-\bar{r})^{-1}\|F^{(1)}(z')-F^{(1)}(z)\|^2_{\mathfrak{g}(x)} && \text{(Lemma 30-(a))} \\
&\quad\le 4(1-\bar{r})^{-3}\|F^{(1)}(z')-F^{(1)}(z)\|_{\mathfrak{g}(\bar{x})}\|F^{(1)}(z')\|_{\mathfrak{g}(\bar{x})} \\
&\qquad+2(1-\bar{r})^{-3}\|F^{(1)}(z')-F^{(1)}(z)\|^2_{\mathfrak{g}(\bar{x})} && \text{(Lemma 30-(c))} \\
&\quad\le 4(1-\bar{r})^{-3}A_{\bar{x},1}\|z-z'\|_{\bar{z}}\|F^{(1)}(z')\|_{\mathfrak{g}(\bar{x})} + 2(1-\bar{r})^{-3}A^2_{\bar{x},1}\|z-z'\|^2_{\bar{z}} && \text{(see (A.5.1))} \\
&\quad\le \{4A_{\bar{x},1}(1-\bar{r})^{-5}(\bar{r}+\|\bar{p}\|_{\mathfrak{g}(\bar{x})^{-1}}) + 4(1-\bar{r})^{-3}A^2_{\bar{x},1}\bar{r}\}\|z-z'\|_{\bar{z}}\ ,
\end{aligned}$$

where we used that $\|F^{(1)}(z')\|_{\mathfrak{g}(\bar{x})}\le(1-\bar{r})^{-2}(\bar{r}+\|\bar{p}\|_{\mathfrak{g}(\bar{x})^{-1}})$ in the last inequality. Combining Lemma 30-(c) and Lemma 32-(c), the second upper bound in (A.5.1) is bounded by

$$\begin{aligned}
&(1-\bar{r})^{-1}\|D\mathfrak{g}(x)[F^{(1)}(z'),F^{(1)}(z')] - D\mathfrak{g}(x')[F^{(1)}(z'),F^{(1)}(z')]\|_{\mathfrak{g}(x)^{-1}} \\
&\quad\le \alpha(\alpha+1)(1-\bar{r})^{-1}/3\left\{(1-\|x'-x\|_{\mathfrak{g}(x)})^{-3}-1\right\}\|F^{(1)}(z')\|^2_{\mathfrak{g}(x)} \\
&\quad\le \alpha(\alpha+1)(1-\bar{r})^{-3}/3\left\{(1-(1-\bar{r})^{-1}\|x'-x\|_{\mathfrak{g}(\bar{x})})^{-3}-1\right\}\|F^{(1)}(z')\|^2_{\mathfrak{g}(\bar{x})} \\
&\quad\le \alpha(\alpha+1)(1-\bar{r})^{-7}(\bar{r}+\|\bar{p}\|_{\mathfrak{g}(\bar{x})^{-1}})^2/3\left\{(1-(1-\bar{r})^{-1}\|x'-x\|_{\mathfrak{g}(\bar{x})})^{-3}-1\right\}\ .
\end{aligned}$$

Using Inequality (a) with $u=(1-\bar{r})^{-1}\|x'-x\|_{\mathfrak{g}(\bar{x})}$, where $u\le 1/5$ by (A.5.1), we obtain

$$(1-(1-\bar{r})^{-1}\|x'-x\|_{\mathfrak{g}(\bar{x})})^{-3}-1 \le 5(1-\bar{r})^{-1}\|x'-x\|_{\mathfrak{g}(\bar{x})}\ .$$

Then, the second upper bound in (A.5.1) is bounded by

$$(5/3)\alpha(\alpha+1)(1-\bar{r})^{-8}(\bar{r}+\|\bar{p}\|_{\mathfrak{g}(\bar{x})^{-1}})^2\|x'-x\|_{\mathfrak{g}(\bar{x})} \ .$$

We now define $h : y \in \mathsf{M} \to \mathfrak{g}(y)^{-1} : D\mathfrak{g}(y)$. Since $\phi \in \mathrm{C}^4(\mathsf{M})$, $h \in \mathrm{C}^1(\mathsf{M})$. Moreover, $[x, x'] \in \mathsf{M}$ by convexity of M and we can define $A_\phi = \sup_{y\in[x,x']} \|Dh(y)\|$, where $\|Dh(y)\| = \sup_{\|u\|_{\mathfrak{g}(\bar{x})}=1} \|Dh(y)[u]\|_{\mathfrak{g}(\bar{x})^{-1}}$. Using the mean value theorem on h, we have

$$\|h(x)-h(x')\|_{\mathfrak{g}(\bar{x})^{-1}} \leq A_\phi\|x'-x\|_{\mathfrak{g}(\bar{x})} \ .$$

By combining (A.5.1), (A.18), (A.5.1) and (A.5.1), we have

$$\begin{aligned}
&\|F^{(2)}(z)-F^{(2)}(z')\|_{\mathfrak{g}(\bar{x})^{-1}} \\
&\quad\leq C_V\|x-x'\|_{\mathfrak{g}(\bar{x})} \\
&\qquad + (1/2)\{4A_{\bar{x},1}(1-\bar{r})^{-5}(\bar{r}+\|\bar{p}\|_{\mathfrak{g}(\bar{x})^{-1}}) + 4A_{\bar{x},1}^2(1-\bar{r})^{-3}\bar{r}\}\|z-z'\|_{\bar{z}} \\
&\qquad + (1/2)\cdot(5/3)\alpha(\alpha+1)(1-\bar{r})^{-8}(\bar{r}+\|\bar{p}\|_{\mathfrak{g}(\bar{x})^{-1}})^2\|x'-x\|_{\mathfrak{g}(\bar{x})} \\
&\qquad + (1/2)A_\phi\|x'-x\|_{\mathfrak{g}(\bar{x})} \ .
\end{aligned}$$

We define $A_{\bar{x},2} = C_V + 2A_{\bar{x},1}(1-\bar{r})^{-5}(\bar{r}+\|\bar{p}\|_{\mathfrak{g}(\bar{x})^{-1}}) + 2A_{\bar{x},1}^2(1-\bar{r})^{-3}\bar{r} + (5/6)\alpha(\alpha+1)(1-\bar{r})^{-8}(\bar{r}+\|\bar{p}\|_{\mathfrak{g}(\bar{x})^{-1}})^2 + A_\phi/2$ such that

$$\|F^{(2)}(z)-F^{(2)}(z')\|_{\mathfrak{g}(\bar{x})} \leq A_{\bar{x},2}\|z-z'\|_{\bar{z}} \ .$$

Finally, we have

$$\begin{aligned}
\|F(z)-F(z')\|_{\bar{z}} &= \|F^{(1)}(z)-F^{(1)}(z')\|_{\mathfrak{g}(\bar{x})} + \|F^{(2)}(z)-F^{(2)}(z')\|_{\mathfrak{g}(\bar{x})^{-1}} \\
&\leq (A_{\bar{x},1}+A_{\bar{x},2})\|z-z'\|_{\bar{z}} \ ,
\end{aligned}$$

i.e., F is $(A_{\bar{x},1}+A_{\bar{x},2})$-Lipschitz-continuous on $\mathsf{B}_\mathfrak{g}$ with respect to $\|\cdot\|_{\bar{z}}$.

We now show that F is Lipschitz-continuous on B with respect to $\|\cdot\|_\mathsf{E}$. Consider $(z, z') \in \mathsf{B}\times\mathsf{B}$ denoted by $z = (x, p)$ and $z' = (x', p')$. In particular, $(z, z') \in \mathsf{B}_\mathfrak{g}\times\mathsf{B}_\mathfrak{g}$. Using the previous result and Lemma 33, we have

$$\|F(z)-F(z')\|_\mathsf{E} \leq C_{\bar{x}}\|F(z)-F(z')\|_{\bar{z}} \leq C_{\bar{x}}(A_{\bar{x},1}+A_{\bar{x},2})\|z-z'\|_{\bar{z}} \leq C_{\bar{x}}^2(A_{\bar{x},1}+A_{\bar{x},2})\|z-z'\|_\mathsf{E} \ ,$$

which proves that F is $C_{\bar{x}}^2(A_{\bar{x},1}+A_{\bar{x},2})$-Lipschitz-continuous on B with respect to $\|\cdot\|_\mathsf{E}$.

Conclusion. Combining Items (a) and (b), we obtain the results (i), (ii) and (iii) of Proposition 34 upon using Cauchy-Lipschitz's theorem. Moreover, for any $t \in J_{z_0}$, $\partial_t H(z(t)) = \partial_x H(z(t))^\top\dot{x}(t) + \partial_p H(z(t))^\top\dot{p}(t) = -\dot{p}(t)^\top\dot{x}(t) + \dot{p}(t)^\top\dot{x}(t) = 0$, which proves the result (iv) of Proposition 34. □

Proof of Proposition 35. Assume **A**0, **A**1. Let $z_0 \in \mathrm{T}^\star\mathsf{M}$, (J_{z_0}, z) as in Proposition 34. Assume that $T_{z_0} = \sup J_{z_0} < +\infty$. Then, by Cauchy-Lipschitz's theorem, z leaves any compact set of E at the neighbourhood of T_{z_0}. By construction of $\|\cdot\|_\mathsf{E}$, this property is induced on both variables $x \in \mathsf{M}$ and $p \in \mathbb{R}^d$, each with respect to $\|\cdot\|_2$. We directly obtain the result of Proposition 35 by continuity of (x, p). □

A.5.2 Proof of reversibility in Algorithm 2

In (RM)HMC, one often sets the time horizon of the Hamiltonian dynamics with a hyperparameter h. However, under **A**0 and **A**1, we are not able to prove that the Hamiltonian dynamics (2.7) is defined at time $h > 0$, given *any* starting point $z_0 \in \mathrm{T}^\star\mathsf{M}$. Actually, this dynamics explodes if the time horizon is finite (see Proposition 35). This theoretical limitation requires to implement some sort of "involution checking step" (see Line 4 in Algorithm 2), which is theoretically explained below.

In the rest of this section, for any $z_0 \in \mathrm{T}^\star\mathsf{M}$, we will denote by $z(\cdot, z_0) \in \mathrm{C}^1(J_{z_0}, \mathrm{T}^\star\mathsf{M})$ the maximal solution to (2.7) with starting point z_0, uniquely defined in Proposition 34. For any $h > 0$, we recall below the definition of the sets E_h and $\bar{\mathsf{E}}_h$ the map T_h introduced in Appendix A.4:

(a) $\mathsf{E}_h = \{z_0 \in \mathrm{T}^\star\mathsf{M} \ : \ h < \sup J_{z_0}\}$,

(b) $\bar{\mathsf{E}}_h = \{z_0 \in \mathsf{E}_h \ : \ h < \sup J_{(s\circ \mathrm{T}_h)(z_0)}\}$,

(c) the map $\mathrm{T}_h : \mathsf{E}_h \to \mathrm{T}^\star\mathsf{M}$ is such that $\mathrm{T}_h(z_0) = z(h, z_0)$ for any $z_0 \in \mathsf{E}_h$.

In particular, we have

$$\bar{\mathsf{E}}_h = \mathsf{E}_h \cap (s \circ \mathrm{T}_h)^{-1}(\mathsf{E}_h) \ . \tag{A.19}$$

We first prove that the restriction of $s \circ \mathrm{T}_h$ to $\bar{\mathsf{E}}_h$ is an involutive integrator, which is crucial to derive the "reversibility" of Algorithm 2 in Theorem 36.

Lemma 37. *Assume* **A**0, **A**1. *Then, for any* $h > 0$, $s \circ \mathrm{T}_h$ *is a* C^1*-diffeomorphism on* $\bar{\mathsf{E}}_h$ *such that* $|\det \mathrm{Jac}[s \circ \mathrm{T}_h]| \equiv 1$.

Proof. Assume **A**0, **A**1. Let $h > 0$. It is clear that T_h (and thus $s \circ \mathrm{T}_h$) is well defined and continuously differentiable on E_h (in particular on $\bar{\mathsf{E}}_h$), by elaborating on the proof of Proposition 34 combined with Cauchy-Lipschitz's theorem. Let $z_0 \in \bar{\mathsf{E}}_h$. By definition of $\bar{\mathsf{E}}_h$, $\mathrm{T}_h(z_0)$, resp. J_{z_0}, and $(\mathrm{T}_h \circ s \circ \mathrm{T}_h)(z_0)$, resp. $J_{(s\circ \mathrm{T}_h)(z_0)}$, are well defined. We now aim to prove that $(\mathrm{T}_h \circ s \circ \mathrm{T}_h)(z_0) = s(z_0)$.

We define $z' : t \in [0, h] \mapsto s(z(h - t, z_0))$, which existence is straightforward since $(h - t) \in J_{z_0}$ for any $t \in [0, h]$. In particular, $z'(0) = (s \circ \mathrm{T}_h)(z_0)$ and z' is a solution of the ODE (2.7) on the interval $[0, h]$. Yet, $[0, h] \subset J_{(s\circ \mathrm{T}_h)(z_0)}$, and $z'(0) = z(0, (s \circ \mathrm{T}_h)(z_0))$. Then, by unicity of $z(\cdot, (s \circ \mathrm{T}_h)(z_0))$ (see Proposition 34), $z(\cdot, (s \circ \mathrm{T}_h)(z_0))$ and z' coincide on $[0, h]$. In particular, we have at time h

$$s(z_0) = z'(h) = z(h, (s \circ \mathrm{T}_h)(z_0)) = (\mathrm{T}_h \circ s \circ \mathrm{T}_h)(z_0) \ .$$

Then for any $z_0 \in \bar{\mathsf{E}}_h$, $(s \circ \mathrm{T}_h \circ s \circ \mathrm{T}_h)(z_0) = z_0$, *i.e.*, $s \circ \mathrm{T}_h$ is an involution on $\bar{\mathsf{E}}_h$. Since $s \circ \mathrm{T}_h \in \mathrm{C}^1(\bar{\mathsf{E}}_h, \mathrm{T}^\star\mathsf{M})$, $s \circ \mathrm{T}_h$ is a C^1-diffeomorphism on $\bar{\mathsf{E}}_h$ such that $|\det \mathrm{Jac}[s \circ \mathrm{T}_h]| \equiv 1$. □

We recall that we denote by $\mathrm{Q}_c : \mathrm{T}^\star\mathsf{M} \times \mathcal{B}(\mathrm{T}^\star\mathsf{M}) \to [0, 1]$, the transition kernel of the (homogeneous) Markov chain $(X_n, P_n)_{n\in[N]}$ obtained with Algorithm 2. We also denote by:

(a) $\mathrm{Q}_0 : \mathrm{T}^\star\mathsf{M} \times \mathcal{B}(\mathrm{T}^\star\mathsf{M}) \to [0, 1]$, the transition kernel referring to Step 1 (also Step 3) in Algorithm 2.

(b) $\mathrm{Q}_{c,1} : \mathrm{T}^\star\mathsf{M} \times \mathcal{B}(\mathrm{T}^\star\mathsf{M}) \to [0, 1]$, the transition kernel referring to Step 2 in Algorithm 2.

We provide below details on Markov kernels Q_0, $\mathrm{Q}_{c,1}$ and Q_c.

Kernel Q_0**.** This kernel corresponds to the Gaussian momentum update with refreshing rate β. For any $(z, z') \in \mathrm{T}^\star\mathsf{M} \times \mathrm{T}^\star\mathsf{M}$, we have

$$\begin{aligned} \mathrm{Q}_0(z, \mathrm{d}z') &= \mathrm{N}_x(p'; \sqrt{1-\beta}p, \beta \mathrm{Id})\mathrm{d}p'\delta_x(\mathrm{d}x') \\ &= (2\pi\beta)^{-d/2} \det(\mathfrak{g}(x))^{-1/2} \exp[-(2\beta)^{-1}\|p' - \sqrt{1-\beta}p\|^2_{\mathfrak{g}(x)^{-1}}]\mathrm{d}p'\delta_x(\mathrm{d}x') \ . \end{aligned} \tag{A.20}$$

Kernel $\mathrm{Q}_{c,1}$**.** This kernel is deterministic and corresponds to the *exact* integration of the Hamiltonian up until time h. For any $(z, z') \in \mathrm{T}^\star\mathsf{M} \times \mathrm{T}^\star\mathsf{M}$, we have

$$\mathrm{Q}_{c,1}(z, \mathrm{d}z') = \mathbb{1}_{\bar{\mathsf{E}}_h}(z)\delta_{\mathrm{T}_h(z)}(\mathrm{d}z') + \mathbb{1}_{\bar{\mathsf{E}}_h^c}(z)\delta_{s(z)}(\mathrm{d}z') \ . \tag{A.21}$$

Kernel Q_c**.** This kernel corresponds to one step of Algorithm 2 (*i.e.*, comprising Steps 1 to 3). For any $(z, z') \in \mathrm{T}^\star\mathsf{M} \times \mathrm{T}^\star\mathsf{M}$, we have

$$\mathrm{Q}_c(z, \mathrm{d}z') = \int_{\mathrm{T}^\star\mathsf{M}\times\mathrm{T}^\star\mathsf{M}} \mathrm{Q}_0(z, \mathrm{d}z_1)\mathrm{Q}_{c,1}(z_1, \mathrm{d}z_2)\mathrm{Q}_0(z_2, \mathrm{d}z') \ . \tag{A.22}$$

We recall that $\bar{\pi}$, as defined in (2.5), admits a density with respect to the product Lebesgue measure given for any $z = (x, p) \in \mathrm{T}^\star\mathsf{M}$ by

$$(\mathrm{d}\bar{\pi}/(\mathrm{d}x\mathrm{d}p))(x, p) = (1/Z) \exp[-(1/2)\|p\|^2_{\mathfrak{g}(x)^{-1}}] \det(\mathfrak{g}(x))^{-1/2} \exp[-V(x)] \ .$$

Lemma 38. *Assume* **A**0. *Then, the Markov kernel* Q_0*, defined in* (A.20)*, is reversible (up to momentum reversal) with respect to* $\bar{\pi}$.

Proof. Assume **A**0. For any $(z, z') \in \mathrm{T}^\star\mathsf{M} \times \mathrm{T}^\star\mathsf{M}$, we have

$$\begin{aligned}
\mathrm{Q}_0(z, \mathrm{d}z')\bar{\pi}(\mathrm{d}z) &= (1/Z)(2\pi)^{-d/2}\beta^{-d/2}\det(\mathfrak{g}(x))^{-1}\exp[-V(x)]\mathrm{d}x\mathrm{d}p\mathrm{d}p'\delta_x(\mathrm{d}x') \\
&\quad \exp[-(2\beta)^{-1}\|p' - \sqrt{1-\beta}p\|^2_{\mathfrak{g}(x)^{-1}}]\exp[-(1/2)\|p\|^2_{\mathfrak{g}(x)^{-1}}] \\
&= (1/Z)(2\pi)^{-d/2}\beta^{-d/2}\det(\mathfrak{g}(x))^{-1}\exp[-V(x)]\mathrm{d}x\mathrm{d}p\mathrm{d}p'\delta_x(\mathrm{d}x') \\
&\quad \exp[-(2\beta)^{-1}\{\|p\|^2_{g(x)^{-1}} - 2\sqrt{1-\beta}\langle p', p\rangle_{g(x)^{-1}} + \|p'\|^2_{g(x)^{-1}}\}] \\
&= (1/Z)(2\pi)^{-d/2}\beta^{-d/2}\det(\mathfrak{g}(x'))^{-1}\exp[-V(x')]\mathrm{d}x'\mathrm{d}p'\mathrm{d}p\delta_x(\mathrm{d}x) \\
&\quad \exp[-(2\beta)^{-1}\{\|p'\|^2_{g(x')^{-1}} - 2\sqrt{1-\beta}\langle p, p'\rangle_{g(x')^{-1}} + \|p\|^2_{g(x')^{-1}}\}] \\
&= \mathrm{Q}_0(z', \mathrm{d}z)\bar{\pi}(\mathrm{d}z') \,,
\end{aligned}$$

which proves the reversibility of Q_0 with respect to $\bar{\pi}$. Since $s_\# \mathrm{Q}_0 = \mathrm{Q}_0$, we conclude the proof with Lemma 27. □

Lemma 39. *Assume* **A**0, **A**1. *Then, for any $h > 0$, the Markov kernel $\mathrm{Q}_{c,1}$ with step-size h, defined in (A.21), is reversible up to momentum reversal with respect to $\bar{\pi}$.*

Proof. Assume **A**0, **A**1. We recall that the set $\bar{\mathsf{E}}_h$ is defined in (A.19). Let $f \in \mathrm{C}(\mathrm{T}^\star\mathsf{M} \times \mathrm{T}^\star\mathsf{M}, \mathbb{R})$ with compact support. According to Definition 25, we aim to show that

$$\int_{\mathrm{T}^\star\mathsf{M}\times\mathrm{T}^\star\mathsf{M}} f(z, z')\mathrm{Q}_{c,1}(z, \mathrm{d}z')\bar{\pi}(\mathrm{d}z) = \int_{\mathrm{T}^\star\mathsf{M}\times\mathrm{T}^\star\mathsf{M}} f(s(z'), s(z))\mathrm{Q}_{c,1}(z, \mathrm{d}z')\bar{\pi}(\mathrm{d}z) \,. \tag{A.23}$$

We denote by I the left integral of (A.23). We have $I = I_1 + I_2$ where

$$I_1 = \int_{\bar{\mathsf{E}}_h} \bar{\pi}(z) f(z, \mathrm{T}_h(z))\mathrm{d}z \,, \qquad I_2 = \int_{\bar{\mathsf{E}}_h^{\mathrm{c}}} \bar{\pi}(z) f(z, s(z))\mathrm{d}z \,.$$

We denote by J the right integral of (A.23). By symmetry, we have $J = J_1 + J_2$ where

$$J_1 = \int_{\bar{\mathsf{E}}_h} \bar{\pi}(z) f((s \circ \mathrm{T}_h)(z), s(z))\mathrm{d}z \,, \qquad J_2 = \int_{\bar{\mathsf{E}}_h^{\mathrm{c}}} \bar{\pi}(z) f(z, s(z))\mathrm{d}z \,.$$

We directly have $I_2 = J_2$. Let us now prove that $I_1 = J_1$. By change of variable $z \mapsto (s \circ \mathrm{T}_h)(z)$ in I_1, we have

$$\begin{aligned}
I_1 &= \int_{\bar{\mathsf{E}}_h} \bar{\pi}(z) f(z, \mathrm{T}_h(z))\mathrm{d}z \\
&= \int_{(s\circ\mathrm{T}_h)(\bar{\mathsf{E}}_h)} \bar{\pi}((s \circ \mathrm{T}_h)(z)) f((s \circ \mathrm{T}_h)(z), s(z))\mathrm{d}z && \text{(Lemma 37)} \\
&= \int_{\bar{\mathsf{E}}_h} \bar{\pi}(z) f((s \circ \mathrm{T}_h)(z), s(z))\mathrm{d}z && \text{(Proposition 34-(iv) \& } (s \circ \mathrm{T}_h)(\bar{\mathsf{E}}_h) = \bar{\mathsf{E}}_h) \\
&= J_1 \,.
\end{aligned}$$

Finally, we obtain $I = J$ and thus prove (A.23) for any continuous function with compact support, which concludes the proof.

□

We are now ready to prove Theorem 36, which states that Q_c is reversible up to momentum reversal with respect to $\bar{\pi}$.

Proof of Theorem 36. Assume **A**0, **A**1. Let $f : \mathrm{T}^\star\mathsf{M} \times \mathrm{T}^\star\mathsf{M} \to \mathbb{R}$ be a continuous function with compact support. We have

$$\begin{aligned}
&\int_{\mathrm{T}^\star\mathsf{M}\times\mathrm{T}^\star\mathsf{M}} f(z, z')\mathrm{Q}_c(z, \mathrm{d}z')\bar{\pi}(\mathrm{d}z) \\
&\quad = \int_{(\mathrm{T}^\star\mathsf{M})^4} f(z, z')\mathrm{Q}_0(z, \mathrm{d}z_1)\mathrm{Q}_{c,1}(z_1, \mathrm{d}z_2)\mathrm{Q}_0(z_2, \mathrm{d}z')\bar{\pi}(\mathrm{d}z) && \text{(see (A.22))} \\
&\quad = \int_{(\mathrm{T}^\star\mathsf{M})^4} f(s(z_1), z')\mathrm{Q}_0(z, \mathrm{d}z_1)\mathrm{Q}_{c,1}(s(z), \mathrm{d}z_2)\mathrm{Q}_0(z_2, \mathrm{d}z')\bar{\pi}(\mathrm{d}z) && \text{(Lemma 38)} \\
&\quad = \int_{(\mathrm{T}^\star\mathsf{M})^4} f(s(z_1), z')\mathrm{Q}_0(s(z), \mathrm{d}z_1)\mathrm{Q}_{c,1}(z, \mathrm{d}z_2)\mathrm{Q}_0(z_2, \mathrm{d}z')\bar{\pi}(\mathrm{d}z) && \text{(momentum reversal on } z) \\
&\quad = \int_{(\mathrm{T}^\star\mathsf{M})^4} f(s(z_1), z')\mathrm{Q}_0(z_2, \mathrm{d}z_1)\mathrm{Q}_{c,1}(z, \mathrm{d}z_2)\mathrm{Q}_0(s(z), \mathrm{d}z')\bar{\pi}(\mathrm{d}z) && \text{(Lemma 39)} \\
&\quad = \int_{(\mathrm{T}^\star\mathsf{M})^4} f(s(z_1), z')\mathrm{Q}_0(z_2, \mathrm{d}z_1)\mathrm{Q}_{c,1}(s(z), \mathrm{d}z_2)\mathrm{Q}_0(z, \mathrm{d}z')\bar{\pi}(\mathrm{d}z) && \text{(momentum reversal on } z) \\
&\quad = \int_{(\mathrm{T}^\star\mathsf{M})^4} f(s(z_1), s(z))\mathrm{Q}_0(z_2, \mathrm{d}z_1)\mathrm{Q}_{c,1}(z', \mathrm{d}z_2)\mathrm{Q}_0(z, \mathrm{d}z')\bar{\pi}(\mathrm{d}z) && \text{(Lemma 38)} \\
&\quad = \int_{\mathrm{T}^\star\mathsf{M}\times\mathrm{T}^\star\mathsf{M}} f(s(z'), s(z))\mathrm{Q}_c(z, \mathrm{d}z')\bar{\pi}(\mathrm{d}z) \,.
\end{aligned}$$

Moreover, $s_\#\bar{\pi} = \bar{\pi}$. Hence, by combining Definition 25 and Lemma 26, we obtain the result of Theorem 36. □

A.6 Numerical integration in HMC

In this section, we first recall the definition of symplectic maps, and then discuss the choice of integrators in RMHMC. Finally, we focus on the implicit integrator defined in (2.14) and provide some notations, which will be notably used in Appendix A.7.

Reminders on symplecticity. We define $J_d = \begin{pmatrix} 0 & \mathrm{I}_d \\ -\mathrm{I}_d & 0 \end{pmatrix} \in \mathbb{R}^{2d\times 2d}$.

Definition 40 (Hairer et al. [2006]). *A linear mapping $A : \mathbb{R}^{2d} \to \mathbb{R}^{2d}$ is called symplectic if $A^\top J_d A = J_d$.*

Definition 41 (Hairer et al. [2006]). *A differentiable map $F : \mathsf{U} \to \mathbb{R}^{2d}$, where $\mathsf{U} \subset \mathbb{R}^{2d}$ is an open set, is called symplectic if the Jacobian matrix $\mathrm{Jac}[F]$ is symplectic, i.e., if $\mathrm{Jac}[F](z)^\top J_d \,\mathrm{Jac}[F](z) = J_d$ for any $z \in \mathsf{U}$. In particular, if F is symplectic, then $|\det \mathrm{Jac}[F]| \equiv 1$.*

Choice of the integrators in RMHMC. Introducing a Riemannian metric $\mathfrak{g}$ in the Hamiltonian *a priori* makes it non separable. Thus, designing an integrator which is (i) symplectic, (ii) reversible and (iii) not too computationally heavy is challenging. The *generalized Leapfrog* integrator (GLI), or equivalently the Störmer-Verlet integrator, combined with fixed point iterations, as chosen by Girolami and Calderhead [2011], is considered as the standard scheme for RMHMC. Brofos and Lederman [2021b] analyze the impact of GLI on the ergodicity of RMHMC under a metric which is designed in the same manner as Cobb et al. [2019]. In particular, they show that the convergence threshold used to compute the fixed-point process only matters to an extent, and identify a diminishing return to using smaller convergence thresholds. They compare the fixed-point approach with Newton's method, which gives less iterations to converge. On the other hand, the *implicit midpoint* integrator (IMI) enjoys the same theoretical properties as GLI, when these two integrators are combined with fixed-point iterations. As shown by Brofos and Lederman [2021a], IMI may also show numerical advantages in terms of reversibility and volume preservation for specific target distributions. However, this integrator is hard to use for reversibility proofs due to its non-separability. Besides this, Cobb et al. [2019] combine a 2-state augmented Hamiltonian, based on the work of Tao [2016], with Strang splitting [Strang, 1968] to propose a symplectic *explicit* integrator which thus has the advantage not to rely on fixed-point iterations. Yet, this integrator does not satisfy reversibility in theory.

Details on the Störmer-Verlet integrator. The scheme defined in (2.14) can be written as $\mathrm{G}_h = \overline{\mathrm{G}}_h \circ \underline{\mathrm{G}}_h$ where:

(a) $\underline{\mathrm{G}}_h : \mathrm{T}^\star\mathsf{M} \to 2^{\mathrm{T}^\star\mathsf{M}}$ is the set-valued map *implicitly* defined by the first two aligns of (2.14), *i.e.*, for any $(z, z') \in \mathrm{T}^\star\mathsf{M} \times \mathrm{T}^\star\mathsf{M}$, $z' \in \underline{\mathrm{G}}_h(z)$ if and only if

$$p' = p - \tfrac{h}{2}\partial_x H_2(x, p'), \qquad x' = x + \tfrac{h}{2}[\partial_p H_2(x, p') + \partial_p H_2(x', p')] \,, \tag{A.24}$$

We also define the map $\underline{\mathrm{g}}_h : \mathrm{T}^\star\mathsf{M} \times \mathrm{T}^\star\mathsf{M} \to \mathbb{R}^n \times \mathbb{R}^n$ by

$$\underline{\mathrm{g}}_h(z, z') = (x' - x - \tfrac{h}{2}[\partial_p H_2(x, p') + \partial_p H_2(x', p')], p' - p + \tfrac{h}{2}\partial_x H_2(x, p')) \,. \tag{A.25}$$

such that $\underline{\mathrm{g}}_h(z, z') = 0$ if and only if $z' \in \underline{\mathrm{G}}_h(z)$.

(b) $\overline{\mathrm{G}}_h : \mathrm{T}^\star\mathsf{M} \to \mathrm{T}^\star\mathsf{M}$ is *explicitly* defined by the last align of (2.14), *i.e.*, for any $z \in \mathrm{T}^\star\mathsf{M}$,

$$\overline{\mathrm{G}}_h(z) = (x, p - \tfrac{h}{2}\partial_x H_2(x, p)) \,. \tag{A.26}$$

A.7 Proofs of Section 2.4.1

In this section, we state several results on the maps defined by the implicit integrators of the Hamiltonian (see Section 2.3.1), in order to prove Proposition 2.

Lemma 42. *Let $\mathsf{U} \subset \mathbb{R}^d$ a non-empty open set and $z \in \mathsf{U}$. Assume there exist a neighbourhood $\mathsf{U}_z \subset \mathsf{U}$ of z, $\mathtt{L} > 0$ and a $\mathtt{L}$-Lipschitz-continuous map $\psi : \mathsf{U}_z \to \mathbb{R}^d$ such that $\psi \in \mathrm{C}^1(\mathsf{U}_z, \mathbb{R}^d)$. Then, for any $h \in (0, 1/\mathtt{L})$ and any $a \in \mathbb{R}^d$, the map $\psi_h : \mathsf{U}_z \to \mathbb{R}^d$ defined for any $z' \in \mathsf{U}_z$ by $\psi_h(z') = z' + h\psi(z') + a$ is a C^1-diffeomorphism on a neighbourhood $\mathsf{U}'_z \subset \mathsf{U}_z$ of z.*

Proof. Assume we are provided with $\mathsf{U}, z, \mathsf{U}_z, \mathtt{L}$ and ψ as described in Lemma 42. Let $h \in (0, 1/\mathtt{L})$ and $a \in \mathbb{R}^d$. We define the map $\psi_h : \mathsf{U}_z \to \mathbb{R}^d$ by $\psi_h(z') = z' + h\psi(z') + a$, for any $z' \in \mathsf{U}_z$. Since $\psi \in \mathrm{C}^1(\mathsf{U}_z, \mathbb{R}^d)$, it is clear that $\psi_h \in \mathrm{C}^1(\mathsf{U}_z, \mathbb{R}^d)$. We have, for any $z' \in \mathsf{U}_z$, $D\psi_h(z') = \mathrm{Id} + hD\psi(z')$. In particular, for any $w \in \mathbb{R}^d$, it comes that

$$\|D\psi_h(z)w\| \geq \|w\| - h\mathtt{L}\|w\| = (1 - hL)\|w\| \; .$$

Since $1 - h\mathtt{L} > 0$, $\mathrm{Jac}[\psi_h](z)$ is invertible. We conclude the proof by using the local inverse function theorem. □

We recall that the set-valued map G_h, defined in (2.14), is the *generalised Leapfrog* integrator, or equivalently the *Störmer-Verlet* integrator, of the non-separable Hamiltonian H_2 defined in Section 2.3.1. We state below a result of local smoothness of this integrator.

Lemma 43. *Let $(\mathsf{M}, \mathfrak{g})$ be a a smooth manifold of $\mathbb{R}^d$ and $h > 0$. Then, for any $(x^{(0)}, x^{(1)}) \in \mathsf{M} \times \mathsf{M}$, the vector $p^{(0)} \in \mathrm{T}^\star_{x^{(0)}}\mathsf{M}$ such that $(x^{(1)}, p^{(1)}) \in \mathrm{G}_h(x^{(0)}, p^{(0)})$ for any $p^{(1)} \in \mathrm{T}^\star_{x^{(1)}}\mathsf{M}$ is uniquely determined by $p^{(0)} = \mathrm{G}_{h,x^{(0)}}(x^{(1)})$, where the map $\mathrm{G}_{h,x^{(0)}} : \mathsf{M} \to \mathrm{T}^\star_{x^{(0)}}\mathsf{M}$ is defined by*

$$\mathrm{G}_{h,x^{(0)}}(y) = p^{(1/2)}(y) - \frac{h}{4} D\mathfrak{g}(x^{(0)})[\mathfrak{g}(x^{(0)})^{-1}p^{(1/2)}(y), \mathfrak{g}(x^{(0)})^{-1}p^{(1/2)}(y)] \; , \tag{A.27}$$

with $p^{(1/2)}(y) = \frac{2}{h}(\mathfrak{g}(x^{(0)})^{-1} + \mathfrak{g}(y)^{-1})^{-1}(y - x^{(0)})$. In particular, if $\mathfrak{g} \in \mathrm{C}^2(\mathsf{M}, \mathbb{R}^{d\times d})$, then, for any $x^{(0)} \in \mathsf{M}$, $\mathrm{G}_{h,x^{(0)}} \in \mathrm{C}^1(\mathsf{M}, \mathrm{T}^\star_{x^{(0)}}\mathsf{M})$.

For any $\alpha \geq 1$, we define $r^\star_\alpha > 0$ by

$$r^\star_\alpha = \min(1/50000, 1/\{1000\alpha(\alpha + 1)\}) \; . \tag{A.28}$$

*We recall that we denote by $\mathsf{W}^0(x, r)$ the open Dikin ellipsoid (w.r.t. $\mathfrak{g}$) at $x \in \mathsf{M}$ of radius $r > 0$, given by $\mathsf{W}^0(x, r) = \{y \in \mathbb{R}^d : \|y - x\|_{\mathfrak{g}(x)} < r\}$. Assume **A**0, **A**1. Let $x^{(0)} \in \mathsf{M}$ and $r \in (0, r^\star_\alpha]$, where α is given by **A**1. Then, for any $x \in \mathsf{W}^0(x^{(0)}, r)$, $x \in \mathsf{M}$ and $\mathrm{Jac}[\mathrm{G}_{h,x^{(0)}}](x)$ is invertible for any $h > 0$.*

Proof. Let $(\mathsf{M}, \mathfrak{g})$ be a a smooth manifold of $\mathbb{R}^d$ and $h > 0$. We first prove the existence and uniqueness of the map defined in (A.27). Let $(x^{(0)}, x^{(1)}) \in \mathsf{M} \times \mathsf{M}$. We define for any $y \in \mathsf{M}$

$$p^{(1/2)}(y) = \frac{2}{h}(\mathfrak{g}(x^{(0)})^{-1} + \mathfrak{g}(y)^{-1})^{-1}(y - x^{(0)}) \; , \qquad \tilde{p}(y) = p^{(1/2)}(y) + \frac{h}{2}\partial_x H_2(x^{(0)}, p^{(1/2)}(y)) \; .$$

Note that these expressions are obtained by simply inverting the first two aligns of (2.14). Then, by definition of G_h, the vector $p \in \mathrm{T}^\star_{x^{(0)}}\mathsf{M}$ such that $(x^{(1)}, p^{(1)}) \in \mathrm{G}_h(x^{(0)}, p)$ for any $p^{(1)} \in \mathrm{T}^\star_{x^{(1)}}\mathsf{M}$ is uniquely determined by $p = \tilde{p}(x^{(1)})$. We thus obtain (A.27), using the expression of $\partial_x H_2$ given in Section 2.3.1. Moreover, it is clear that $\mathrm{G}_{h,x^{(0)}}$ is continuously differentiable if $\mathfrak{g} \in \mathrm{C}^2(\mathsf{M}, \mathbb{R}^{d\times d})$.

We are now going to prove the second result of Lemma 43. Assume **A**0, **A**1. Let $x^{(0)} \in \mathsf{M}$ and $r \in (0, r^\star_\alpha]$, where $r^\star_\alpha$ is defined in (A.28) and α is given by **A**1. In particular $r^\star_\alpha \leq 1/11$. For the sake of clarity, we will now denote $\mathfrak{g}(x^{(0)})$ by $\mathfrak{g}_0$ for the rest of the proof.

We define the open Dikin ellipsoid $\mathsf{B}_0 = \mathsf{W}^0(x^{(0)}, r)$. Since $r < 1$, $\mathsf{B}_0 \subset \mathsf{M}$ by Lemma 30-(b). Let $h > 0$. We now define the following maps on B_0

$$\mathrm{M}_0 = \left\{ \begin{array}{ll} \mathsf{B}_0 & \to S_d^{++}(\mathbb{R}) \\ y & \mapsto \mathfrak{g}_0^{-1} + \mathfrak{g}(y)^{-1} \end{array} \right. , \qquad \mathrm{K}_0 = \left\{ \begin{array}{ll} \mathsf{B}_0 & \to \mathbb{R}^d \\ y & \mapsto \mathfrak{g}_0^{-1}\mathrm{M}_0(y)^{-1}(y - x^{(0)}) \end{array} \right. ,$$

$$\phi_0 = \left\{ \begin{array}{ll} \mathsf{B}_0 & \to \mathbb{R}^d \\ y & \mapsto \mathrm{M}_0(y)D\mathfrak{g}_0[\mathrm{K}_0(y), \mathrm{K}_0(y)] \end{array} \right. , \qquad \tilde{\phi}_0 = \left\{ \begin{array}{ll} \mathsf{B}_0 & \to \mathbb{R}^d \\ y & \mapsto \frac{h}{2}\mathrm{M}_0(y)\mathrm{G}_{h,x^{(0)}}(y) \end{array} \right. .$$

Note that we have, for any $y \in \mathsf{B}_0$, $\tilde{\phi}(y) = (y - x^{(0)}) - \frac{1}{2}\phi_0(y)$. Since $\mathfrak{g}$ is twice continuously differentiable on B_0, the maps previously defined are continuously differentiable. We now compute the derivatives of these maps.

Let $y \in \mathsf{B}_0$ and $w \in \mathbb{R}^d$. We have

$$\begin{aligned}
D\mathrm{M}_0(y)[w] &= -\mathfrak{g}(y)^{-1} D\mathfrak{g}(y)[w]\mathfrak{g}(y)^{-1} \,, \\
D\mathrm{K}_0(y)[w] &= -\mathfrak{g}_0^{-1}\mathrm{M}_0(y)^{-1} D\mathrm{M}_0(y)[w]\mathrm{M}_0(y)^{-1}(y - x^{(0)}) + \mathfrak{g}_0^{-1}\mathrm{M}_0(y)^{-1} w \\
&= -\mathfrak{g}_0^{-1}\{\mathfrak{g}(y)\mathfrak{g}_0^{-1} + \mathrm{I}_d\}^{-1}\mathfrak{g}(y) D\mathrm{M}_0(y)[w]\mathfrak{g}(y)\{\mathfrak{g}_0^{-1}\mathfrak{g}(y) + \mathrm{I}_d\}^{-1}(y - x^{(0)}) \\
&\quad + \mathfrak{g}_0^{-1}\mathrm{M}_0(y)^{-1} w \\
&= \mathfrak{g}_0^{-1}\{\mathfrak{g}(y)\mathfrak{g}_0^{-1} + \mathrm{I}_d\}^{-1} D\mathfrak{g}(y)[w]\{\mathfrak{g}_0^{-1}\mathfrak{g}(y) + \mathrm{I}_d\}^{-1}(y - x^{(0)}) + \mathfrak{g}_0^{-1}\mathrm{M}_0(y)^{-1} w \,, \\
D\phi_0(y)[w] &= D\mathrm{M}_0(y)[w] D\mathfrak{g}_0[\mathrm{K}_0(y), \mathrm{K}_0(y)] + \mathrm{M}_0(y) D\mathfrak{g}_0[D\mathrm{K}_0(y)[w], \mathrm{K}_0(y)] \\
&\quad + \mathrm{M}_0(y) D\mathfrak{g}_0[\mathrm{K}_0(y), D\mathrm{K}_0(y)[w]] \,, \\
D\tilde{\phi}_0(y)[w] &= \mathrm{I}_d - \frac{1}{2} D\phi_0(y)[w] \,.
\end{aligned}$$

We have in particular $\tilde{\phi}_0(x^{(0)}) = 0$ and $D\phi_0(x^{(0)}) = 0$. Let us first study the smoothness of $D\phi_0$ on B_0.

Smoothness of $D\phi_0$. We prove here that $D\phi_0$ is $(r_\alpha^\star)^{-1}$-Lipschitz-continuous on B_0 with respect to $\|\cdot\|_{\mathfrak{g}_0}$. Let $(y, y') \in \mathsf{B}_0$ and $w \in \mathbb{R}^d$. We have

$$\begin{aligned}
&\|D\phi_0(y)[w] - D\phi_0(y')[w]\|_{\mathfrak{g}_0} \\
&\quad \le \|D\mathrm{M}_0(y)[w] D\mathfrak{g}_0[\mathrm{K}_0(y), \mathrm{K}_0(y)] - D\mathrm{M}_0(y')[w] D\mathfrak{g}_0[\mathrm{K}_0(y'), \mathrm{K}_0(y')]\|_{\mathfrak{g}_0} \\
&\qquad + \|\mathrm{M}_0(y) D\mathfrak{g}_0[D\mathrm{K}_0(y)[w], \mathrm{K}_0(y)] - \mathrm{M}_0(y') D\mathfrak{g}_0[D\mathrm{K}_0(y')[w], \mathrm{K}_0(y')]\|_{\mathfrak{g}_0} \\
&\qquad + \|\mathrm{M}_0(y) D\mathfrak{g}_0[\mathrm{K}_0(y), D\mathrm{K}_0(y)[w]] - \mathrm{M}_0(y') D\mathfrak{g}_0[\mathrm{K}_0(y'), D\mathrm{K}_0(y')[w]]\|_{\mathfrak{g}_0} \,.
\end{aligned} \tag{A.29}$$

Remark that the two last terms in (A.29) are very similar and can be bounded in the same way. The rest of the proof on the Lipschitz-continuity of $D\phi_0$ is separated in two steps, each of them consisting of bounding from above a term of (A.29).

Step 1. Let us first bound the first term in (A.29). For any $x \in \mathsf{B}_0$, we denote $D\mathrm{M}_0(x)[w] D\mathfrak{g}_0[\mathrm{K}_0(x), \mathrm{K}_0(x)]$ by $a_1(x)$. We have

$$\begin{aligned}
&\mathfrak{g}_0^{1/2}(a_1(y) - a_1(y')) \\
&\quad = \{\mathfrak{g}_0^{1/2}(D\mathrm{M}_0(y)[w] - D\mathrm{M}_0(y')[w])\mathfrak{g}_0^{1/2}\}\{\mathfrak{g}_0^{-1/2} D\mathfrak{g}_0[\mathrm{K}_0(y), \mathrm{K}_0(y)]\} \\
&\qquad + \{\mathfrak{g}_0^{1/2} D\mathrm{M}_0(y')[w]\mathfrak{g}_0^{1/2}\}\{\mathfrak{g}_0^{-1/2}(D\mathfrak{g}_0[\mathrm{K}_0(y), \mathrm{K}_0(y)] - D\mathfrak{g}_0[\mathrm{K}_0(y'), \mathrm{K}_0(y')])\} \,.
\end{aligned} \tag{A.30}$$

We now aim to bound each one of the four terms that appear in (A.30).

Step 1.1. First, we have

$$\begin{aligned}
\mathfrak{g}_0^{1/2}(D\mathrm{M}_0(y)[w] - D\mathrm{M}_0(y')[w])\mathfrak{g}_0^{1/2} &= \mathfrak{g}_0^{1/2}(\mathfrak{g}(y')^{-1} D\mathfrak{g}(y')[w]\mathfrak{g}(y')^{-1} - \mathfrak{g}(y)^{-1} D\mathfrak{g}(y)[w]\mathfrak{g}(y)^{-1})\mathfrak{g}_0^{1/2} \\
&= \mathfrak{g}_0^{1/2}\{\mathfrak{g}(y')^{-1} - \mathfrak{g}(y)^{-1}\} D\mathfrak{g}(y')[w]\mathfrak{g}(y')^{-1}\mathfrak{g}_0^{1/2} \\
&\quad + \mathfrak{g}_0^{1/2}\mathfrak{g}(y)^{-1}\{D\mathfrak{g}(y')[w] - D\mathfrak{g}(y)[w]\}\mathfrak{g}(y')^{-1}\mathfrak{g}_0^{1/2} \\
&\quad + \mathfrak{g}_0^{1/2}\mathfrak{g}(y)^{-1} D\mathfrak{g}(y)[w]\{\mathfrak{g}(y')^{-1} - \mathfrak{g}(y)^{-1}\}\mathfrak{g}_0^{1/2} \,.
\end{aligned}$$

We recall that $r \le r_\alpha^\star \le 1/11$, and thus, we have for any $x \in \{y, y'\}$ the following inequalities

$$\begin{aligned}
\|\mathfrak{g}_0^{1/2}\{\mathfrak{g}(y')^{-1} - \mathfrak{g}(y)^{-1}\}\mathfrak{g}_0^{1/2}\|_2 &\le 3(1-r)^{-3}\|y - y'\|_{\mathfrak{g}_0} \,, && \text{(on the model of (A.17))} \\
\|\mathfrak{g}_0^{1/2}\mathfrak{g}(x)^{-1}\mathfrak{g}_0^{1/2}\|_2 &\le (1-r)^{-2} \,, && \text{(Lemma 30-(b))} \\
\|\mathfrak{g}_0^{-1/2} D\mathfrak{g}(x)[w]\mathfrak{g}_0^{-1/2}\|_2 &\le (1-r)^{-2}\|\mathfrak{g}(x)^{-1/2} D\mathfrak{g}(x)[w]\mathfrak{g}(x)^{-1/2}\|_2 \,.
\end{aligned}$$

In particular, we have for any $x \in \{y, y'\}$

$$\begin{aligned}
\|\mathfrak{g}(x)^{-1/2} D\mathfrak{g}(x)[w]\mathfrak{g}(x)^{-1/2}\|_2 &= \sup_{\{w' \in \mathbb{R}^d \,:\, \|w'\|_2 = 1\}} \|D\mathfrak{g}(x)[w, \mathfrak{g}(x)^{-1/2} w']\|_{\mathfrak{g}(x)^{-1}} \\
&\le 2\|w\|_{\mathfrak{g}(x)} \sup_{\{w' \in \mathbb{R}^d \,:\, \|w'\|_2 = 1\}} \|\mathfrak{g}(x)^{-1/2} w'\|_{\mathfrak{g}(x)} && \text{(Lemma 30-(a))} \\
&\le 2(1-r)^{-1}\|w\|_{\mathfrak{g}_0} \,, && \text{(Lemma 30-(c))}
\end{aligned} \tag{A.31}$$

and using again that $r \le r_\alpha^\star \le 1/11$, we have

$$\begin{aligned}
&\|\mathfrak{g}(y)^{-1/2}\{D\mathfrak{g}(y')[w] - D\mathfrak{g}(y)[w]\}\mathfrak{g}(y')^{-1/2}\|_2 \qquad (A.32)\\
&\quad = \sup_{\{w' \in \mathbb{R}^d \,:\, \|w'\|_2 = 1\}} \|D\mathfrak{g}(y')[w, \mathfrak{g}(y')^{-1/2}w'] - D\mathfrak{g}(y)[w, \mathfrak{g}(y')^{-1/2}w']\}\|_{\mathfrak{g}(y)^{-1}}\\
&\quad \le \alpha(\alpha+1)/3\|w\|_{\mathfrak{g}(y)}((1 - \|y - y'\|_{\mathfrak{g}(y)})^{-3} - 1) \qquad \text{(Lemma 32-(c))}\\
&\quad \le 5\alpha(\alpha+1)/3(1-r)^{-2}\|w\|_{\mathfrak{g}_0}\|y - y'\|_{\mathfrak{g}_0} . \qquad \text{(on the model of (A.5.1))}
\end{aligned}$$

Thus, by using (A.31) and (A.32), we have

$$\|\mathfrak{g}_0^{1/2}(D\mathrm{M}_0(y)[w] - D\mathrm{M}_0(y')[w])\mathfrak{g}_0^{1/2}\|_2 \le (12(1-r)^{-8} + 5\alpha(\alpha+1)/3(1-r)^{-6})\|y - y'\|_{\mathfrak{g}_0}\|w\|_{\mathfrak{g}_0} . \qquad (A.33)$$

Step 1.2. Secondly, we have

$$\begin{aligned}
\|\mathfrak{g}_0^{-1/2}D\mathfrak{g}_0[\mathrm{K}_0(y), \mathrm{K}_0(y)]\|_2 &\le 2\|\mathrm{K}_0(y)\|_{\mathfrak{g}_0}^2 && \text{(Lemma 30-(a))}\\
&\le 2\|\mathfrak{g}_0^{-1/2}\mathrm{M}_0(y)^{-1}(y - x^{(0)})\|_2^2 && \text{(definition of } \mathrm{K}_0)\\
&\le 2\|\mathfrak{g}_0^{-1/2}\mathrm{M}_0(y)^{-1}\mathfrak{g}_0^{-1/2}\|_2^2\|y - x^{(0)}\|_{\mathfrak{g}_0}^2\\
&\le 2r^2\|\mathfrak{g}_0^{-1/2}\mathrm{M}_0(y)^{-1}\mathfrak{g}_0^{-1/2}\|_2^2 ,
\end{aligned}$$

where we have by Lemma 30-(b)

$$\mathfrak{g}_0^{-1/2}\mathrm{M}_0(y)^{-1}\mathfrak{g}_0^{-1/2} = (\mathrm{I}_d + \mathfrak{g}_0^{1/2}\mathfrak{g}(y)^{-1}\mathfrak{g}_0^{1/2})^{-1} \preceq (1 + (1-r)^2)^{-1}\mathrm{I}_d . \qquad (A.34)$$

Hence, we obtain

$$\|\mathfrak{g}_0^{-1/2}D\mathfrak{g}_0[\mathrm{K}_0(y), \mathrm{K}_0(y)]\|_2 \le 2r^2(1 + (1-r)^2)^{-2} . \qquad (A.35)$$

Step 1.3. Thirdly, we have

$$\begin{aligned}
\|\mathfrak{g}_0^{1/2}D\mathrm{M}_0(y')[w]\mathfrak{g}_0^{1/2}\|_2 &\le (1-r)^{-2}\|\mathfrak{g}(x)^{-1/2}D\mathfrak{g}(y')[w]\mathfrak{g}(x)^{-1/2}\|_2 && (A.36)\\
&\le 2(1-r)^{-3}\|w\|_{\mathfrak{g}_0} . && \text{(using (A.31))}
\end{aligned}$$

Step 1.4. Fourthly, we have

$$\begin{aligned}
&\|\mathfrak{g}_0^{-1/2}(D\mathfrak{g}_0[\mathrm{K}_0(y), \mathrm{K}_0(y)] - D\mathfrak{g}_0[\mathrm{K}_0(y'), \mathrm{K}_0(y')])\|_2 \qquad (A.37)\\
&\quad \le 2\|D\mathfrak{g}_0[\mathrm{K}_0(y) - \mathrm{K}_0(y'), \mathrm{K}_0(y)]\|_{\mathfrak{g}_0^{-1}} + \|D\mathfrak{g}_0[\mathrm{K}_0(y) - \mathrm{K}_0(y'), \mathrm{K}_0(y) - \mathrm{K}_0(y')]\|_{\mathfrak{g}_0^{-1}}\\
&\quad \le 2\|\mathrm{K}_0(y) - \mathrm{K}_0(y')\|_{\mathfrak{g}_0}\|\mathrm{K}_0(y)\|_{\mathfrak{g}_0} + \|\mathrm{K}_0(y) - \mathrm{K}_0(y')\|_{\mathfrak{g}_0}\|\mathrm{K}_0(y) - \mathrm{K}_0(y')\|_{\mathfrak{g}_0} \qquad \text{(Lemma 30-(a))}\\
&\quad \le 4\|\mathrm{M}_0(y)^{-1}(y - x^{(0)}) - \mathrm{M}_0(y')^{-1}(y' - x^{(0)})\|_{\mathfrak{g}_0^{-1}}\|\mathrm{M}_0(y)^{-1}(y - x^{(0)})\|_{\mathfrak{g}_0^{-1}}\\
&\qquad + 2\|\mathrm{M}_0(y)^{-1}(y - x^{(0)}) - \mathrm{M}_0(y')^{-1}(y' - x^{(0)})\|_{\mathfrak{g}_0^{-1}}^2 .
\end{aligned}$$

In particular, $\mathrm{M}_0(y)^{-1}(y - x^{(0)}) - \mathrm{M}_0(y')^{-1}(y' - x^{(0)}) = \mathrm{M}_0(y)^{-1}(y - y') + (\mathrm{M}_0(y)^{-1} - \mathrm{M}_0(y')^{-1})(y' - x^{(0)})$, and thus we have

$$\begin{aligned}
&\|\mathrm{M}_0(y)^{-1}(y - x^{(0)}) - \mathrm{M}_0(y')^{-1}(y' - x^{(0)})\|_{\mathfrak{g}_0^{-1}} \qquad (A.38)\\
&\quad \le \|\mathfrak{g}_0^{-1/2}\mathrm{M}_0(y)^{-1}\mathfrak{g}_0^{-1/2}\|_2\|y - y'\|_{\mathfrak{g}_0} + r\|\mathfrak{g}_0^{-1/2}(\mathrm{M}_0(y)^{-1} - \mathrm{M}_0(y')^{-1})\mathfrak{g}_0^{-1/2}\|_2\\
&\quad \le (1 + (1-r)^2)^{-1}\|y - y'\|_{\mathfrak{g}_0} + r\|\mathfrak{g}_0^{-1/2}(\mathrm{M}_0(y')^{-1} - \mathrm{M}_0(y')^{-1})\mathfrak{g}_0^{-1/2}\|_2 . \qquad \text{(using (A.34))}
\end{aligned}$$

We now aim to find an upper bound for $\|\mathfrak{g}_0^{-1/2}(\mathrm{M}_0(y)^{-1} - \mathrm{M}_0(y')^{-1})\mathfrak{g}_0^{-1/2}\|_2$. We have

$$\begin{aligned}
&\mathfrak{g}_0^{-1/2}(\mathrm{M}_0(y)^{-1} - \mathrm{M}_0(y')^{-1})\mathfrak{g}_0^{-1/2}\\
&\quad = (\mathrm{I}_d + \mathfrak{g}_0^{1/2}\mathfrak{g}(y)^{-1}\mathfrak{g}_0^{1/2})^{-1} - (\mathrm{I}_d + \mathfrak{g}_0^{1/2}\mathfrak{g}(y')^{-1}\mathfrak{g}_0^{1/2})^{-1}\\
&\quad = (\mathrm{I}_d + \mathfrak{g}_0^{1/2}\mathfrak{g}(y')^{-1}\mathfrak{g}_0^{1/2} + \mathfrak{g}_0^{1/2}\{\mathfrak{g}(y)^{-1} - \mathfrak{g}(y')^{-1}\}\mathfrak{g}_0^{1/2})^{-1} - (\mathrm{I}_d + \mathfrak{g}_0^{1/2}\mathfrak{g}(y')^{-1}\mathfrak{g}_0^{1/2})^{-1}\\
&\quad = \mathrm{B}(y')^{-1/2}[(\mathrm{I}_d + \mathrm{B}(y')^{-1/2}\mathfrak{g}_0^{1/2}\{\mathfrak{g}(y)^{-1} - \mathfrak{g}(y')^{-1}\}\mathfrak{g}_0^{1/2}\mathrm{B}(y')^{-1/2})^{-1} - \mathrm{I}_d]\mathrm{B}(y')^{-1/2} ,
\end{aligned}$$

where $\mathrm{B}(y') = \mathrm{I}_d + \mathfrak{g}_0^{1/2}\mathfrak{g}(y')^{-1}\mathfrak{g}_0^{1/2}$. In particular, we have

$$(1+(1-r)^{-2})^{-1/2}\mathrm{I}_d \preceq \mathrm{B}(y')^{-1/2} \preceq (1+(1-r)^2)^{-1/2}\mathrm{I}_d\,, \tag{A.39}$$

by Lemma 30-(b). Note that (A.5.1) still holds, where x, x' and $\bar{r}$ are respectively replaced by y, y' and r. Thus, on the model on (A.17), we have

$$\{(1-\|y'-y\|_{\mathfrak{g}(y)})^2-1\}(1-r)^2\mathrm{I}_d \preceq \mathfrak{g}_0^{1/2}\{\mathfrak{g}(y)^{-1}-\mathfrak{g}(y')^{-1}\}\mathfrak{g}_0^{1/2} \preceq \{(1-\|y'-y\|_{\mathfrak{g}(y)})^{-2}-1\}(1-r)^{-2}\mathrm{I}_d\,,$$

and then

$$\begin{aligned}(1+\{(1-\|y'-y\|_{\mathfrak{g}(y)})^2-1\}(1-r)^2(1+(1-r)^{-2})^{-1})\mathrm{I}_d \\ \preceq \mathrm{I}_d + \mathrm{B}(y')^{-1/2}\mathfrak{g}_0^{1/2}\{\mathfrak{g}(y)^{-1}-\mathfrak{g}(y')^{-1}\}\mathfrak{g}_0^{1/2}\mathrm{B}(y')^{-1/2} \\ \preceq (1+\{(1-\|y'-y\|_{\mathfrak{g}(y)})^{-2}-1\}(1-r)^{-2}(1+(1-r)^2)^{-1})\mathrm{I}_d\,.\end{aligned}$$

Therefore, we have $\|\mathfrak{g}_0^{-1/2}(\mathrm{M}_0(y')^{-1}-\mathrm{M}_0(y')^{-1})\mathfrak{g}_0^{-1/2}\|_2 \le (1+(1-r)^2)^{-1}\max(|f_1(r)|, f_2(r))$ where

$$\begin{aligned}f_1(r) &= (1+\{(1-\|y'-y\|_{\mathfrak{g}(y)})^{-2}-1\}(1-r)^{-2}(1+(1-r)^2)^{-1})^{-1}-1,\\ f_2(r) &= (1+\{(1-\|y'-y\|_{\mathfrak{g}(y)})^{2}-1\}(1-r)^{2}(1+(1-r)^{-2})^{-1})^{-1}-1\,.\end{aligned}$$

We now aim to control the upper bound $\max(|f_1(r)|, f_2(r))$.

(a) We first bound $|f_1(r)|$. Since $\|y'-y\|_{\mathfrak{g}(y)} \ge 0$, it is clear that $f_1(r) \le 0$. Using Inequality (a) with $u = \|y'-y\|_{\mathfrak{g}(y)}$, where $u \le 1/5$ by (A.5.1), we obtain

$$\begin{aligned}|f_1(r)| &= -f_1(r)\\ &\le 1-(1+3(1-r)^{-2}(1+(1-r)^2)^{-1}\|y'-y\|_{\mathfrak{g}(y)})^{-1}\\ &\le 3(1-r)^{-2}(1+(1-r)^2)^{-1}\|y'-y\|_{\mathfrak{g}(y)} && \text{(Inequality (d))}\\ &\le 3(1-r)^{-3}(1+(1-r)^2)^{-1}\|y'-y\|_{\mathfrak{g}_0}\,. && \text{(Lemma 30-(c))}\end{aligned}$$

(b) We now bound $f_2(r)$. We have

$$\begin{aligned}f_2(r) &\le (1-2(1-r)^2(1+(1-r)^{-2})^{-1}\|y'-y\|_{\mathfrak{g}(y)})^{-1}-1\\ &\le 4(1-r)^2(1+(1-r)^{-2})^{-1}\|y'-y\|_{\mathfrak{g}(y)}\\ &\le 4(1-r)(1+(1-r)^{-2})^{-1}\|y'-y\|_{\mathfrak{g}_0}\,, && \text{(Lemma 30-(c))}\end{aligned}$$

where we used (i) Inequality (d) in the first line with $u = \|y'-y\|_{\mathfrak{g}(y)}$ and (ii) Inequality (b) in the second line with $u = 2(1-r)^2(1+(1-r)^{-2})^{-1}\|y'-y\|_{\mathfrak{g}(y)} \le 4r(1-r)(1+(1-r)^{-2})^{-1} \le 1/2$ with $r \le 1/11$.

We recall that $r \in (0,1)$, and thus, we have respectively $(1-r)^{-3} \ge (1-r)$ and $(1+(1-r)^2)^{-1} \ge (1+(1-r)^{-2})^{-1}$. Therefore, $\max(|f_1(r)|, f_2(r)) \le (4/3)|f_1(r)|$ and

$$\begin{aligned}\|\mathfrak{g}_0^{-1/2}(\mathrm{M}_0(y')^{-1}-\mathrm{M}_0(y')^{-1})\mathfrak{g}_0^{-1/2}\|_2 &\le 4(1-r)^{-3}(1+(1-r)^2)^{-2}\|y'-y\|_{\mathfrak{g}_0}\,,\\ \|\mathrm{K}_0(y)-\mathrm{K}_0(y')\|_{\mathfrak{g}_0} &\le (1+(1-r)^2)^{-1}\{1+4r(1-r)^{-3}(1+(1-r)^2)^{-1}\}\|y'-y\|_{\mathfrak{g}_0}\,.\end{aligned} \tag{A.40}$$

By combining (A.37), (A.38) and (A.40), we finally have

$$\begin{aligned}\|\mathfrak{g}_0^{-1/2}(D\mathfrak{g}_0[\mathrm{K}_0(y),\mathrm{K}_0(y)]-D\mathfrak{g}_0[\mathrm{K}_0(y'),\mathrm{K}_0(y')])\|_2 \\ \le 4r(1+(1-r)^2)^{-2}\{1+4r(1-r)^{-3}(1+(1-r)^2)^{-1}\}\|y-y'\|_{\mathfrak{g}_0} \\ \quad + 4r(1+(1-r)^2)^{-2}\{1+4r(1-r)^{-3}(1+(1-r)^2)^{-1}\}^2\|y-y'\|_{\mathfrak{g}_0} \\ \le 4r(1+(1-r)^2)^{-2}\{2+4r(1-r)^{-3}(1+(1-r)^2)^{-1}\}^2\|y-y'\|_{\mathfrak{g}_0}\,.\end{aligned}$$

Conclusion of Step 1. By combining the results of Steps 1.1 to 1.4, it comes that

$$\begin{aligned}\|D\mathrm{M}_0(y)[w]D\mathfrak{g}_0[\mathrm{K}_0(y),\mathrm{K}_0(y)] - D\mathrm{M}_0(y')[w]D\mathfrak{g}_0[\mathrm{K}_0(y'),\mathrm{K}_0(y')]\|_{\mathfrak{g}_0} \\ = \|\mathfrak{g}_0^{1/2}(a_1(y)-a_1(y'))\|_2 \\ \le \{2r^2(1+(1-r)^2)^{-2}(12(1-r)^{-8}+5\alpha(\alpha+1)/3(1-r)^{-6}) \\ \quad + 8r(1-r)^{-3}(1+(1-r)^2)^{-2}\{2+4r(1-r)^{-3}(1+(1-r)^2)^{-1}\}^2\}\|y-y'\|_{\mathfrak{g}_0}\|w\|_{\mathfrak{g}_0}\,.\end{aligned}$$

Step 2. Let us now bound the second term in (A.29). For any $x \in \mathsf{B}_0$, we denote $\mathrm{M}_0(y)D\mathfrak{g}_0[D\mathrm{K}_0(y)[w], \mathrm{K}_0(y)]$ by $a_2(x)$. We have

$$\begin{aligned}\mathfrak{g}_0^{1/2}(a_2(y) - a_2(y')) = \{\mathfrak{g}_0^{1/2}(\mathrm{M}_0(y) - \mathrm{M}_0(y'))\mathfrak{g}_0^{1/2}\}\{\mathfrak{g}_0^{-1/2}D\mathfrak{g}_0[D\mathrm{K}_0(y)[w], \mathrm{K}_0(y)]\} \\ + \{\mathfrak{g}_0^{1/2}\mathrm{M}_0(y')\mathfrak{g}_0^{1/2}\}\{\mathfrak{g}_0^{-1/2}(D\mathfrak{g}_0[D\mathrm{K}_0(y)[w], \mathrm{K}_0(y)] - D\mathfrak{g}_0[D\mathrm{K}_0(y')[w], \mathrm{K}_0(y')])\} .\end{aligned} \tag{A.41}$$

We now aim to bound each of the four terms that appear in (A.41).

Step 2.1. First, we have

$$\begin{aligned}\|\mathfrak{g}_0^{1/2}(\mathrm{M}_0(y) - \mathrm{M}_0(y'))\mathfrak{g}_0^{1/2}\|_2 &= \|\mathfrak{g}_0^{1/2}(\mathfrak{g}(y)^{-1} - \mathfrak{g}(y')^{-1})\mathfrak{g}_0^{1/2}\|_2 \\ &\le 3(1-r)^{-3}\|y - y'\|_{\mathfrak{g}_0} . \qquad \text{(on the model of (A.17))}\end{aligned}$$

Step 2.2. Secondly, we have

$$\begin{aligned}\|\mathfrak{g}_0^{-1/2}D\mathfrak{g}_0[D\mathrm{K}_0(y)[w], \mathrm{K}_0(y)]\|_2 &\le 2\|D\mathrm{K}_0(y)[w]\|_{\mathfrak{g}_0}\|\mathrm{M}_0(y)^{-1}(y - x^{(0)})\|_{\mathfrak{g}_0^{-1}} \qquad \text{(Lemma 30-(a))} \\ &\le 2r(1 + (1-r)^2)^{-1}\|D\mathrm{K}_0(y)[w]\|_{\mathfrak{g}_0} , \qquad \text{(see Step 1.2.)}\end{aligned}$$

where

$$\begin{aligned}\|D\mathrm{K}_0(y)[w]\|_{\mathfrak{g}_0} \le r\|\mathfrak{g}_0^{-1/2}(\mathfrak{g}(y)\mathfrak{g}_0^{-1} + \mathrm{I}_d)^{-1}\mathfrak{g}_0^{1/2}\|_2^2\|\mathfrak{g}_0^{-1/2}\mathfrak{g}(y)^{1/2}\|_2^2\|\mathfrak{g}(y)^{-1/2}D\mathfrak{g}(y)[w]\mathfrak{g}(y)^{-1/2}\|_2 \\ + \quad \|\mathrm{M}_0(y)^{-1}w\|_{\mathfrak{g}_0^{-1}} .\end{aligned}$$

Using (A.31) and (A.34), it comes that

$$\|D\mathrm{K}_0(y)[w]\|_{\mathfrak{g}_0} \le \{2r(1-r)^{-3}(1 + (1-r)^2)^{-2} + (1 + (1-r)^2)^{-1}\}\|w\|_{\mathfrak{g}_0} .$$

Then, we have

$$\|\mathfrak{g}_0^{-1/2}D\mathfrak{g}_0[D\mathrm{K}_0(y)[w], \mathrm{K}_0(y)]\|_2 \le 2r(1 + (1-r)^2)^{-2}\{1 + 2r(1-r)^{-3}(1 + (1-r)^2)^{-1}\}\|w\|_{\mathfrak{g}_0} .$$

Step 2.3. Thirdly, we have $\|\mathfrak{g}_0^{1/2}\mathrm{M}_0(y')\mathfrak{g}_0^{1/2}\|_2 = \|\mathfrak{g}_0^{1/2}\mathfrak{g}(y')^{-1}\mathfrak{g}_0^{1/2} + \mathrm{I}_d\|_2 \le 1 + (1-r)^{-2}$.

Step 2.4. Fourthly, using Lemma 30-(a), we have

$$\begin{aligned}&\|\mathfrak{g}_0^{-1/2}(D\mathfrak{g}_0[D\mathrm{K}_0(y)[w], \mathrm{K}_0(y)] - D\mathfrak{g}_0[D\mathrm{K}_0(y')[w], \mathrm{K}_0(y')])\|_2 \\ &\quad\le \|D\mathfrak{g}_0[D\mathrm{K}_0(y)[w] - D\mathrm{K}_0(y')[w], \mathrm{K}_0(y)]\|_{\mathfrak{g}_0^{-1}} + \|D\mathfrak{g}_0[D\mathrm{K}_0(y)[w], \mathrm{K}_0(y) - \mathrm{K}_0(y')]\|_{\mathfrak{g}_0^{-1}} \\ &\qquad + \|D\mathfrak{g}_0[D\mathrm{K}_0(y)[w] - D\mathrm{K}_0(y')[w], \mathrm{K}_0(y) - \mathrm{K}_0(y')]\|_{\mathfrak{g}_0^{-1}} \\ &\quad\le 2\|D\mathrm{K}_0(y)[w] - D\mathrm{K}_0(y')[w]\|_{\mathfrak{g}_0}\|\mathrm{K}_0(y)]\|_{\mathfrak{g}_0} + 2\|D\mathrm{K}_0(y)[w]\|_{\mathfrak{g}_0}\|\mathrm{K}_0(y) - \mathrm{K}_0(y')\|_{\mathfrak{g}_0} \\ &\qquad + 2\|D\mathrm{K}_0(y)[w] - D\mathrm{K}_0(y')[w]\|_{\mathfrak{g}_0}\|\mathrm{K}_0(y) - \mathrm{K}_0(y')\|_{\mathfrak{g}_0} \\ &\quad\le 2\{\|\mathrm{K}_0(y)]\|_{\mathfrak{g}_0} + \|\mathrm{K}_0(y) - \mathrm{K}_0(y')\|_{\mathfrak{g}_0}\}\|D\mathrm{K}_0(y)[w] - D\mathrm{K}_0(y')[w]\|_{\mathfrak{g}_0} \\ &\qquad + 2\|D\mathrm{K}_0(y)[w]\|_{\mathfrak{g}_0}\|\mathrm{K}_0(y) - \mathrm{K}_0(y')\|_{\mathfrak{g}_0} .\end{aligned}$$

We recall that the following inequalities hold

$$\begin{aligned}\|\mathrm{K}_0(y)]\|_{\mathfrak{g}_0} &\le r(1 + (1-r)^2)^{-1} , \\ \|D\mathrm{K}_0(y)[w]\|_{\mathfrak{g}_0} &\le (1 + (1-r)^2)^{-1}\{1 + 2r(1-r)^{-3}(1 + (1-r)^2)^{-1}\}\|w\|_{\mathfrak{g}_0} , \\ \|\mathrm{K}_0(y) - \mathrm{K}_0(y')\|_{\mathfrak{g}_0} &\le (1 + (1-r)^2)^{-1}\{1 + 4r(1-r)^{-3}(1 + (1-r)^2)^{-1}\}\|y' - y\|_{\mathfrak{g}_0} .\end{aligned}$$

We are now going to bound $\|D\mathrm{K}_0(y)[w] - D\mathrm{K}_0(y')[w]\|_{\mathfrak{g}_0}$. We have

$$\|D\mathrm{K}_0(y)[w] - D\mathrm{K}_0(y')[w]\|_{\mathfrak{g}_0} \le \|(\mathrm{M}_0(y)^{-1} - \mathrm{M}_0(y')^{-1})w\|_{\mathfrak{g}_0^{-1}} + A_1 + A_2 + A_3 + A_4 ,$$

where

$$
\begin{aligned}
A_1 &= \|\{(\mathrm{I}_d + \mathfrak{g}_0^{-1/2}\mathfrak{g}(y)\mathfrak{g}_0^{-1/2})^{-1} - (\mathrm{I}_d + \mathfrak{g}_0^{-1/2}\mathfrak{g}(y')\mathfrak{g}_0^{-1/2})^{-1}\} \\
&\qquad \times \mathfrak{g}_0^{-1/2} D\mathfrak{g}(y)[w]\mathfrak{g}_0^{-1/2}(\mathrm{I}_d + \mathfrak{g}_0^{-1/2}\mathfrak{g}(y)\mathfrak{g}_0^{-1/2})^{-1}\mathfrak{g}_0^{1/2}(y - x^{(0)})\|_2 \,, \\
A_2 &= \|(\mathrm{I}_d + \mathfrak{g}_0^{-1/2}\mathfrak{g}(y')\mathfrak{g}_0^{-1/2})^{-1}\mathfrak{g}_0^{-1/2}\{D\mathfrak{g}(y)[w] - D\mathfrak{g}(y')[w]\}\mathfrak{g}_0^{-1/2}(\mathrm{I}_d + \mathfrak{g}_0^{-1/2}\mathfrak{g}(y)\mathfrak{g}_0^{-1/2})^{-1}\mathfrak{g}_0^{1/2}(y - x^{(0)})\|_2 \,, \\
A_3 &= \|(\mathrm{I}_d + \mathfrak{g}_0^{-1/2}\mathfrak{g}(y')\mathfrak{g}_0^{-1/2})^{-1}\mathfrak{g}_0^{-1/2} D\mathfrak{g}(y)[w]\mathfrak{g}_0^{-1/2} \\
&\qquad \times \{(\mathrm{I}_d + \mathfrak{g}_0^{-1/2}\mathfrak{g}(y)\mathfrak{g}_0^{-1/2})^{-1} - (\mathrm{I}_d + \mathfrak{g}_0^{-1/2}\mathfrak{g}(y')\mathfrak{g}_0^{-1/2})^{-1}\}\mathfrak{g}_0^{1/2}(y - x^{(0)})\|_2 \,, \\
A_3 &= \|(\mathrm{I}_d + \mathfrak{g}_0^{-1/2}\mathfrak{g}(y')\mathfrak{g}_0^{-1/2})^{-1}\mathfrak{g}_0^{-1/2} D\mathfrak{g}(y)[w]\mathfrak{g}_0^{-1/2}(\mathrm{I}_d + \mathfrak{g}_0^{-1/2}\mathfrak{g}(y')\mathfrak{g}_0^{-1/2})^{-1}\mathfrak{g}_0^{1/2}(y - y')\|_2 \,.
\end{aligned}
$$

In particular, we have

$$
\begin{aligned}
\|(\mathrm{M}_0(y')^{-1} - \mathrm{M}_0(y')^{-1})w\|_{\mathfrak{g}_0^{-1}} &\le 4(1-r)^{-3}(1 + (1-r)^2)^{-2}\|y' - y\|_{\mathfrak{g}_0}\|w\|_{\mathfrak{g}_0} \,, && \text{((A.40))} \\
\max(A_1, A_3) &\le 8r(1-r)^{-6}(1 + (1-r)^2)^{-3}\|y - y'\|_{\mathfrak{g}_0}\|w\|_{\mathfrak{g}_0} \,, && \text{((A.31), (A.34), (A.40))} \\
A_2 &\le 5\alpha(\alpha+1)/3r(1-r)^{-4}(1 + (1-r)^2)^{-2}\|y - y'\|_{\mathfrak{g}_0}\|w\|_{\mathfrak{g}_0} \,, && \text{((A.32), (A.34))} \\
A_4 &\le 2(1-r)^{-3}(1 + (1-r)^2)^{-2}\|y - y'\|_{\mathfrak{g}_0}\|w\|_{\mathfrak{g}_0} \,. && \text{((A.31) and (A.34))}
\end{aligned}
$$

Therefore, we have

$$
\begin{aligned}
&\|\mathfrak{g}_0^{-1/2}(D\mathfrak{g}_0[D\mathrm{K}_0(y)[w], \mathrm{K}_0(y)] - D\mathfrak{g}_0[D\mathrm{K}_0(y')[w], \mathrm{K}_0(y')])\|_2 \\
&\quad \le 2r(1 + (1-r)^2)^{-1}\{3 + 4r(1-r)^{-3}(1 + (1-r)^2)^{-1})\} \\
&\qquad \times \{4(1-r)^{-3}(1 + (1-r)^2)^{-2} + 16r(1-r)^{-6}(1 + (1-r)^2)^{-3} \\
&\qquad\quad + 5\alpha(\alpha+1)/3r(1-r)^{-4}(1 + (1-r)^2)^{-2} + 2(1-r)^{-3}(1 + (1-r)^2)^{-2}\}\|y - y'\|_{\mathfrak{g}_0}\|w\|_{\mathfrak{g}_0} \\
&\qquad + 2\{1 + 2r(1-r)^{-3}(1 + (1-r)^2)^{-1}\} \\
&\qquad\quad \times (1 + (1-r)^2)^{-2}(1 + 4r(1-r)^{-3}(1 + (1-r)^2)^{-1})\|y - y'\|_{\mathfrak{g}_0}\|w\|_{\mathfrak{g}_0} \,.
\end{aligned}
$$

Conclusion of Step 2. By combining the results of Steps 2.1 to 2.4, it comes that

$$
\begin{aligned}
&\|\mathrm{M}_0(y)D\mathfrak{g}_0[D\mathrm{K}_0(y)[w], \mathrm{K}_0(y)] - \mathrm{M}_0(y')D\mathfrak{g}_0[D\mathrm{K}_0(y')[w], \mathrm{K}_0(y')]\|_{\mathfrak{g}_0} \\
&\quad = \|\mathfrak{g}_0^{1/2}(a_2(y) - a_2(y'))\|_2 \\
&\quad \le \{6r(1-r)^{-3}(1 + (1-r)^2)^{-2}[1 + 2r(1-r)^{-3}(1 + (1-r)^2)^{-1}] \\
&\qquad + 2r[1 + (1-r)^{-2}](1 + (1-r)^2)^{-1}\{3 + 4r(1-r)^{-3}(1 + (1-r)^2)^{-1})\} \\
&\qquad\quad \times \{4(1-r)^{-3}(1 + (1-r)^2)^{-2} + 16r(1-r)^{-6}(1 + (1-r)^2)^{-3} \\
&\qquad\qquad + 5\alpha(\alpha+1)/3r(1-r)^{-4}(1 + (1-r)^2)^{-2} + 2(1-r)^{-3}(1 + (1-r)^2)^{-2}\} \\
&\qquad + 2[1 + (1-r)^{-2}]\{1 + 2r(1-r)^{-3}(1 + (1-r)^2)^{-1}\} \\
&\qquad\quad \times (1 + (1-r)^2)^{-2}(1 + 4r(1-r)^{-3}(1 + (1-r)^2)^{-1})\}\|y - y'\|_{\mathfrak{g}_0}\|w\|_{\mathfrak{g}_0} \,.
\end{aligned}
$$

Conclusion. Finally, we have $\|D\phi_0(y)[w] - D\phi_0(y')[w]\|_{\mathfrak{g}_0} \le c(r)\|y - y'\|_{\mathfrak{g}_0}\|w\|_{\mathfrak{g}_0}$, where

$$
\begin{aligned}
c(r) &\le 24r^2(1-r)^{-12} + 10\alpha(\alpha+1)/3r^2(1-r)^{-10} \\
&\quad + 32r(1-r)^{-7} + 128r^2(1-r)^{-12} + 128r^3(1-r)^{-17} \\
&\quad + 12r(1-r)^{-7} + 24r^2(1-r)^{-12} \\
&\quad + \{48r(1-r)^{-9} + 192r^2(1-r)^{-14} + 20\alpha(\alpha+1)r^2(1-r)^{-10} + 24r(1-r)^{-9}\}\{1 + (1-r)^{-2}\} \\
&\quad + \{64r^2(1-r)^{-14} + 256r^3(1-r)^{-19} + 80\alpha(\alpha+1)/3r^3(1-r)^{-15} + 32r^2(1-r)^{-14}\}\{1 + (1-r)^{-2}\} \\
&\quad + \{4(1-r)^{-4} + 24r(1-r)^{-9} + 32r^2(1-r)^{-14}\}\{1 + (1-r)^{-2}\} \,,
\end{aligned}
$$

by combining the results of Step 1 (two first lines) and Step 2 (following lines). Hence, using that $r \le 1/11$, we have

$$
\begin{aligned}
c(r) &\le \max(24 \times 256, 5 \times 80\alpha(\alpha+1)/3)(1-r)^{-21} \\
&\le \max(6144, 400\alpha(\alpha+1)/3)15/2 \\
&\le \max(46080, 1000\alpha(\alpha+1)) \\
&\le 1/r_\alpha^\star \,, && \text{(see (A.28))}
\end{aligned}
$$

i.e., for any $(y, y') \in \mathsf{B}_0 \times \mathsf{B}_0$ and $w \in \mathbb{R}^d$, we have

$$\|D\phi_0(y)[w] - D\phi_0(y)[w]\|_{\mathfrak{g}_0} \leq (r_\alpha^\star)^{-1}\|y - y'\|_{\mathfrak{g}_0}\|w\|_{\mathfrak{g}_0} ,$$

and thus

$$\|D\phi_0(y) - D\phi_0(y)\|_{\mathfrak{g}_0} \leq (r_\alpha^\star)^{-1}\|y - y'\|_{\mathfrak{g}_0} .$$

This last inequality finally proves that $D\phi_0$ is $(r_\alpha^\star)^{-1}$-Lipschitz-continuous on B_0 with respect to $\|\cdot\|_{\mathfrak{g}_0}$.

Inequality on $D\tilde{\phi}_0$. Elaborating on the smoothness of $D\phi_0$, we prove here that $\|D\tilde{\phi}_0(y)\|_{\mathfrak{g}_0} > 1/2$ for any $y \in \mathsf{B}_0$. Let $y \in \mathsf{B}_0$, we have

$$\begin{aligned} \|\mathrm{I}_d\|_{\mathfrak{g}_0} &= \|D\tilde{\phi}_0(y) + 1/2 D\phi_0(y)\|_{\mathfrak{g}_0} , \\ 1 &\leq \|D\tilde{\phi}_0(y)\|_{\mathfrak{g}_0} + 1/2\|D\phi_0(y)\|_{\mathfrak{g}_0} , \\ 1 - 1/2\|D\phi_0(y)\|_{\mathfrak{g}_0} &\leq \|D\tilde{\phi}_0(y)\|_{\mathfrak{g}_0} . \end{aligned} \tag{A.42}$$

We recall that $D\phi_0(x^{(0)}) = 0$ and that $D\phi_0$ is $(r_\alpha^\star)^{-1}$-Lipschitz-continuous on B_0. Thus, we obtain

$$\|D\phi_0(y)\|_{\mathfrak{g}_0} = \|D\phi_0(y) - D\phi_0(x^{(0)})\|_{\mathfrak{g}_0} \leq (r_\alpha^\star)^{-1}\|y - x^{(0)}\|_{\mathfrak{g}_0} < (r_\alpha^\star)^{-1} r \leq 1 ,$$

and then $\|D\tilde{\phi}_0(y)\|_{\mathfrak{g}_0} > 1/2$, which proves the result.

Smoothness of ϕ_0 and $\tilde{\phi}_0$. We prove here that ϕ_0 (and thus $\tilde{\phi}_0$) is Lipschitz-continuous on B_0 with respect to $\|\cdot\|_{\mathfrak{g}_0}$. Let $(y, y') \in \mathsf{B}_0 \times \mathsf{B}_0$. We have

$$\begin{aligned} \|\phi_0(y) - \phi_0(y')\|_{\mathfrak{g}_0} \leq{}& \|\{\mathrm{M}_0(y) - \mathrm{M}_0(y')\} D\mathfrak{g}_0[\mathrm{K}_0(y), \mathrm{K}_0(y)]\|_{\mathfrak{g}_0} \\ &+ \|\mathrm{M}_0(y')\{D\mathfrak{g}_0[\mathrm{K}_0(y), \mathrm{K}_0(y)] - D\mathfrak{g}_0[\mathrm{K}_0(y'), \mathrm{K}_0(y')]\}\|_{\mathfrak{g}_0} . \end{aligned} \tag{A.43}$$

To prove the Lipschitz-continuity of ϕ_0, we are going to proceed in two steps, each of them consisting of bounding from above a term of (A.43).

Step 1. Let us first bound the first term in (A.43). Since $r \leq 1/11$, we have

$$\begin{aligned} &\|\{\mathrm{M}_0(y) - \mathrm{M}_0(y')\} D\mathfrak{g}_0[\mathrm{K}_0(y), \mathrm{K}_0(y)]\|_{\mathfrak{g}_0} \\ &\quad \leq \|\mathfrak{g}_0^{1/2}\{\mathrm{M}_0(y) - \mathrm{M}_0(y')\}\mathfrak{g}_0^{1/2}\|_2 \|D\mathfrak{g}_0[\mathrm{K}_0(y), \mathrm{K}_0(y)]\|_{\mathfrak{g}_0^{-1}} \\ &\quad \leq 3(1-r)^{-3}\|y - y'\|_{\mathfrak{g}_0} \times 2\|\mathrm{K}_0(y)\|_{\mathfrak{g}_0}^2 && \text{(see (A.17) and Lemma 30-(a))} \\ &\quad \leq 6r^2(1-r)^{-3}(1 + (1-r)^2)^{-2}\|y - y'\|_{\mathfrak{g}_0} . && \text{(see (A.35))} \end{aligned}$$

Step 2. Let us now bound the second term in (A.43). We have

$$\begin{aligned} &\|\mathrm{M}_0(y')\{D\mathfrak{g}_0[\mathrm{K}_0(y), \mathrm{K}_0(y)] - D\mathfrak{g}_0[\mathrm{K}_0(y'), \mathrm{K}_0(y')]\}\|_{\mathfrak{g}_0} \\ &\quad \leq \|\mathfrak{g}_0^{1/2}\mathrm{M}_0(y')\mathfrak{g}_0^{1/2}\|_2 \|D\mathfrak{g}_0[\mathrm{K}_0(y), \mathrm{K}_0(y)] - D\mathfrak{g}_0[\mathrm{K}_0(y'), \mathrm{K}_0(y')]\|_{\mathfrak{g}_0^{-1}} \\ &\quad \leq \{1 + (1-r)^{-2}\}\{2\|D\mathfrak{g}_0[\mathrm{K}_0(y) - \mathrm{K}_0(y'), \mathrm{K}_0(y)]\|_{\mathfrak{g}_0^{-1}} + \|D\mathfrak{g}_0[\mathrm{K}_0(y) - \mathrm{K}_0(y'), \mathrm{K}_0(y) - \mathrm{K}_0(y')]\|_{\mathfrak{g}_0^{-1}}\} \\ &\quad \leq \{1 + (1-r)^{-2}\}\{4\|\mathrm{K}_0(y) - \mathrm{K}_0(y')\|_{\mathfrak{g}_0}\|\mathrm{K}_0(y)\|_{\mathfrak{g}_0} + 2\|\mathrm{K}_0(y) - \mathrm{K}_0(y')\|_{\mathfrak{g}_0}^2\} \\ &\quad \leq 1\{1 + (1-r)^{-2}\}\|\mathrm{K}_0(y) - \mathrm{K}_0(y')\|_{\mathfrak{g}_0}\{2\|\mathrm{K}_0(y)\|_{\mathfrak{g}_0} + \|\mathrm{K}_0(y) - \mathrm{K}_0(y')\|_{\mathfrak{g}_0}\} , \end{aligned}$$

where we recall that the following inequalities hold

$$\begin{aligned} \|\mathrm{K}_0(y)]\|_{\mathfrak{g}_0} &\leq r(1 + (1-r)^2)^{-1} , \\ \|\mathrm{K}_0(y) - \mathrm{K}_0(y')\|_{\mathfrak{g}_0} &\leq (1 + (1-r)^2)^{-1}\{1 + 4r(1-r)^{-3}(1 + (1-r)^2)^{-1}\}\|y' - y\|_{\mathfrak{g}_0} . \end{aligned}$$

Thus, we have

$$\begin{aligned} &\|\mathrm{M}_0(y')\{D\mathfrak{g}_0[\mathrm{K}_0(y), \mathrm{K}_0(y)] - D\mathfrak{g}_0[\mathrm{K}_0(y'), \mathrm{K}_0(y')]\}\|_{\mathfrak{g}_0} \\ &\quad \leq 4r\{1 + (1-r)^2\}^{-2}\{1 + 4r(1-r)^{-3}(1 + (1-r)^2)^{-1}\}^2\|y - y'\|_{\mathfrak{g}_0} . \end{aligned}$$

Conclusion. Since $r \le 1/11$, we finally have

$$\begin{aligned}\|\phi_0(y)-\phi_0(y')\|_{\mathfrak{g}_0} &\le \{6r^2(1-r)^{-3}(1+(1-r)^2)^{-2}\\ &\quad 4r\{1+(1-r)^2\}^{-2}\{1+4r(1-r)^{-3}(1+(1-r)^2)^{-1}\}^2\}\|y-y'\|_{\mathfrak{g}_0}\\ &\le \{6r^2(1-r)^{-7}+4r(1-r)^{-4}+32r^2(1-r)^{-9}+64r^3(1-r)^{-14}\}\|y-y'\|_{\mathfrak{g}_0}\\ &\le 4\times 64r(1-r)^{-14}\|y-y'\|_{\mathfrak{g}_0}\\ &\le 4864/5r\|y-y'\|_{\mathfrak{g}_0}\,.\end{aligned}$$

Therefore, ϕ_0 and $\tilde{\phi}_0$ are respectively $(4864/5r)$ and $(1+2432/5r)$-Lipschitz-continuous on B_0 with respect to $\|\cdot\|_{\mathfrak{g}_0}$.

We are now ready to prove the second result of Lemma 43.

Invertibility of $\mathrm{Jac}[\mathrm{G}_{h,x^{(0)}}]$**.** Let $x \in \mathsf{B}_0$ and $w \in \mathbb{R}^d$. We have $(h/2)\mathrm{G}_{h,x^{(0)}} = \mathrm{M}_0(y)^{-1}\tilde{\phi}_0(y)$ and thus,

$$\begin{aligned}(h/2)D\mathrm{G}_{h,x^{(0)}}(x)[w] &= \mathrm{M}_0(x)^{-1}D\tilde{\phi}_0(x)[w] + D\mathrm{M}_0(x)^{-1}[w]\tilde{\phi}_0(x)\,,\\ D\tilde{\phi}_0(x)[w] &= (h/2)\mathrm{M}_0(x)D\mathrm{G}_{h,x^{(0)}}(x)[w] - \mathrm{M}_0(x)D\mathrm{M}_0(x)^{-1}[w]\tilde{\phi}_0(x)\,.\end{aligned}$$

Then, we have

$$\begin{aligned}\|D\tilde{\phi}_0(x)[w]\|_{\mathfrak{g}_0} &\le (h/2)\|\mathfrak{g}_0^{1/2}\mathrm{M}_0(x)\mathfrak{g}_0^{1/2}\|_2\|D\mathrm{G}_{h,x^{(0)}}(x)[w]\|_{\mathfrak{g}_0^{-1}}\\ &\quad + \|\mathfrak{g}_0^{1/2}\mathrm{M}_0(x)\mathfrak{g}_0^{1/2}\|_2\|\mathfrak{g}_0^{-1/2}D\mathrm{M}_0(x)^{-1}[w]\mathfrak{g}_0^{-1/2}\|_2\|\tilde{\phi}_0(x)\|_{\mathfrak{g}_0}\\ &\le \{1+(1-r)^{-2}\}\{(h/2)\|D\mathrm{G}_{h,x^{(0)}}(x)[w]\|_{\mathfrak{g}_0^{-1}} + \|\mathfrak{g}_0^{-1/2}D\mathrm{M}_0(x)^{-1}[w]\mathfrak{g}_0^{-1/2}\|_2\|\tilde{\phi}_0(x)\|_{\mathfrak{g}_0}\}\,,\end{aligned} \tag{A.44}$$

where $D\mathrm{M}_0(x)^{-1}[w] = -\mathrm{M}_0(x)^{-1}D\mathrm{M}_0(x)[w]\mathrm{M}_0(x)^{-1}$ and

$$\begin{aligned}\|\mathfrak{g}_0^{-1/2}D\mathrm{M}_0(x)^{-1}[w]\mathfrak{g}_0^{-1/2}\|_2 &\le \|\mathfrak{g}_0^{-1/2}\mathrm{M}_0(x)^{-1}\mathfrak{g}_0^{-1/2}\|_2^2\|\mathfrak{g}_0^{-1/2}D\mathrm{M}_0(x)[w]\mathfrak{g}_0^{-1/2}\|_2\\ &\le 2(1-r)^{-3}(1+(1-r)^2)^{-1}\|w\|_{\mathfrak{g}_0} \qquad \text{(see (A.36))}\\ &\le 2(1-r)^{-7}\|w\|_{\mathfrak{g}_0}\,.\end{aligned} \tag{A.45}$$

Moreover, using the Lipschitz-continuity of $\tilde{\phi}_0$ on B_0 and $\tilde{\phi}_0(x^{(0)}) = 0$, we have

$$\|\tilde{\phi}_0(x)\|_{\mathfrak{g}_0} \le \|\tilde{\phi}_0(x)-\tilde{\phi}_0(x^{(0)})\|_{\mathfrak{g}_0} + \|\tilde{\phi}_0(x^{(0)})\|_{\mathfrak{g}_0} \le (1+2432/5r)\|x-x^{(0)}\|_{\mathfrak{g}_0} < (1+2432/5r)r\,. \tag{A.46}$$

Since $r \le 1/11$, we have $1+(1-r)^{-2} \le 5/2$, and by combining (A.44),(A.45) and(A.46), we obtain

$$\|D\tilde{\phi}_0(x)[w]\| \le (5h/4)\|D\mathrm{G}_{h,x^{(0)}}(x)[w]\|_{\mathfrak{g}_0^{-1}} + 4874(1-r)^{-7}r\|w\|_{\mathfrak{g}_0}\,.$$

Since $\|D\tilde{\phi}_0(x)[w]\|_{\mathfrak{g}_0} > \|w\|_{\mathfrak{g}_0}/2$ and $(1-r)^{-7} \le 2$ with $r \le 1/11$, the last inequality becomes

$$\begin{aligned}(5h/4)\|D\mathrm{G}_{h,x^{(0)}}(x)[w]\|_{\mathfrak{g}_0^{-1}} &\ge \{1/2 - 9748r\}\|w\|_{\mathfrak{g}_0}\,,\\ &\ge 1/2 - r/(4r_\alpha^\star) \ge 1/4\,. \qquad \text{(see (A.28))}\end{aligned}$$

In particular, $\mathrm{Jac}[\mathrm{G}_{h,x^{(0)}}](x)$ is invertible, which concludes the proof. □

We prove below that, for any $z^{(0)} \in \mathrm{T}^\star\mathsf{M}$, the set $\underline{\mathrm{G}}_h(z^{(0)})$ is reduced to a single point, for h small enough depending on $z^{(0)}$.

Lemma 44. *For any $w \ge 0$, any $r \in (0,1)$ and any $\alpha \ge 1$, we define*

$$h_1(w,r,\alpha) = \min\left(\frac{1}{1+(1-r)^{-2}+2(2r+w)}, \frac{(1-r)^3}{3(r+w)}, \frac{r}{(r+w)(1+3r/2)+1/2(r+w)^2}\right)\,. \tag{A.47}$$

*We recall that the set-valued map $\underline{\mathrm{G}}_h$ is defined in (A.24). Assume **A**0, **A**1. Then, for any $r \in (0,1/11]$, any $z=(x,p) \in \mathrm{T}^\star\mathsf{M}$ and any $h \in (0, h_1(\|p\|_{\mathfrak{g}(x)^{-1}}, r, \alpha))$, there exists a unique $z' \in \mathsf{B}_{\|\cdot\|_z}(z,r) \subset \mathrm{T}^\star\mathsf{M}$ such that $z' \in \underline{\mathrm{G}}_h(z)$.*

Proof. Assume **A**0, **A**1. Let $r \in (0, 1/11]$, $z = (x, p) \in \mathrm{T}^\star\mathsf{M}$ and $h \in (0, h_1(\|p\|_{\mathfrak{g}(x)^{-1}}, r, \alpha))$, where h_1 is defined in (A.47). We recall that the map $\underline{\mathrm{g}}_h$ is defined in (A.25). We define $\mathsf{B} = \mathsf{B}_{\|\cdot\|_z}(z, r)$ and the map $\underline{\mathrm{g}}_{h,z} : \mathrm{T}^\star\mathsf{M} \to \mathbb{R}^n \times \mathbb{R}^n$ by $\underline{\mathrm{g}}_{h,z}(z') = z' - \underline{\mathrm{g}}_h(z, z')$ for any $z' \in \mathrm{T}^\star\mathsf{M}$. Then, $\underline{\mathrm{g}}_{h,z}(z') = z'$ if and only if $z' \in \underline{\mathrm{G}}_h(z)$. The proof is divided in two steps.

Step 1. We first prove that $\underline{\mathrm{g}}_{h,z}(\mathsf{B}) \subset \mathsf{B}$. Let $z' \in \mathsf{B}$, we have by Lemma 30-(a)

$$\begin{aligned}
\|\underline{\mathrm{g}}_{h,z}(z') - z\|_z &= \tfrac{h}{2}\|\{\mathfrak{g}(x)^{-1} + \mathfrak{g}(x')^{-1}\}p'\|_{\mathfrak{g}(x)} + \tfrac{h}{4}\|D\mathfrak{g}(x)[\mathfrak{g}(x)^{-1}p', \mathfrak{g}(x)^{-1}p']\|_{\mathfrak{g}(x)^{-1}} \\
&= \tfrac{h}{2}\|2\mathfrak{g}(x)^{-1}p' + \{\mathfrak{g}(x')^{-1} - \mathfrak{g}(x)^{-1}\}p'\|_{\mathfrak{g}(x)} + \tfrac{h}{4}\|D\mathfrak{g}(x)[\mathfrak{g}(x)^{-1}p', \mathfrak{g}(x)^{-1}p']\|_{\mathfrak{g}(x)^{-1}} \\
&\le \tfrac{h}{2}(2\|p'\|_{\mathfrak{g}(x)^{-1}} + \|\mathfrak{g}(x)^{1/2}(\mathfrak{g}(x')^{-1} - \mathfrak{g}(x)^{-1})\mathfrak{g}(x)^{1/2}\|_2\|p'\|_{\mathfrak{g}(x)^{-1}}) + \tfrac{h}{2}\|p'\|^2_{\mathfrak{g}(x)^{-1}} ,
\end{aligned}$$

where the following inequalities hold

(a) $\|p'\|_{\mathfrak{g}(x)^{-1}} = \|p' - p + p\|_{\mathfrak{g}(x)^{-1}} \le r + \|p\|_{\mathfrak{g}(x)^{-1}}$.

(b) $\|\mathfrak{g}(x)^{1/2}(\mathfrak{g}(x')^{-1} - \mathfrak{g}(x)^{-1})\mathfrak{g}(x)^{1/2}\|_2 \le 3\|x - x'\|_{\mathfrak{g}(x)} \le 3r$, on the model of (A.17), using $r \le 1/11$.

Therefore, we have

$$\begin{aligned}
\|\underline{\mathrm{g}}_{h,z}(z') - z\|_z &\le h((r + \|p\|_{\mathfrak{g}(x)^{-1}}) + \tfrac{3}{2}r(r + \|p\|_{\mathfrak{g}(x)^{-1}}) + \tfrac{1}{2}(r + \|p\|_{\mathfrak{g}(x)^{-1}})^2) \\
&\le h((r + \|p\|_{\mathfrak{g}(x)^{-1}})(1 + \tfrac{3}{2}r) + \tfrac{1}{2}(r + \|p\|_{\mathfrak{g}(x)^{-1}})^2) \\
&\le hr/h_1(\|p\|_{\mathfrak{g}(x)^{-1}}, r, \alpha) < r ,
\end{aligned}$$

which proves the statement.

Step 2. We now prove that $\underline{\mathrm{g}}_{h,z}$ is a contraction on B. This proof notably recovers some elements of the proof of Proposition 34 (see Appendix A.5). Let $(z_1, z_2) \in \mathsf{B} \times \mathsf{B}$ with $z_1 = (x_1, p_1)$ and $z_2 = (x_2, p_2)$. Remark that $x_1, x_2 \in \mathsf{W}^0(x, 1) \times \mathsf{W}^0(x, 1)$. Let us first bound $\|\underline{\mathrm{g}}^{(1)}_{h,z}(z_1) - \underline{\mathrm{g}}^{(1)}_{h,z}(z_2)\|_{\mathfrak{g}(x)}$. We have

$$\begin{aligned}
\underline{\mathrm{g}}^{(1)}_{h,z}(z_1) - \underline{\mathrm{g}}^{(1)}_{h,z}(z_2) &= \tfrac{h}{2}\{\mathfrak{g}(x)^{-1}(p_1 - p_2) + \mathfrak{g}(x_1)^{-1}p_1 - \mathfrak{g}(x_2)^{-1}p_2\} \\
&= \tfrac{h}{2}\{\mathfrak{g}(x)^{-1}(p_1 - p_2) + \mathfrak{g}(x_1)^{-1}(p_1 - p_2) + (\mathfrak{g}(x_1)^{-1} - \mathfrak{g}(x_2)^{-1})p_2\} ,
\end{aligned}$$

and then, since $r \le 1/11$, we have on the model of (A.17)

$$\begin{aligned}
\|\underline{\mathrm{g}}^{(1)}_{h,z}(z_1) - \underline{\mathrm{g}}^{(1)}_{h,z}(z_2)\|_{\mathfrak{g}(x)} &\le \tfrac{h}{2}\{1 + \|\mathfrak{g}(x)^{1/2}\mathfrak{g}(x_1)^{-1}\mathfrak{g}(x)^{1/2}\|_2\}\|p_1 - p_2\|_{\mathfrak{g}(x)^{-1}} \\
&\quad + \tfrac{h}{2}\|\mathfrak{g}(x)^{1/2}(\mathfrak{g}(x_1)^{-1} - \mathfrak{g}(x_2)^{-1})\mathfrak{g}(x)^{1/2}\|_2\|p_2\|_{\mathfrak{g}(x)^{-1}} \\
&\le \tfrac{h}{2}\{1 + (1 - r)^{-2}\}\|p_1 - p_2\|_{\mathfrak{g}(x)^{-1}} + \tfrac{3h}{2}(1 - r)^{-3}\|x_1 - x_2\|_{\mathfrak{g}(x)}(r + \|p\|_{\mathfrak{g}(x)^{-1}}) .
\end{aligned}$$

Let us now bound $\|\underline{\mathrm{g}}^{(2)}_{h,z}(z_1) - \underline{\mathrm{g}}^{(2)}_{h,z}(z_2)\|_{\mathfrak{g}(x)^{-1}}$. We have

$$\underline{\mathrm{g}}^{(2)}_{h,z}(z_1) - \underline{\mathrm{g}}^{(2)}_{h,z}(z_2) = \tfrac{h}{4}\{D\mathfrak{g}(x)[\mathfrak{g}(x)^{-1}p_1, \mathfrak{g}(x)^{-1}p_1] - D\mathfrak{g}(x)[\mathfrak{g}(x)^{-1}p_2, \mathfrak{g}(x)^{-1}p_2]\} ,$$

which gives using Lemma 30-(a)

$$\begin{aligned}
\|\underline{\mathrm{g}}^{(2)}_{h,z}(z_1) - \underline{\mathrm{g}}^{(2)}_{h,z}(z_2)\|_{\mathfrak{g}(x)^{-1}} &= \tfrac{h}{2}\|D\mathfrak{g}(x)[\mathfrak{g}(x)^{-1}(p_1 - p_2), \mathfrak{g}(x)^{-1}p_2]\|_{\mathfrak{g}(x)^{-1}} \\
&\quad + \tfrac{h}{4}\|D\mathfrak{g}(x)[\mathfrak{g}(x)^{-1}(p_1 - p_2), \mathfrak{g}(x)^{-1}(p_1 - p_2)]\|_{\mathfrak{g}(x)^{-1}} \\
&\le h\|\mathfrak{g}(x)^{-1}(p_1 - p_2)\|_{\mathfrak{g}(x)}\|\mathfrak{g}(x)^{-1}p_2\|_{\mathfrak{g}(x)} + \tfrac{h}{2}\|\mathfrak{g}(x)^{-1}(p_1 - p_2)\|^2_{\mathfrak{g}(x)} \\
&\le h(r + \|p\|_{\mathfrak{g}(x)^{-1}})\|p_1 - p_2\|_{\mathfrak{g}(x)^{-1}} + hr\|p_1 - p_2\|_{\mathfrak{g}(x)^{-1}} .
\end{aligned}$$

Finally, it comes that

$$\begin{aligned}
\|\underline{\mathrm{g}}_{h,z}(z_1) - \underline{\mathrm{g}}_{h,z}(z_2)\|_z &\le \tfrac{h}{2}\{1 + (1 - r)^{-2} + 4r + 2\|p\|_{\mathfrak{g}(x)^{-1}}\}\|p_1 - p_2\|_{\mathfrak{g}(x)^{-1}} \\
&\quad + \tfrac{3h}{2}(1 - r)^{-3}(r + \|p\|_{\mathfrak{g}(x)^{-1}})\|x_1 - x_2\|_{\mathfrak{g}(x)} \\
&\le (1/2)h\{\|x_1 - x_2\|_{\mathfrak{g}(x)} + \|p_1 - p_2\|_{\mathfrak{g}(x)^{-1}}\}/h_1(\|p\|_{\mathfrak{g}(x)^{-1}}, r, \alpha) \\
&< (1/2)\|z_1 - z_2\|_z,
\end{aligned}$$

which proves that $\underline{\mathrm{g}}_{h,z}$ is a contraction on B.

Conclusion. We obtain the result of Lemma 44 by applying the fixed point theorem on $\underline{\mathrm{g}}_{h,z}$ and B. □

Elaborating on Lemma 44, we prove in Lemma 45 that the only element of $\underline{\mathrm{G}}_h(z^{(0)})$ verifies smoothness properties if h is chosen small enough, depending on $z^{(0)}$.

Lemma 45. *For any $z=(x,p)\in \mathrm{T}^\star\mathsf{M}$, any $r\in(0,1)$ and any $\alpha\geq 1$, we define*

$$\bar{h}(z,r,\alpha)=\min\left(h_1(\|p\|_{\mathfrak{g}(x)^{-1}},r,\alpha),h_2(z),h_3(z,r)\right) \tag{A.48}$$

where h_1 is defined in (A.47), and h_2 and h_3 are defined by

$$h_2(z)=\frac{1}{3/2+3\|p\|_{\mathfrak{g}(x)^{-1}}}\,,\qquad h_3(z,r)=\frac{1}{1+(1-r)^{-1}(r+\|p\|_{\mathfrak{g}(x)^{-1}})}\,.$$

*We recall that $r_\alpha^\star$ is defined in (A.28). We also recall that the maps $\underline{\mathrm{G}}_h$, $\underline{\mathrm{g}}_h$, $\overline{\mathrm{G}}_h$ and $\mathrm{G}_{h,x^{(0)}}$ are respectively defined in (A.24), (A.25), (A.26) and (A.27). Assume **A**0, **A**1. Then, for any $z^{(0)}\in\mathrm{T}^\star\mathsf{M}$ and any $h\in(0,\bar{h}(z^{(0)},r_\alpha^\star,\alpha))$, there exists a unique element $z_h^{(1/2)}\in\mathrm{T}^\star\mathsf{M}$ such that $z_h^{(1/2)}\in\underline{\mathrm{G}}_h(z^{(0)})\cap\mathsf{B}_{\|\cdot\|_{z^{(0)}}}(z^{(0)},r_\alpha^\star)$. Moreover, we have*

(a) $\mathrm{Jac}_{z'}[\underline{\mathrm{g}}_h](z^{(0)},z_h^{(1/2)})$ and $\mathrm{Jac}[\overline{\mathrm{G}}_h](z_h^{(1/2)})$ are invertible,

(b) $\mathrm{Jac}[\mathrm{G}_{h,x^{(0)}}](x_h^{(1/2)})$ is invertible.

Proof. Assume **A**0, **A**1. Let $z^{(0)}\in\mathrm{T}^\star\mathsf{M}$ and $h\in(0,\bar{h}(z^{(0)},r_\alpha^\star,\alpha))$, where $\bar{h}$ is defined in (A.48). Since $r_\alpha^\star\leq 1/11$, Lemma 44 ensures the existence and uniqueness of $z_h^{(1/2)}=(x_h^{(1/2)},p_h^{(1/2)})\in\mathrm{T}^\star\mathsf{M}$ such that $z_h^{(1/2)}\in\underline{\mathrm{G}}_h(z^{(0)})\cap\mathsf{B}_{\|\cdot\|_{z^{(0)}}}(z^{(0)},r_\alpha^\star)$. Moreover, we have $x_h^{(1/2)}\in\mathsf{W}^0(x^{(0)},r_\alpha^\star)$, and thus $\mathrm{Jac}[\mathrm{G}_{h,x^{(0)}}](x_h^{(1/2)})$ is invertible by Lemma 43. Let us prove that $\mathrm{Jac}_{z'}[\underline{\mathrm{g}}_h](z^{(0)},z_h^{(1/2)})$ and $\mathrm{Jac}[\overline{\mathrm{G}}_h](z_h^{(1/2)})$ are invertible.

Invertibility of $\mathrm{Jac}_{z'}[\underline{\mathrm{g}}_h](z^{(0)},z_h^{(1/2)})$. We first remark that $\mathrm{Jac}_{z'}[\underline{\mathrm{g}}_h](z^{(0)},z_h^{(1/2)})=\mathrm{Jac}[\psi_{0,h}](z_h^{(1/2)})$ where $\psi_{0,h}=\mathrm{Id}+\frac{h}{2}\psi_0$ and ψ_0 is defined on $\mathrm{T}^\star\mathsf{M}$ by

$$\psi_0(z)=(-\{\mathfrak{g}(x^{(0)})^{-1}+\mathfrak{g}(y)^{-1}\}^{-1}p,-\tfrac{1}{2}D\mathfrak{g}(x^{(0)})[\mathfrak{g}(x^{(0)})^{-1}p,\mathfrak{g}(x^{(0)})^{-1}p])\,.$$

We recall that $\|z_h^{(1/2)}-z^{(0)}\|_{z^{(0)}}<r_\alpha^\star\leq 1/11$ and thus define $r_1=1/11-\|z_h^{(1/2)}-z^{(0)}\|_{z^{(0)}}>0$ and $\mathsf{B}_1=\mathsf{B}_{\|\cdot\|_{z^{(0)}}}(z_h^{(1/2)},r_1)$. We set $\tilde{r}=1/11$. Note that $\mathsf{B}_1\subset\mathsf{B}_{\|\cdot\|_{z^{(0)}}}(z^{(0)},1/11)$. We show below that ψ_0 is Lipschitz-continuous on B_1 with respect to $\|\cdot\|_{z^{(0)}}$. Let $(z,z')\in\mathsf{B}_1\times\mathsf{B}_1$. Similarly to the proof of Lemma 44, we have

$$\begin{aligned}\|\psi_0^{(1)}(z)-\psi_0^{(1)}(z')\|_{\mathfrak{g}(x^{(0)})}&\leq\|p-p'\|_{\mathfrak{g}(x^{(0)})^{-1}}\\&\quad+\|\mathfrak{g}(x^{(0)})^{1/2}\mathfrak{g}(x)^{-1}\mathfrak{g}(x^{(0)})^{1/2}\|_2\|p-p'\|_{\mathfrak{g}(x^{(0)})^{-1}}\\&\quad+\|\mathfrak{g}(x^{(0)})^{1/2}\{\mathfrak{g}(x)^{-1}-\mathfrak{g}(x')^{-1}\}\mathfrak{g}(x^{(0)})^{1/2}\|_2\|p\|_{\mathfrak{g}(x^{(0)})^{-1}}\\&\leq(1+(1-\tilde{r})^{-2})\|p-p'\|_{\mathfrak{g}(x^{(0)})^{-1}}\\&\quad+3(1-\tilde{r})^{-3}(\tilde{r}+\|p^{(0)}\|_{\mathfrak{g}(x^{(0)})^{-1}})\|x-x'\|_{\mathfrak{g}(x^{(0)})}\,.\end{aligned}$$

In the same manner, we have

$$\|\psi_0^{(2)}(z)-\psi_0^{(2)}(z')\|_{\mathfrak{g}(x^{(0)})^{-1}}\leq 2(2\tilde{r}+\|p^{(0)}\|_{\mathfrak{g}(x^{(0)})^{-1}})\|p-p'\|_{\mathfrak{g}(x^{(0)})^{-1}}\,.$$

Therefore, recalling that $\tilde{r}=1/11$, we obtain with the previous inequalities

$$\begin{aligned}\|\psi_0(z)-\psi_0(z')\|_{z^{(0)}}&\leq\{3+6\|p^{(0)}\|_{\mathfrak{g}(x^{(0)})^{-1}}\}\|z-z'\|_{z^{(0)}}\\&\leq 2\|z-z'\|_{z^{(0)}}/h_2(z^{(0)})\,.\end{aligned}$$

Hence, ψ_0 is $(2/h_2(z^{(0)}))$-Lipschitz-continuous on B_1 with respect to $\|\cdot\|_{z^{(0)}}$. Then, since $h<h_2(z^{(0)})$, $\mathrm{Jac}[\psi_{0,h}](z)$ is invertible for any $z\in\mathsf{B}_1$ by Lemma 42. In particular, $\mathrm{Jac}_{z'}[\underline{\mathrm{g}}_h](z^{(0)},z_h^{(1/2)})=\mathrm{Jac}[\psi_{0,h}](z_h^{(1/2)})$ is invertible.

Invertibility of $\mathrm{Jac}[\overline{\mathrm{G}}_h](z_h^{(1/2)})$. Let us remark that $\overline{\mathrm{G}}_h = \mathrm{Id} + h\zeta$ where ζ is defined on $\mathrm{T}^\star\mathsf{M}$ by

$$\zeta(z) = (0, \frac{1}{4} D\mathfrak{g}(x)[\mathfrak{g}(x)^{-1}p, \mathfrak{g}(x)^{-1}p]).$$

We define $r_2 = 1/2$ and $\mathsf{B}_2 = \mathsf{B}_{\|\cdot\|_{z_h^{(1/2)}}}(z_h^{(1/2)}, r_2)$. We show below that ζ is Lipschitz-continuous on B_2 with respect to $\|\cdot\|_{z_h^{(1/2)}}$, with a Lipschitz constant which does not depend on $z_h^{(1/2)}$. Let $(z, z') \in \mathsf{B}_2 \times \mathsf{B}_2$. Using Lemma 30-(a) with $r_2 = 1/2$, it is clear that

$$\|\zeta(z) - \zeta(z')\|_{z_h^{(1/2)}} \leq (1 + \|p_h^{(1/2)}\|_{\mathfrak{g}(x_h^{(1/2)})^{-1}}) \|z - z'\|_{z_h^{(1/2)}} .$$

Moreover, since $\|z_h^{(1/2)} - z^{(0)}\|_{z^{(0)}} < r_\alpha^\star$, we have by Lemma 33

$$\begin{aligned} \|p_h^{(1/2)}\|_{\mathfrak{g}(x_h^{(1/2)})^{-1}} &\leq (1 - r_\alpha^\star)^{-1} \|p_h^{(1/2)}\|_{\mathfrak{g}(x^{(0)})^{-1}} \\ &\leq (1 - r_\alpha^\star)^{-1} (r_\alpha^\star + \|p^{(0)}\|_{\mathfrak{g}(x^{(0)})^{-1}}) , \end{aligned}$$

and thus

$$\begin{aligned} \|\zeta(z) - \zeta(z')\|_{z_h^{(1/2)}} &\leq \{1 + (1 - r_\alpha^\star)^{-1} (r_\alpha^\star + \|p^{(0)}\|_{\mathfrak{g}(x^{(0)})^{-1}})\} \|z - z'\|_{z_h^{(1/2)}} \\ &\leq \|z - z'\|_{z_h^{(1/2)}} / h_3(z^{(0)}, r_\alpha^\star) . \end{aligned}$$

Hence, ζ is $(1/h_3(z^{(0)}, r_\alpha^\star))$-Lipschitz-continuous on B_2 with respect to $\|\cdot\|_{z_h^{(1/2)}}$. Since $h < h_3(z^{(0)}, r_\alpha^\star)$, $\mathrm{Jac}[\overline{\mathrm{G}}_h](z)$ is invertible for any $z \in \mathsf{B}_2$ by Lemma 42. In particular, $\mathrm{Jac}[\overline{\mathrm{G}}_h](z_h^{(1/2)})$ is invertible, which concludes the proof. □

The following lemma states the existence of a diffeomorphism between $z^{(0)} \in \mathrm{T}^\star\mathsf{M}$ and $z^{(1)} \in \mathrm{G}_h(z^{(0)})$ under smoothness conditions verified by $z^{(1)}$.

Lemma 46. *Let $(z^{(0)}, z^{(1)}) \in \mathrm{T}^\star\mathsf{M} \times \mathrm{T}^\star\mathsf{M}$. We recall that the maps $\underline{\mathrm{g}}_h$, $\overline{\mathrm{G}}_h$ and G_h are respectively defined in (A.25), (A.26) and (2.14). Assume that $\mathfrak{g} \in \mathrm{C}^2(\mathsf{M}, \mathbb{R}^{d\times d})$ and that there exist $h > 0$ and $z_h^{(1/2)} \in \mathrm{T}^\star\mathsf{M}$ with the following properties:*

(a) $\underline{\mathrm{g}}_h(z^{(0)}, z_h^{(1/2)}) = 0$ and $z^{(1)} = \overline{\mathrm{G}}_h(z_h^{(1/2)})$.

(b) $\mathrm{Jac}_{z'}[\underline{\mathrm{g}}_h](z^{(0)}, z_h^{(1/2)})$ and $\mathrm{Jac}[\overline{\mathrm{G}}_h](z_h^{(1/2)})$ are invertible.

Then, there exists a neighbourhood $\mathsf{U} \subset \mathrm{T}^\star\mathsf{M}$ of $z^{(0)}$ and a C^1-diffeomorphism $\xi : \mathsf{U} \to \xi(\mathsf{U}) \subset \mathrm{T}^\star\mathsf{M}$ such that (i) $\xi(z^{(0)}) = z^{(1)}$, (ii) for any $z \in \mathsf{U}$, $\xi(z) \in \mathrm{G}_h(z)$ and (iii) $|\det \mathrm{Jac}\, \xi| \equiv 1$.

Proof. Let $(z^{(0)}, z^{(1)}) \in \mathrm{T}^\star\mathsf{M} \times \mathrm{T}^\star\mathsf{M}$. Assume that $\mathfrak{g} \in \mathrm{C}(\mathsf{M}, \mathbb{R}^{d\times d})$ and consider $h > 0$ and $z_h^{(1/2)} \in \mathrm{T}^\star\mathsf{M}$ as described in Lemma 46. Under the assumption on $\mathfrak{g}$, $\underline{\mathrm{g}}_h$ and $\overline{\mathrm{G}}_h$ are continuously differentiable respectively on $\mathrm{T}^\star\mathsf{M} \times \mathrm{T}^\star\mathsf{M}$ and $\mathrm{T}^\star\mathsf{M}$. We are going to prove Lemma 46, by first deriving intermediary results on $\underline{\mathrm{g}}_h$ and $\overline{\mathrm{G}}_h$.

Result on $\underline{\mathrm{g}}_h$. We recall that $\underline{\mathrm{g}}_h(z^{(0)}, z_h^{(1/2)}) = 0$ and $\mathrm{Jac}_{z'}[\underline{\mathrm{g}}_h](z^{(0)}, z_h^{(1/2)})$ is invertible. Then, by applying the implicit function theorem on $\underline{\mathrm{g}}_h$ at $(z^{(0)}, z_h^{(1/2)})$, we obtain the existence of a neighbourhood $\mathsf{U}_0 \subset \mathrm{T}^\star\mathsf{M}$ of $z^{(0)}$ and a C^1-diffeomorphism $\xi_0 : \mathsf{U}_0 \to \xi_0(\mathsf{U}_0) \subset \mathrm{T}^\star\mathsf{M}$ such that $\xi_0(z^{(0)}) = z_h^{(1/2)}$ and for any $z \in \mathsf{U}_0$, $\underline{\mathrm{g}}_h(z, \xi_0(z)) = 0$, *i.e.*, $\xi_0(z) \in \underline{\mathrm{G}}_h(z)$. Moreover, for any $z \in \mathsf{U}_0$, the Jacobian of ξ_0 at z is given by

$$\mathrm{Jac}[\xi_0](z) = -\mathrm{Jac}_{z'}[\underline{\mathrm{g}}_h](z, \xi_0(z))^{-1} \mathrm{Jac}_z[\underline{\mathrm{g}}_h](z, \xi_0(z)) .$$

Result on $\overline{\mathrm{G}}_h$. We recall that $z^{(1)} = \overline{\mathrm{G}}_h(z_h^{(1/2)})$ and $\mathrm{Jac}[\overline{\mathrm{G}}_h](z_h^{(1/2)})$ is invertible. Then, we apply the inverse function theorem on $\overline{\mathrm{G}}_h$ at $z_h^{(1/2)}$ and obtain the existence of a neighbourhood $\mathsf{U}_1 \subset \mathrm{T}^\star\mathsf{M}$ of $z_h^{(1/2)}$ such that $\xi_1 = \overline{\mathrm{G}}_h|_{\mathsf{U}_1}$ is a C^1-diffeomorphism on U_1.

Final result. We now consider the subset U and the map ξ, respectively defined by

(a) $\mathsf{U} = \xi_0^{-1}\left(\mathsf{U}_1 \cap \xi_0(\mathsf{U}_0)\right) \subset \mathrm{T}^\star\mathsf{M}$, neighbourhood of $z^{(0)}$.

(b) $\xi = \xi_1|_{\xi_0(\mathsf{U})} \circ \xi_0$, C^1-diffeomorphism on U such that $\xi(z^{(0)}) = z^{(1)}$, and for any $z \in \mathsf{U}$, $\xi(z) \in \mathrm{G}_h(z)$.

Let us now prove that $|\det \mathrm{Jac}\, \xi| \equiv 1$. Let $z \in \mathsf{U}$. We define $z_0 = \xi_0(z)$. By the chain rule, we have

$$\begin{aligned}\mathrm{Jac}[\xi](z) &= \mathrm{Jac}[\xi_1 \circ \xi_0](z)\\ &= \mathrm{Jac}[\xi_1](\xi_0(z))\,\mathrm{Jac}[\xi_0](z)\\ &= -\,\mathrm{Jac}[\xi_1](\xi_0(z))\,\mathrm{Jac}_{z'}[\underline{\mathrm{g}}_h](z,\xi_0(z))^{-1}\,\mathrm{Jac}_z[\underline{\mathrm{g}}_h](z,\xi_0(z))\,,\end{aligned}$$

where we have

(a) $\mathrm{Jac}[\xi_1](z') = \begin{pmatrix} \mathrm{I}_d & 0 \\ -\frac{h}{2}\partial_{x,x}H_2(z') & \mathrm{I}_d - \frac{h}{2}\partial_{x,p}H_2(z') \end{pmatrix}$,

(b) $\mathrm{Jac}_{z'}[\underline{\mathrm{g}}_h](z,z') = \begin{pmatrix} \mathrm{I}_d - \frac{h}{2}\partial_{x,p}H_2(z') & -\frac{h}{2}\{\partial_{p,p}H_2(x,p') + \partial_{p,p}H_2(x',p')\} \\ 0 & \mathrm{I}_d + \frac{h}{2}\partial_{x,p}H_2(x,p') \end{pmatrix}$,

(c) $\mathrm{Jac}_z[\underline{\mathrm{g}}_h](z,z') = \begin{pmatrix} -\mathrm{I}_d - \frac{h}{2}\partial_{x,p}H_2(x,p') & 0 \\ \frac{h}{2}\partial_{x,x}H_2(x,p') & -\mathrm{I}_d \end{pmatrix}$.

Therefore, we obtain

$$\begin{aligned}&|\det \mathrm{Jac}[\xi](z)|\\ &\quad= \det(\mathrm{I}_d - \frac{h}{2}\partial_{x,p}H_2(z_0))\det(\{\mathrm{I}_d - \frac{h}{2}\partial_{x,p}H_2(z_0)\}\{\mathrm{I}_d + \frac{h}{2}\partial_{x,p}H_2(x,p_0)\})^{-1}\det(\mathrm{I}_d + \frac{h}{2}\partial_{x,p}H_2(x,p_0)) = 1\,,\end{aligned}$$

which concludes the proof. □

Lemma 47. *Let $h > 0$. We recall that the set-valued map F_h is defined in Section 2.3.1. Let $z \in \mathrm{T}^\star\mathsf{M}$. Assume that there exist $(z', z'') \in \mathrm{T}^\star\mathsf{M} \times \mathrm{T}^\star\mathsf{M}$ such that*

(a) $z' \in \mathrm{F}_h(z)$,

(b) $z'' \in \mathrm{F}_h(z)$, and

(c) $x' = x''$.

Then, we have $p' = p''$, and thus $z' = z''$.

Proof. Let $h > 0$. Let $z \in \mathrm{T}^\star\mathsf{M}$. We consider $(z', z'') \in \mathrm{F}_h(z) \times \mathrm{F}_h(z)$ such that $x' = x'' = \tilde{x}$. By using the last two aligns from (2.14) with z' and z'', we have $p' = -p^{(1/2)} + \frac{h}{2}\mathfrak{g}(\tilde{x})^{-1}p^{(1/2)} = p''$, where $p^{(1/2)} = \frac{2}{h}(\mathfrak{g}(x)^{-1} + \mathfrak{g}(\tilde{x})^{-1})^{-1}(\tilde{x} - x)$, which concludes the proof. □

Under the assumption **A**1, we define, for any $z^{(0)} \in \mathrm{T}^\star\mathsf{M}$, $h^\star(z^{(0)}) = \bar{h}(s(z^{(0)}), r_\alpha^\star, \alpha)$, where α is given by **A**1, and $r_\alpha^\star$ and $\bar{h}$ are respectively defined in (A.28) and (A.48). Note that $h^\star$ appears in Proposition 2, for which we derive the proof below.

Proof of Proposition 2. We recall that maps $\underline{\mathrm{G}}_h$, $\underline{\mathrm{g}}_h$ and $\overline{\mathrm{G}}_h$ are respectively defined in (A.24), (A.25) and (A.26). Assume **A**0, **A**1. Let $z^{(0)} \in \mathrm{T}^\star\mathsf{M}$. We define $\tilde{z}^{(0)} = s(z^{(0)})$. Let $h \in (0, h^\star(z^{(0)}))$. By using Lemma 45 on $\tilde{z}^{(0)}$, we obtain the existence of $z_h^{(1/2)} \in \mathrm{T}^\star\mathsf{M}$ with the following properties:

(a) $z_h^{(1/2)} \in \underline{\mathrm{G}}_h(\tilde{z}^{(0)})$, *i.e.*, $\underline{\mathrm{g}}_h(\tilde{z}^{(0)}, z_h^{(1/2)}) = 0$,

(b) $\mathrm{Jac}_{z'}[\underline{\mathrm{g}}_h](\tilde{z}^{(0)}, z_h^{(1/2)})$ and $\mathrm{Jac}[\overline{\mathrm{G}}_h](z_h^{(1/2)})$ are invertible,

(c) $\mathrm{Jac}[\mathrm{G}_{h,x^{(0)}}](x_h^{(1/2)})$ is invertible.

We then define $z_h^{(1)} = \overline{\mathrm{G}}_h(z_h^{(1/2)})$. In particular, $z_h^{(1)} \in \mathrm{G}_h(\tilde{z}^{(0)})$, *i.e.*, $z_h^{(1)} \in \mathrm{F}_h(z^{(0)})$ and $\mathrm{Jac}[\mathrm{G}_{h,x^{(0)}}](x_h^{(1)}) = \mathrm{Jac}[\mathrm{G}_{h,x^{(0)}}](x_h^{(1/2)})$ is invertible. By combining the properties of $z_h^{(1/2)}$ and with Lemma 46, we also obtain the existence of a neighbourhood $\mathsf{U}' \subset \mathrm{T}^\star\mathsf{M}$ of $\tilde{z}^{(0)}$ and a C^1-diffeomorphism $\xi_h : \mathsf{U}' \to \xi_h(\mathsf{U}') \subset \mathrm{T}^\star\mathsf{M}$ such that (i) $\xi_h(\tilde{z}^{(0)}) = z_h^{(1)}$, (ii) for any $z \in \mathsf{U}'$, $\xi_h(z) \in \mathrm{G}_h(z)$ and (iii) $|\det \mathrm{Jac}\, \xi_h| \equiv 1$. Therefore, we can define the subset U and the map γ_h of Proposition 2 by

(a) $\mathsf{U} = s(\mathsf{U}')$, neighbourhood of $z^{(0)}$ in $\mathrm{T}^\star\mathsf{M}$,

(b) $\gamma_h = \xi_h \circ s$, such that (i) $\gamma_h(z^{(0)}) = z_h^{(1)}$, (ii) for any $z \in \mathsf{U}$, $\gamma_h(z) \in \mathrm{F}_h(z)$ and (iii) $\det \mathrm{Jac}[\gamma_h] \equiv 1$.

We now prove that U can be reduced to a smaller subset such that $\gamma_h(z)$ is the only element of $\mathrm{F}_h(z)$ in $\gamma_h(\mathsf{U})$ for any $z \in \mathsf{U}$. Motivated by Lemma 43, we first define, for any $(x, x') \in \mathsf{M} \times \mathsf{M}$, $\mathrm{F}_{h,x}(x')$ as the only element $p \in \mathrm{T}_x^\star\mathsf{M}$ such that $(x', p') \in \mathrm{F}_h(x, p)$ for any $p' \in \mathrm{T}_{x'}^\star\mathsf{M}$. It is clear that $\mathrm{F}_{h,x}(x') = -\mathrm{G}_{h,x}(x')$, where $\mathrm{G}_{h,x}$ is defined in (A.27). We also define $\mathrm{Z}_h : \mathsf{M} \times \mathsf{M} \to \mathrm{T}^\star\mathsf{M}$ by

$$\mathrm{Z}_h(x, x') = (x, \mathrm{F}_{h,x}(x')) = (x, -\mathrm{G}_{h,x}(x')) \;,$$

which is continuously differentiable since $\mathfrak{g} \in \mathrm{C}^2(\mathsf{M}, \mathbb{R}^{d\times d})$. Besides this, we obtain by Lemma 43 that $\mathrm{Jac}[\mathrm{G}_{h,x^{(0)}}](x^{(0)}, x_h^{(1)})$ is invertible, and then $\mathrm{Jac}[\mathrm{Z}_h](x_h^{(1)})$ is invertible. Therefore, by applying the inverse function theorem on Z_h at $(x^{(0)}, x_h^{(1)})$, it comes that Z_h is a C^1-diffeomorphism in a neighbourhood of $(x^{(0)}, x_h^{(1)})$. We are now going to prove the result by contradiction. Assume for now that there is no neighbourhood U of $z^{(0)}$ such that $\gamma_h(z)$ is the only element of $\mathrm{F}_h(z)$ in $\gamma_h(\mathsf{U})$, for any $z \in \mathsf{U}$. Then, we can find a sequence $(z_i)_{i\in\mathbb{N}}$ which converges to $z^{(0)}$ such that for any $i \in \mathbb{N}$, there exist two different elements $z_{i,1} \in \mathrm{F}_h(z_i)$ and $z_{i,2} \in \mathrm{F}_h(z_i)$. Therefore, for any $i \in \mathbb{N}$, $x_{i,1} \neq x_{i,2}$ and by Lemma 47, we have

$$\mathrm{F}_{h,x_i}(x_{i,1}) = \mathrm{F}_{h,x_i}(x_{i,2}) = p_i \;,$$

and thus

$$Z_h(x_i, x_{i,1}) = Z_h(x_i, x_{i,2}) = z_i \;.$$

Moreover, by continuity of γ_h, the sequences $(z_{i,1})_{i\in\mathbb{N}}$ and $(z_{i,2})_{i\in\mathbb{N}}$ converge to $z_h^{(1)}$, and therefore, $(x_{i,1})_{i\in\mathbb{N}}$ and $(x_{i,2})_{i\in\mathbb{N}}$ also converge to $x^{(1)}$. Combined with (A.7), this result of convergence is in contradiction with the fact that Z_h is a diffeomorphism in a neighbourhood of $(x^{(0)}, x_h^{(1)})$. Therefore, we can reduce U to a smaller subset such that $\gamma_h(z)$ is the only element of $\mathrm{F}_h(z)$ in $\gamma_h(\mathsf{U})$ for any $z \in \mathsf{U}$, which concludes the proof of Proposition 2. □

A.8 Modification of n-BHMC algorithm with step-size conditioning

In the rest of the paper, for any $z^{(0)} \in \mathrm{T}^\star\mathsf{M}$, we will denote by $h_\star(z^{(0)})$ the value of $h_\star$ given by **A**2.

Beyond n-BHMC. A crucial part of the proof of reversibility in BHMC, as much for c-BHMC as for n-BHMC, relies on local *symplectic* properties of the integrator of the Hamiltonian dynamics. Although Algorithm 1 can be implemented without any practical limitation, it is hard to state such properties for its numerical integrator Φ_h under **A**0, **A**1 and **A**2, given *any* value of h. Indeed, we know from **A**2 that Φ_h is a local involution around $z^{(0)} \in \mathrm{T}^\star\mathsf{M}$ when $h < h_\star(z^{(0)})$; however, we cannot ensure this result when $h > h_\star(z^{(0)})$... To circumvent this issue, we propose to study Theoretical n-BHMC (Tn-BHMC), presented in Algorithm 3. In this modified version of n-BHMC, we actually enforce a condition on h to be small enough. We now get into the details of this new algorithm and assume **A**2 for the rest of this section.

Theoretical motivations. Let $h > 0$. We recall the definition of the set A_h introduced in Section 2.4.2

$$\mathsf{A}_h = \{z \in \mathrm{T}^\star\mathsf{M} \,:\, h < \min_{\tilde{z}\in\mathsf{B}_{\|\cdot\|_z}(z,1)} h_\star(\tilde{z})\} \;.$$

It is clear that $\mathsf{A}_h \subset \mathrm{dom}_{\Phi_h}$: indeed, if $z^{(0)} \in \mathsf{A}_h$, we have in particular that $h < h_\star(z^{(0)})$, and therefore $z^{(0)} \in \mathrm{dom}_{\Phi_h}$ by **A**2. Let $z^{(0)} \in \mathsf{A}_h$. By **A**2, we know that Φ_h is an involution on a neighbourhood of $z^{(0)}$; in particular, it comes that $(\Phi_h \circ \Phi_h)(z^{(0)}) = z^{(0)}$. Hence, the condition (b) of the "involution checking step" in Algorithm 1 is *de facto* satisfied. This naturally leads to replace the condition "$z \in \mathrm{dom}_{\Phi_h}$" by the more restrictive condition "$z \in \mathsf{A}_h$".

Algorithm 3: Tn-BHMC with Momentum Refreshment

Input: $(x_0, p_0) \in \mathrm{T}^\star\mathsf{M}$, $\beta \in (0,1]$, $N \in \mathbb{N}$, $h > 0$, $\eta > 0$, Φ_h with domain dom_{Φ_h}, A_h

Output: $(X_n, P_n)_{n\in[N]}$

1 **for** $n = 1, ..., N$ **do**
2 Step 1: $G_n \sim \mathrm{N}_x(0, \mathrm{I}_d)$, $\tilde{P}_n \leftarrow \sqrt{1-\beta}P_{n-1} + \sqrt{\beta}G_n$
3 Step 2: solving a discretized version of ODE (2.7)
4 $X'_n, P'_n \leftarrow X_{n-1}, \tilde{P}_n$; $X_n^{(0)}, P_n^{(0)} \leftarrow (s \circ \mathrm{S}_{h/2})(X_n, \tilde{P}_n)$
5 **if** $Z_n^{(0)} = (X_n^{(0)}, P_n^{(0)}) \in \mathsf{A}_h$ **then**
6 $Z_n^{(1)} = \Phi_h(Z_n^{(0)})$
7 **if** $Z_n^{(1)} \in \mathsf{A}_h$ **then**
8 $X'_n, P'_n \leftarrow (s \circ \mathrm{S}_{h/2})(Z_n^{(1)})$
9 **end**
10 **end**
11 Step 3: $A_n \leftarrow \min(1, \exp[-H(X'_n, P'_n) + H(X_{n-1}, \tilde{P}_n)])$
12 $U_n \sim U[0,1]$
13 **if** $U_n \leq A_n$ **then** $\bar{X}_n, \bar{P}_n \leftarrow X'_n, P'_n$
14 **else** $\bar{X}_n, \bar{P}_n \leftarrow X_{n-1}, \tilde{P}_n$
15 Step 4: $X_n, \hat{P}_n \leftarrow s(\bar{X}_n, \bar{P}_n)$
16 Step 5: $G'_n \sim \mathrm{N}_x(0, \mathrm{I}_d)$, $P_n \leftarrow \sqrt{1-\beta}\hat{P}_n + \sqrt{\beta}G'_n$
17 **end**

Implementation of Tn-BHMC. In Algorithm 3, we highlight in yellow the modifications of Tn-BHMC in contrast to n-BHMC. Namely, we replace

(a) "$Z_n^{(0)} \in \mathrm{dom}_{\Phi_h}$" (Line 5 in Algorithm 1) by "$Z_n^{(0)} \in \mathsf{A}_h$" (Line 5 in Algorithm 3).

(b) "$Z_n^{(1)} \in \mathrm{dom}_{\Phi_h}$" (Line 8 in Algorithm 1) by "$Z_n^{(1)} \in \mathsf{A}_h$" (Line 7 in Algorithm 3).

Note that there is no need to maintain the "involution checking step" of Algorithm 1, since it is automatically verified once $Z_n^{(0)} \in \mathsf{A}_h$. On the whole, these new conditions are more *restrictive* than the conditions of Algorithm 1 since $\mathsf{A}_h \subset \mathrm{dom}_{\Phi_h}$; moreover, they can be thought as conditions directly applied on the step-size h. The specific choice of A_h, instead of another subset of dom_{Φ_h}, is actually sufficient to derive the proof of reversibility of Tn-BHMC (see Section 2.4.2).

A.9 Proofs of Section 2.4.2

A.9.1 Expression and properties of $r^\star$

Given a Riemannian manifold $(\mathsf{M}, \mathfrak{g})$, we define $r^\star(x)$ for any $x \in \mathsf{M}$ by

$$r^\star(x) = \min(\|\mathfrak{g}(x)\|_2^{-1/2}, \|\mathfrak{g}(x)^{-1}\|_2^{-1/2}) \,. \tag{A.49}$$

Note that $r^\star$ is used in **A**2 and that $r^\star(x) = 1/C_x$, where C_x is defined in Lemma 33. We prove below that $r^\star$ has a smooth behaviour on M under our main assumptions.

Lemma 48. *Assume **A**0, **A**1. Also assume that $x \in \mathsf{M} \mapsto \|\mathfrak{g}^{-1}(x)\|_2$ is bounded from above. As defined in (A.49), $r^\star : \mathsf{M} \to (0, +\infty)$ satisfies the following properties:*

(a) $r^\star(x) \to 0$ as $x \to \partial M$.

(b) There exists $\mathtt{L} > 0$ such that $r^\star$ is $\mathtt{L}$-Lipschitz-continuous on M with respect to $\|\cdot\|_2$.

(c) There exists $\mathtt{M} > 0$ such that $r^\star(x) \leq \mathtt{M}$ for any $x \in \mathsf{M}$.

Proof. Assume **A**0, **A**1. Also assume that $x \in \mathsf{M} \mapsto \|\mathfrak{g}^{-1}(x)\|_2$ is bounded from above. We first define $r_1 : x \in \mathsf{M} \mapsto \|\mathfrak{g}(x)\|_2^{-1/2}$ and $r_2 : x \in \mathsf{M} \mapsto \|\mathfrak{g}(x)^{-1}\|_2^{-1/2}$, such that $r^\star = \min(r_1, r_2)$. Since $\mathfrak{g} \in \mathrm{C}^2(\mathsf{M}, \mathbb{R}^{d\times d})$, it is clear that r_1 and r_2 are continuously differentiable on M. We have:

(a) $r_1(x) \to 0$ as $x \to \partial\mathsf{M}$ by Lemma 29, and

(b) $r_2(x) \not\to 0$ as $x \to \partial\mathsf{M}$, since $1/r_2^2 : x \mapsto \|\mathfrak{g}(x)^{-1}\|_2$ is bounded on M.

Combining the fact that $r_2(x) > 0$ for any $x \in \mathsf{M}$ and Item (a), we obtain Item (a) of Lemma 48. We denote by d the distance induced by $\|\cdot\|_2$ and now define, for any $\varepsilon > 0$,

(i) $\mu_\varepsilon = \inf_{\{y \in \mathsf{M}\,:\, d(y,\partial\mathsf{M}) \leq \epsilon\}} r_2(y)$.

(ii) $\mathsf{M}_\varepsilon = \mathrm{Int}(\{x \in \mathsf{M} \,:\, d(x,\partial\mathsf{M}) \leq \varepsilon, r_1(x) \leq \mu_\varepsilon\})$, open set in M.

(iii) $\mathsf{M}_{-\varepsilon} = \mathsf{M} \backslash \mathsf{M}_\varepsilon$, closed and bounded (and thus, compact) set in M.

Using Items (a) and (b), we can ensure the existence of some $\varepsilon \in (0, \mathrm{diam}(\mathsf{M}))$ such that (i) $\mu_\varepsilon > 0$ and (ii) M_ε and $\mathsf{M}_{-\varepsilon}$ are not empty. We consider such ε for the rest of the proof.

Smoothness of r_2. We define $\delta = d(\mathsf{M}^{\mathrm{c}}, \mathsf{M}_{-\varepsilon})$ and $\mathsf{M}_\delta = \mathsf{M}^{\mathrm{c}} + \bar{\mathsf{B}}(0, \delta/4)$. Note that

(i) $\delta > 0$ since $\mathsf{M}_{-\varepsilon} \subset \mathsf{M}$,

(ii) M_δ is closed since the ball $\bar{\mathsf{B}}(0, \delta/4)$ is compact, and

(iii) $\mathsf{M}_\delta \cap \mathsf{M}_{-\varepsilon} = \emptyset$.

According to the smooth version Urysohn's lemma applied to M_δ and $\mathsf{M}_{-\varepsilon}$, there exists $\chi \in \mathrm{C}^1(\mathbb{R}^d, [0,1])$ such that $\chi(\mathsf{M}_{-\varepsilon}) = 1$ and $\chi(\mathsf{M}_\delta) = 0$. We then define $\tilde{r}_2 : \mathbb{R}^d \to (0, +\infty)$ by $\tilde{r}_2 = \chi r_2 + (1-\chi)\mu_\varepsilon$. In particular, (i) there exists $\mathsf{L}_2 > 0$ such that $\tilde{r}_2$ is L_2-Lipschitz-continuous on M with respect to $\|\cdot\|_2$, since $\tilde{r}_2 \in \mathrm{C}^1(\mathbb{R}^d, [0,1])$, and (ii) for any $x \in \mathsf{M}_\varepsilon$, $\tilde{r}_2(x) > \mu_\varepsilon$.

Smoothness of r_1. We now prove that r_1 is 1-Lipschitz continuous on M with respect to $\|\cdot\|_2$. Let $x \in \mathsf{M}$. Note that $r_1(x) = (\|\mathfrak{g}(x)\|_2^2)^{-1/4}$ and we thus have

$$\nabla r_1(x) = (-1/4)\|\mathfrak{g}(x)\|_2^{-5/2} h(x) : D\mathfrak{g}(x)\,,$$

where $h(x) = \partial_{\mathfrak{g}(x)}\|\mathfrak{g}(x)\|_2^2 = 2\|\mathfrak{g}(x)\|_2 u(x)u(x)^\top$, $u(x)$ being a normal eigenvector of $\mathfrak{g}(x)$ corresponding to the eigenvalue $\|\mathfrak{g}(x)\|_2$. Hence,

$$\begin{aligned}
\|\nabla r_1(x)\|_2 &\leq (1/2)\|\mathfrak{g}(x)\|_2^{-3/2}\|u(x)u(x)^\top : D\mathfrak{g}(x)\|_2 \\
&\leq (1/2)\|\mathfrak{g}(x)\|_2^{-3/2}\|u(x)u(x)^\top\|_2\|D\mathfrak{g}(x)\|_2 \\
&\leq (1/2)\|\mathfrak{g}(x)\|_2^{-3/2} \times 2\|\mathfrak{g}(x)\|_2^{3/2} \leq 1 \qquad \text{(Definition 1-(c))}
\end{aligned}$$

which proves the result on r_1.

Smoothness of r. Let $x \in \mathsf{M}$. We can face two cases: either, $x \in \mathsf{M}_{-\varepsilon}$, then $r_2(x) = \tilde{r}_2(x)$; or, $x \in \mathsf{M}_\varepsilon$, then $\tilde{r}_2(x) > \mu_\varepsilon \geq r_1(x)$ and $r_2(x) \geq r_1(x)$ by definition of M_ε. Thus, we have for any $x \in \mathsf{M}$, $r^\star(x) = \min(r_1(x), \tilde{r}_2(x))$ where r_1 and $\tilde{r}_2$ are respectively 1 and L_2 Lipschitz-continuous on M with respect to $\|\cdot\|_2$. By observing that $2\min(r_1, \tilde{r}_2) = r_1 + \tilde{r}_2 - |r_1 - \tilde{r}_2|$, we set $\mathsf{L} = 1 + \mathsf{L}_2$ and thus obtain Item (b) of Lemma 48. Finally, Item (c) of Lemma 48 directly derives from Item (b), since M is bounded. □

Note that the extra-assumption on $\mathfrak{g}^{-1}$ used in Lemma 48 is not directly ensured by self-concordance but can be proved when M is a polytope, as shown below.

Lemma 49. *Consider a polytope M defined by m constraints with $m > d$ such that $\mathsf{M} = \{x : \mathrm{A}x < b\}$, where $\mathrm{A} \in \mathbb{R}^{m \times d}$ is a full-rank matrix and $b \in \mathbb{R}^m$. We endow M with the Riemannian metric $\mathfrak{g}(x) = D^2\phi(x)$ where $\phi : \mathsf{M} \to \mathbb{R}$ is the logarithmic barrier given for any $x \in \mathsf{M}$ by $\phi(x) = -\sum_{i=1}^m \ln\left(b_i - \mathrm{A}_i^\top x\right)$. In particular, M and $\mathfrak{g}$ verify **A**0 and **A**1. Then, the function $r : \mathsf{M} \to (0, +\infty)$ defined by $r(x) = \|\mathfrak{g}^{-1}(x)\|_2$, for any $x \in \mathsf{M}$, is bounded from above.*

Proof. Consider such manifold M and metric $\mathfrak{g}$. We aim to show that the smallest eigenvalue of $\mathfrak{g}(x)$ is bounded from below for any $x \in \mathsf{M}$ by a constant $c > 0$, which does not depend on x, *i.e.*, for any $h \in \mathbb{R}^d$ and any $x \in \mathsf{M}$, $\mathfrak{g}(x)[h,h] \geq c\|h\|_2^2$.

Since A is full-ranked, $\mathrm{A}^\top\mathrm{A}$ is positive-definite. In particular, for any $h \in \mathbb{R}^d$, $(\mathrm{A}^\top\mathrm{A})[h,h] \geq \lambda_{\min}(\mathrm{A}^\top\mathrm{A})\|h\|_2^2$, where $\lambda_{\min}(\mathrm{A}^\top\mathrm{A}) > 0$ is the smallest eigenvalue of $\mathrm{A}^\top\mathrm{A}$. We recall that we have for any $x \in \mathsf{M}$, $\mathfrak{g}(x) = \mathrm{A}^\top S(x)^{-2}\mathrm{A}$, where $S(x) = \mathrm{Diag}(b_i - \mathrm{A}_i^\top x)_{i \in [m]}$. Let $i \in [m]$. The function $r_i : x \in \mathsf{M} \mapsto S(x)_{i,i}^{-2}$ has the following properties: (i) r_i is continuous on M, (ii) $r_i(x) > 0$ for any $x \in \mathsf{M}$ and (iii) $r(x) \to +\infty$ as $x \to \partial\mathsf{M}$.

Thus, there exists $c_i > 0$ such that for any $x \in \mathsf{M}$, $r_i(x) \geq c_i$. We define $\tilde{c} = \min_{i\in[m]} c_i$ and we have for any $x \in \mathsf{M}$, $S(x)^{-2} \succeq \tilde{c}\mathrm{I}_d$. We now define $c = \tilde{c}\lambda_{\min}(\mathrm{A}^\top \mathrm{A})$ and we have for any $x \in \mathsf{M}$

$$\mathfrak{g}(x)[h,h] \geq \tilde{c} \cdot (\mathrm{A}^\top \mathrm{A})[h,h] \geq c\|h\|_2^2 \, .$$

In particular, $\mathfrak{g}(x)^{-1} \preceq (1/c)\mathrm{I}_d$, *i.e.*, $\|\mathfrak{g}(x)^{-1}\|_2 \leq (1/c)$, which concludes the proof. □

A.9.2 Markov kernels of Algorithm 1

Based on the model of $\bar{\mathsf{E}}_h$ defined in (A.19), we define the set $\bar{\mathsf{E}}_h^\Phi \subset \mathrm{T}^\star\mathsf{M}$, which will ensure that the maps derived from implicit integrators of n-BHMC are properly expressed

$$\bar{\mathsf{E}}_h^\Phi = (s \circ \mathrm{S}_{h/2})^{-1}(\mathrm{dom}_{\Phi_h} \cap \Phi_h^{-1}(\mathrm{dom}_{\Phi_h})) = (s \circ \mathrm{S}_{h/2})(\mathrm{dom}_{\Phi_h} \cap \Phi_h^{-1}(\mathrm{dom}_{\Phi_h})) \, . \tag{A.50}$$

For any $(z, z') \in \bar{\mathsf{E}}_h^\Phi \times \mathrm{T}^\star\mathsf{M}$, we define the acceptance probability $\bar{a}(z'|z)$ to move from z to z' by

$$\bar{a}(z'|z) = a(z' \mid z)\mathbb{1}_{(s\circ \mathrm{S}_{h/2})(z)=(\Phi_h \circ \Phi_h \circ s \circ \mathrm{S}_{h/2})(z)} = a(z' \mid z)\mathbb{1}_{z=(\mathrm{R}_h^\Phi \circ \mathrm{R}_h^\Phi)(z)} \, ,$$

where $a(z' \mid z)$ is the acceptance probability defined in (2.19). We denote by $\mathrm{Q}_n : \mathrm{T}^\star\mathsf{M} \times \mathcal{B}(\mathrm{T}^\star\mathsf{M}) \to [0,1]$, the transition kernel of the (homogeneous) Markov chain $(x_n, p_n)_{n\in[N]}$ generated by Algorithm 1. We also denote by:

(a) $\mathrm{Q}_0 : \mathrm{T}^\star\mathsf{M} \times \mathcal{B}(\mathrm{T}^\star\mathsf{M}) \to [0,1]$, the transition kernel referring to Step 1 (also Step 5) in Algorithm 1, defined in (A.20).

(b) $\mathrm{Q}_{n,1} : \mathrm{T}^\star\mathsf{M} \times \mathcal{B}(\mathrm{T}^\star\mathsf{M}) \to [0,1]$, the transition kernel referring to Step 2-3-4 in Algorithm 1.

We provide below details on Markov kernels Q_n and $\mathrm{Q}_{n,1}$.

Kernel $\mathrm{Q}_{n,1}$. This kernel is deterministic and corresponds to the *numerical* integration of the Hamiltonian up until time h. For any $(z, z') \in \mathrm{T}^\star\mathsf{M} \times \mathrm{T}^\star\mathsf{M}$, we have

$$\mathrm{Q}_{n,1}(z, \mathrm{d}z') = \mathbb{1}_{\bar{\mathsf{E}}_h^\Phi}(z)(s_\# \mathrm{Q}_{n,2})(z, \mathrm{d}z') + \mathbb{1}_{(\bar{\mathsf{E}}_h^\Phi)^c}(z)\delta_{s(z)}(\mathrm{d}z') \, , \tag{A.51}$$

where

$$\mathrm{Q}_{n,2}(z, \mathrm{d}z') = \bar{a}(\mathrm{R}_h^\Phi(z) \mid z)\delta_{\mathrm{R}_h^\Phi(z)}(\mathrm{d}z') + [1 - \bar{a}(\mathrm{R}_h^\Phi(z) \mid z)]\delta_z(\mathrm{d}z') \, . \tag{A.52}$$

Kernel Q_n. This kernel corresponds to one step of Algorithm 1 (*i.e.*, comprising Steps 1 to 5). For any $(z, z') \in \mathrm{T}^\star\mathsf{M} \times \mathrm{T}^\star\mathsf{M}$, we have

$$\mathrm{Q}_n(z, \mathrm{d}z') = \int_{\mathrm{T}^\star\mathsf{M}\times\mathrm{T}^\star\mathsf{M}} \mathrm{Q}_0(z, \mathrm{d}z_1)\mathrm{Q}_{n,1}(z_1, \mathrm{d}z_2)\mathrm{Q}_0(z_2, \mathrm{d}z') \, .$$

A.9.3 Proof of reversibility in Algorithm 3

Let $h > 0$. Using notation from **A**2, we recall definition of the set A_h introduced in Section 2.4.2

$$\mathsf{A}_h = \{z \in \mathrm{T}^\star\mathsf{M} \, : \, h < \min_{\tilde{z}\in\mathsf{B}_{\|\cdot\|_z}(z,1)} h_\star(\tilde{z})\} \subset \mathrm{dom}_{\Phi_h} \, . \tag{A.53}$$

We also recall that $\bar{\pi}$, as defined in (2.5), admits a density with respect to the product Lebesgue measure given for any $z = (x, p) \in \mathrm{T}^\star\mathsf{M}$ by

$$(\mathrm{d}\bar{\pi}/(\mathrm{d}x\mathrm{d}p))(x, p) = (1/Z)\exp[-(1/2)\|p\|^2_{\mathfrak{g}(x)^{-1}}]\det(\mathfrak{g}(x))^{-1/2}\exp[-V(x)] \, .$$

Since Algorithm 3 can be thought as a *restrictive* version of Algorithm 1, the Markov kernels from Tn-BHMC are similar to the kernels from n-BHMC, defined in Appendix A.9.2. In particular, the transition kernel corresponding to the Gaussian momentum update (Step 1 and 5 in Algorithm 3, Step 1 and 5 in Algorithm 1) is the same and is reversible (up to momentum reversal) with respect to $\bar{\pi}$ (see Lemma 38). Namely, we replace

(a) the set $\bar{\mathsf{E}}_h^\Phi$, defined in (A.50), by $\tilde{\mathsf{E}}_h^\Phi$,

(b) the kernels $\mathrm{Q}_{n,1}$ and $\mathrm{Q}_{n,2}$, respectively defined in (A.51) and (A.52), by Q_1 and Q_2,

where

$$\tilde{\mathsf{E}}_h^{\Phi} = (s \circ \mathrm{S}_{h/2})^{-1}(\mathsf{A}_h \cap \Phi_h^{-1}(\mathsf{A}_h)) = (s \circ \mathrm{S}_{h/2})(\mathsf{A}_h \cap \Phi_h^{-1}(\mathsf{A}_h)) \;, \tag{A.54}$$

$$\mathrm{Q}_1(z, \mathrm{d}z') = \mathbb{1}_{\tilde{\mathsf{E}}_h^{\Phi}}(z)(s_{\#}\mathrm{Q}_2)(z, \mathrm{d}z') + \mathbb{1}_{(\tilde{\mathsf{E}}_h^{\Phi})^{\mathrm{c}}}(z)\delta_{s(z)}(\mathrm{d}z') \;, \tag{A.55}$$

$$\mathrm{Q}_2(z, \mathrm{d}z') = a(\mathrm{R}_h^{\Phi}(z) \mid z)\delta_{\mathrm{R}_h^{\Phi}}(\mathrm{d}z') + [1 - a(\mathrm{R}_h^{\Phi}(z) \mid z)]\delta_z(\mathrm{d}z') \;. \tag{A.56}$$

We denote by $\mathrm{Q} : \mathrm{T}^\star\mathsf{M} \times \mathcal{B}(\mathrm{T}^\star\mathsf{M}) \to [0,1]$, the transition kernel of the (homogeneous) Markov chain $(x_n, p_n)_{n\in[N]}$ generated by Algorithm 3. For any $(z, z') \in \mathrm{T}^\star\mathsf{M} \times \mathrm{T}^\star\mathsf{M}$, we have

$$\mathrm{Q}(z, \mathrm{d}z') = \textstyle\int_{\mathrm{T}^\star\mathsf{M}\times\mathrm{T}^\star\mathsf{M}} \mathrm{Q}_0(z, \mathrm{d}z_1)\mathrm{Q}_1(z_1, \mathrm{d}z_2)\mathrm{Q}_0(z_2, \mathrm{d}z') \;. \tag{A.57}$$

We first turn to the reversibility up to momentum reversal of Q_1 with respect to $\bar{\pi}$. We start with the following lemma which is key to establish this result.

Lemma 50. *Assume* **A**0, **A**1, **A**2. *Then, for any compact set* $\mathsf{G} \subset \mathrm{T}^\star\mathsf{M}$, *for any* $g \in \mathrm{C}(\mathrm{T}^\star\mathsf{M} \times \mathrm{T}^\star\mathsf{M}, \mathbb{R})$ *such that* $\mathrm{supp}(g) \subset \mathsf{G} \times \mathsf{G}$, *we have*

$$\textstyle\int_{\mathsf{G}_h} g(z, \Phi_h(z))\mathrm{d}z = \int_{\mathsf{G}_h} g(\Phi_h(z), z)\mathrm{d}z \;, \tag{A.58}$$

where $\mathsf{G}_h = \mathsf{F}_h \cap \mathsf{G} \cap \Phi_h^{-1}(\mathsf{G})$, $\mathsf{F}_h = \mathsf{A}_h \cap \Phi_h^{-1}(\mathsf{A}_h)$, A_h *being defined in* (A.53).

Proof. Assume **A**0, **A**1, **A**2. Let $\mathsf{G} \subset \mathrm{T}^\star\mathsf{M}$ be a compact set, $g \in \mathrm{C}(\mathrm{T}^\star\mathsf{M} \times \mathrm{T}^\star\mathsf{M}, \mathbb{R})$ such that $\mathrm{supp}(g) \subset \mathsf{G} \times \mathsf{G}$. We first define the sets $\mathsf{F}_h = \mathsf{A}_h \cap \Phi_h^{-1}(\mathsf{A}_h)$, $\mathsf{G}_h = \mathsf{F}_h \cap \mathsf{G} \cap \Phi_h^{-1}(\mathsf{G})$ and the integrals I' and J'

$$I' = \textstyle\int_{\mathsf{G}_h} g(z, \Phi_h(z))\mathrm{d}z \;\;, \quad J' = \int_{\mathsf{G}_h} g(\Phi_h(z), z)\mathrm{d}z \;. \tag{A.59}$$

We recall the existence under **A**0 and **A**1 of $\mathtt{L} > 0$ and $\mathtt{M} > 0$ such that $r^\star$, given by **A**2 and defined in (A.49), is $\mathtt{L}$-Lipschitz-continuous on M with respect to $\|\cdot\|_2$ and bounded from above by $\mathtt{M}$ (see Lemma 48).

We define $r : \mathrm{T}^\star\mathsf{M} \to (0, +\infty)$ by $r(x,p) = r^\star(x)/\mathtt{m}$ for any $(x,p) \in \mathrm{T}^\star\mathsf{M}$, where $\mathtt{m} = \max\{1/\lambda, 4\mathtt{M}, 4\mathtt{L}\}$ and λ is given by **A**2. Using the properties of $r^\star$, it comes that

(a) r is $\mathtt{L}$-Lipschitz on $\mathrm{T}^\star\mathsf{M}$ with respect to $\|\cdot\|_2$,

(b) $r(x,p) \leq 1/(4\mathtt{L}C_x)$ for any $(x,p) \in \mathrm{T}^\star\mathsf{M}$, where C_x is defined in Lemma 33,

(c) $r \leq 1/4$, and

(d) $r \leq \lambda r^\star$.

Note that $\mathsf{G} \subset \cup_{z\in\mathsf{G}} \mathsf{B}_{\|\cdot\|_z}(z, r(z))$. Since G is a compact set, there exist $K \in \mathbb{N}$ and $(z_i)_{i\in[K]} \in \mathrm{T}^\star\mathsf{M}^K$ such that $\mathsf{G} \subset \bigcup_{i=1}^K \mathsf{B}_i$, where $\mathsf{B}_i = \mathsf{B}_{\|\cdot\|_{z_i}}(z_i, r(z_i))$.

We consider the sequence $\{\mathsf{V}_i\}_{i=1}^K$ constructed as follows: $\mathsf{V}_1 = \mathsf{G} \cap \mathsf{B}_1$ and for any $i \in \{2, \ldots, K\}$, $\mathsf{V}_i = (\mathsf{G} \cap \mathsf{B}_i) \cap (\cup_{j=1}^{i-1}\mathsf{V}_j)^{\mathrm{c}}$. Then, we have that, for any $i \in \{1, \ldots, N\}$, $\cup_{j=1}^i \mathsf{V}_j = \mathsf{G} \cap (\cup_{j=1}^i \mathsf{B}_j)$, and that for any $i_1, i_2 \in \{1, \ldots, K\}$, $\mathsf{V}_{i_1} \cap \mathsf{V}_{i_2} = \emptyset$ if $i_1 \neq i_2$. Therefore, we get that $\mathsf{G} = \sqcup_{i=1}^K \mathsf{V}_i$ and $\Phi_h^{-1}(\mathsf{G}) = \sqcup_{i=1}^K \Phi_h^{-1}(\mathsf{V}_i)$. In particular, for any $\mathsf{A} \in \mathcal{B}(\mathrm{T}^\star\mathsf{M})$ and any $\zeta \in \mathrm{C}_c(\mathrm{T}^\star\mathsf{M}, \mathbb{R})$

$$\textstyle\int_{\mathsf{G}\cap\mathsf{A}} \zeta(z)\mathrm{d}z = \sum_{i=1}^K \int_{\mathsf{V}_i\cap\mathsf{A}} \zeta(z)\mathrm{d}z \;, \tag{A.60}$$

and for any $\tilde{\mathsf{A}} \in \mathcal{B}(\mathrm{T}^\star\mathsf{M})$ and any $\tilde{\zeta} \in \mathrm{C}_c(\mathrm{T}^\star\mathsf{M}, \mathbb{R})$

$$\textstyle\int_{\Phi_h^{-1}(\mathsf{G})\cap\tilde{\mathsf{A}}} \tilde{\zeta}(z)\mathrm{d}z = \sum_{i=1}^K \int_{\Phi_h^{-1}(\mathsf{V}_i)\cap\tilde{\mathsf{A}}} \tilde{\zeta}(z)\mathrm{d}z \;. \tag{A.61}$$

Using (A.60) and (A.59), we obtain $I' = \sum_{i=1}^K I'_i$, where $I'_i = \int_{\mathsf{V}_i\cap\Phi_h^{-1}(\mathsf{G})\cap\mathsf{F}_h} g(z, \Phi_h(z))\mathrm{d}z$ for any $i \in [K]$. We are now going to show that, for any $i \in [K]$

$$I'_i = \int_{\Phi_h^{-1}(\mathsf{V}_i)\cap\mathsf{G}\cap\mathsf{F}_h} g(\Phi_h(z), z)\mathrm{d}z \;. \tag{A.62}$$

Let $i \in [K]$. We proceed by making the following case disjunction:

(a) Either $\mathsf{V}_i \cap \Phi_h^{-1}(\mathsf{G}) \cap \mathsf{F}_h = \emptyset$, and then $I_i' = 0$. We prove by contradiction in this case that $\Phi_h^{-1}(\mathsf{V}_i) \cap \mathsf{G} \cap \mathsf{F}_h = \emptyset$. Assume that there exists $z \in \Phi_h^{-1}(\mathsf{V}_i) \cap \mathsf{G} \cap \mathsf{F}_h$. By definition of F_h, $z \in \mathsf{A}_h$, $\Phi_h(z) \in \mathsf{A}_h$, and we get that $\Phi_h(\Phi_h(z)) = z$ using **A**2. Hence, we have:

- $\Phi_h(z) \in \mathsf{V}_i$,
- $\Phi_h(\Phi_h(z)) = z \in \mathsf{G}$,
- $\Phi_h(z) \in \mathsf{A}_h$,
- $\Phi_h(\Phi_h(z)) = z \in \mathsf{A}_h$.

Therefore, we get $\Phi_h(z) \in \mathsf{V}_i \cap \Phi_h^{-1}(\mathsf{G}) \cap \mathsf{F}_h = \emptyset$, which is absurd. Finally, (A.62) holds since

$$I_i' = 0 = \textstyle\int_{\Phi_h^{-1}(\mathsf{V}_i) \cap \mathsf{G} \cap \mathsf{F}_h} g(\Phi_h(z), z) \mathrm{d}z \ .$$

(b) Either, there exists some $\tilde{z} \in \mathsf{V}_i \cap \Phi_h^{-1}(\mathsf{G})$ such that $\tilde{z} \in \mathsf{F}_h$. In particular, $\tilde{z} \in \mathsf{B}_i$, and thus $\|\tilde{z} - z_i\|_{z_i} < r(z_i) < 1$. In this case, by combining **A**2 with Lemma 30-(c), we have for any $z' \in \mathrm{T}^\star\mathsf{M}$

$$\|z'\|_{\tilde{z}} \leq (1 - \|\tilde{z} - z_i\|_{z_i})^{-1} \|z'\|_{z_i} < (1 - r(z_i))^{-1} \|z'\|_{z_i} \ .$$

By considering $z' = z_i - \tilde{z}$, it comes that

$$\|z_i - \tilde{z}\|_{\tilde{z}} < (1 - r(z_i))^{-1} \|\tilde{z} - z_i\|_{z_i} < (1 - r(z_i))^{-1} r(z_i) \ .$$

By combining properties (a) and (b) of r with Lemma 33, we have the following upper bound of $r(z_i)$

$$r(z_i) \leq r(\tilde{z}) + \mathrm{L} \|\tilde{z} - z_i\|_2 \leq r(\tilde{z}) + \mathrm{L} C_{x_i} \|\tilde{z} - z_i\|_{z_i} < r(\tilde{z}) + \mathrm{L} C_{x_i} r(z_i) < r(\tilde{z}) + 1/4 \ ,$$

and thus

$$\|z_i - \tilde{z}\|_{\tilde{z}} < (1 - r(\tilde{z}) - 1/4)^{-1} (r(\tilde{z}) + 1/4) < 1 \ ,$$

using property (c) of r. Hence, $z_i \in \mathsf{B}_{\|\cdot\|_{\tilde{z}}}(\tilde{z}, 1)$. Moreover, $\tilde{z} \in \mathsf{F}_h \subset \mathsf{A}_h$, and then it comes that $h < h_\star(z_i)$. Therefore, by combining property (d) of r with **A**2, we have

(i) $\mathsf{V}_i \subset \mathsf{B}_i \subset \mathsf{B}_{\|\cdot\|_{z_i}}(z_i, \lambda r^\star(z_i)) \subset \mathrm{dom}_{\Phi_h}$,

(ii) the restriction of Φ_h to V_i is a C^1-diffeomorphism such that $\Phi_h \circ \Phi_h = \mathrm{Id}$.

This last result provides proper assumptions on Φ_h to operate a change of variable in I_i'. We now define $\mathsf{G}_{1,h} = \Phi_h(\mathsf{V}_i \cap \Phi_h^{-1}(\mathsf{G}) \cap \mathsf{F}_h)$ and $\mathsf{G}_{2,h} = \Phi_h^{-1}(\mathsf{V}_i) \cap \mathsf{G} \cap \mathsf{F}_h$, and prove that $\mathsf{G}_{1,h} = \mathsf{G}_{2,h}$ in two steps.

(i) We first prove that $\mathsf{G}_{1,h} \subset \mathsf{G}_{2,h}$. Let $z \in \mathsf{G}_{1,h}$. Then, there exists $z' \in \mathsf{V}_i \cap \Phi_h^{-1}(\mathsf{G}) \cap \mathsf{F}_h$ such that $z = \Phi_h(z')$. Since Φ_h is an involution on V_i, we have $\Phi_h(z) = z'$. Since $z' \in \mathsf{V}_i \cap \mathsf{A}_h$, it comes that $z \in \Phi_h^{-1}(\mathsf{V}_i) \cap \Phi_h^{-1}(\mathsf{A}_h)$. Moreover, $z' \in \Phi_h^{-1}(\mathsf{G}) \cap \Phi_h^{-1}(\mathsf{A}_h)$, and thus, $z \in \mathsf{G} \cap \mathsf{A}_h$. Then, we have $z \in \mathsf{G}_{2,h}$, which proves this first result.

(ii) We now prove that $\mathsf{G}_{2,h} \subset \mathsf{G}_{1,h}$. Let $z \in \mathsf{G}_{2,h}$. In particular, $z \in \mathsf{G}$. Then, there exists $j \in [K]$ such that $z \in \mathsf{V}_j$. Therefore, $z \in \mathsf{B}_j$ and thus $\|z - z_j\|_{z_j} < r(z_j) < 1$. Since $z \in \mathsf{A}_h$, we obtain that $h < h_\star(z_j)$ with the same computations as those written above. In particular, this proves with **A**2 that Φ_h is an involution on V_j. Then, $z = \Phi_h(\Phi_h(z))$, with $\Phi_h(z) \in \mathsf{V}_i \cap \mathsf{A}_h$ and $\Phi_h(z) \in \Phi_h^{-1}(\mathsf{G}) \cap \Phi_h^{-1}(\mathsf{A}_h)$, since $z \in \mathsf{G} \cap \mathsf{A}_h$. Therefore, we have $z \in \mathsf{G}_{1,h}$, which proves this last result.

Given the fact that Φ_h is an involution on V_i, we operate the change of variable $z \mapsto \Phi_h(z)$ in I_i' and obtain

$$I_i' = \textstyle\int_{\Phi_h(\mathsf{V}_i \cap \Phi_h^{-1}(\mathsf{G}) \cap \mathsf{F}_h)} g(\Phi_h(z), z) \mathrm{d}z = \int_{\Phi_h^{-1}(\mathsf{V}_i) \cap \mathsf{G} \cap \mathsf{F}_h} g(\Phi_h(z), z) \mathrm{d}z \ ,$$

which gives (A.62).

Therefore, combining (A.60), (A.61) and (A.62), we get

$$I' = \textstyle\sum_{i=1}^K \int_{\mathsf{V}_i \cap \Phi_h^{-1}(\mathsf{G}) \cap \mathsf{F}_h} g(z, \Phi_h(z)) \mathrm{d}z = \sum_{i=1}^K \int_{\Phi_h^{-1}(\mathsf{V}_i) \cap \mathsf{G} \cap \mathsf{F}_h} g(\Phi_h(z), z) \mathrm{d}z = J' \ ,$$

which concludes the proof. □

We are now ready to establish the reversibility up to momentum reversal of Q_1, defined in (A.55), with respect to $\bar{\pi}$.

Lemma 51. *Assume* **A**0, **A**1, **A**2. *Then, for any $h > 0$, the Markov kernel Q_1 with step-size h, defined in (A.55), is reversible up to momentum reversal with respect to $\bar{\pi}$.*

Proof. Assume **A**0, **A**1, **A**2. Let $h > 0$. We recall that the kernels Q_1 and Q_2 are respectively defined in (A.55) and (A.56). We define the transition kernel $\mathrm{Q}_3 : \mathrm{T}^\star\mathsf{M} \times \mathcal{B}(\mathrm{T}^\star\mathsf{M}) \to [0,1]$ by

$$\mathrm{Q}_3(z, \mathrm{d}z') = \mathbb{1}_{\tilde{\mathsf{E}}_h^\Phi}(z)\mathrm{Q}_2(z, \mathrm{d}z') + \mathbb{1}_{(\tilde{\mathsf{E}}_h^\Phi)^{\mathrm{c}}}(z)\delta_z(\mathrm{d}z') \,, \tag{A.63}$$

such that $\mathrm{Q}_1 = s_\# \mathrm{Q}_3$. The rest of the proof is divided into two parts. First, we prove that Q_3 is reversible with respect to $\bar{\pi}$. Then, we prove that Q_1 is reversible up to momentum reversal with respect to $\bar{\pi}$.

(a) Let $f \in \mathrm{C}(\mathrm{T}^\star\mathsf{M} \times \mathrm{T}^\star\mathsf{M}, \mathbb{R})$ with compact support. We consider a compact set K with respect to the topology induced by the set $\{\mathsf{B}_{\|\cdot\|_z}(z, r), z \in \mathrm{T}^\star\mathsf{M}, r \in (0,1)\}$ such that $\mathrm{supp}(f) \subset \mathsf{K} \times \mathsf{K}$. According to Definition 25, we aim to show that

$$\int_{\mathrm{T}^\star\mathsf{M}\times\mathrm{T}^\star\mathsf{M}} f(z, z')\mathrm{Q}_3(z, \mathrm{d}z')\bar{\pi}(\mathrm{d}z) = \int_{\mathrm{T}^\star\mathsf{M}\times\mathrm{T}^\star\mathsf{M}} f(z', z)\mathrm{Q}_3(z, \mathrm{d}z')\bar{\pi}(\mathrm{d}z) \,. \tag{A.64}$$

We denote by I the left integral of (A.64). By combining (A.56) and (A.63), we have $I = I_1 + I_2 + I_3$ where

$$\begin{aligned}
I_1 &= \int_{\tilde{\mathsf{E}}_h^\Phi} \bar{\pi}(z) a(\mathrm{R}_h^\Phi(z) \mid z) f(z, \mathrm{R}_h^\Phi(z))\mathrm{d}z \,, \\
I_2 &= \int_{\tilde{\mathsf{E}}_h^\Phi} \bar{\pi}(z)[1 - a(\mathrm{R}_h^\Phi(z) \mid z)] f(z, z)\mathrm{d}z \,, \\
I_3 &= \int_{(\tilde{\mathsf{E}}_h^\Phi)^{\mathrm{c}}} \bar{\pi}(z) f(z, z)\mathrm{d}z \,.
\end{aligned}$$

Since $\mathrm{supp}(f) \subset \mathsf{K} \times \mathsf{K}$, we have for any $z \in \mathsf{K}^{\mathrm{c}}$, $f(z, \cdot) = 0$ and $f(\cdot, z) = 0$. Note also that for any $z \in ((\mathrm{R}_h^\Phi)^{-1}(\mathsf{K}))^{\mathrm{c}}$, $\mathrm{R}_h^\Phi(z) \notin \mathsf{K}$, and thus $f(\mathrm{R}_h^\Phi(z), \cdot) = 0$ and $f(\cdot, \mathrm{R}_h^\Phi(z)) = 0$. By combining these preliminary results with (2.19), we get

$$\begin{aligned}
I_1 &= \int_{\tilde{\mathsf{E}}_h^\Phi \cap \mathsf{K} \cap (\mathrm{R}_h^\Phi)^{-1}(\mathsf{K})} \bar{\pi}(z) a(\mathrm{R}_h^\Phi(z) \mid z) f(z, \mathrm{R}_h^\Phi(z))\mathrm{d}z \\
&= (1/Z) \int_{\tilde{\mathsf{E}}_h^\Phi \cap \mathsf{K} \cap (\mathrm{R}_h^\Phi)^{-1}(\mathsf{K})} \min\{e^{-H(z)}, e^{-H(\mathrm{R}_h^\Phi(z))}\} f(z, \mathrm{R}_h^\Phi(z))\mathrm{d}z \\
I_2 &= \int_{\tilde{\mathsf{E}}_h^\Phi \cap \mathsf{K}} \bar{\pi}(z)[1 - a(\mathrm{R}_h^\Phi(z) \mid z)] f(z, z)\mathrm{d}z \,, \\
I_3 &= \int_{(\tilde{\mathsf{E}}_h^\Phi)^{\mathrm{c}}} \bar{\pi}(z) f(z, z)\mathrm{d}z \,.
\end{aligned}$$

We denote by J the right integral of (A.64). By symmetry, we have $J = J_1 + J_2 + J_3$ where

$$\begin{aligned}
J_1 &= \int_{\tilde{\mathsf{E}}_h^\Phi \cap \mathsf{K} \cap (\mathrm{R}_h^\Phi)^{-1}(\mathsf{K})} \bar{\pi}(z) a(\mathrm{R}_h^\Phi(z) \mid z) f(\mathrm{R}_h^\Phi(z), z)\mathrm{d}z \\
&= (1/Z) \int_{\tilde{\mathsf{E}}_h^\Phi \cap \mathsf{K} \cap (\mathrm{R}_h^\Phi)^{-1}(\mathsf{K})} \min\{e^{-H(z)}, e^{-H(\mathrm{R}_h^\Phi(z))}\} f(\mathrm{R}_h^\Phi(z), z)\mathrm{d}z \,, \\
J_2 &= \int_{\tilde{\mathsf{E}}_h^\Phi \cap \mathsf{K}} \bar{\pi}(z)[1 - a(\mathrm{R}_h^\Phi(z) \mid z)] f(z, z)\mathrm{d}z \,, \\
J_3 &= \int_{(\tilde{\mathsf{E}}_h^\Phi)^{\mathrm{c}}} \bar{\pi}(z) f(z, z)\mathrm{d}z \,.
\end{aligned}$$

We directly have $I_2 = J_2$ and $I_3 = J_3$. Let us now prove that $I_1 = J_1$.

We recall that $s \circ \mathrm{S}_{h/2}$ is a symplectic C^1-diffeomorphism (see Section 2.3.1) and $\tilde{\mathsf{E}}_h^\Phi = (s \circ \mathrm{S}_{h/2})(\mathsf{F}_h)$ where $\mathsf{F}_h = \mathsf{A}_h \cap \Phi_h^{-1}(\mathsf{A}_h)$, using (A.54). We define $\mathsf{K}_h = \mathsf{F}_h \cap (s \circ \mathrm{S}_{h/2})(\mathsf{K}) \cap \Phi_h^{-1}((s \circ \mathrm{S}_{h/2})(\mathsf{K}))$, such that $\tilde{\mathsf{E}}_h^\Phi \cap \mathsf{K} \cap (\mathrm{R}_h^\Phi)^{-1}(\mathsf{K}) = (s \circ \mathrm{S}_{h/2})(\mathsf{K}_h)$, and we operate the change of variable $z \mapsto (s \circ \mathrm{S}_{h/2})(z)$ in I_1 and J_1

$$\begin{aligned}
I_1 &= (1/Z) \int_{\mathsf{K}_h} \min\{e^{-H((s\circ\mathrm{S}_{h/2})(z))}, e^{-H((s\circ\mathrm{S}_{h/2}\circ\Phi_h)(z))}\} f((s \circ \mathrm{S}_{h/2})(z), (s \circ \mathrm{S}_{h/2} \circ \Phi_h)(z))\mathrm{d}z \,, \\
J_1 &= (1/Z) \int_{\mathsf{K}_h} \min\{e^{-H((s\circ\mathrm{S}_{h/2})(z))}, e^{-H((s\circ\mathrm{S}_{h/2}\circ\Phi_h)(z))}\} f((s \circ \mathrm{S}_{h/2} \circ \Phi_h)(z), (s \circ \mathrm{S}_{h/2})(z))\mathrm{d}z \,.
\end{aligned}$$

We now define the map $g : \mathrm{T}^\star\mathsf{M} \times \mathrm{T}^\star\mathsf{M} \to \mathbb{R}$ and the set $\mathsf{G} \subset \mathrm{T}^\star\mathsf{M}$ by

$$\begin{aligned}
g(z, z') &= \min\{e^{-H((s\circ\mathrm{S}_{h/2})(z))}, e^{-H((s\circ\mathrm{S}_{h/2})(z'))}\} f((s \circ \mathrm{S}_{h/2})(z), (s \circ \mathrm{S}_{h/2})(z')) \,, \\
\mathsf{G} &= (s \circ \mathrm{S}_{h/2})(\mathsf{K}) \,.
\end{aligned}$$

Note that G is a compact set of $\mathrm{T}^\star\mathsf{M}$ by continuity of $s \circ \mathrm{S}_{h/2}$ and g is a continuous function by continuity of H and $s \circ \mathrm{S}_{h/2}$. Then, we obtain $I_1 = J_1$, by applying Lemma 50 with g and G. Finally, we obtain $I = J$ and thus prove (A.64) for any continuous function f with compact support.

(b) Let $f : \mathrm{T}^\star\mathsf{M} \times \mathrm{T}^\star\mathsf{M} \to \mathbb{R}$ be a continuous function with compact support. We have

$$\begin{aligned}
&\int_{\mathrm{T}^\star\mathsf{M}\times\mathrm{T}^\star\mathsf{M}} f(z,z')\mathrm{Q}_1(z,\mathrm{d}z')\bar{\pi}(\mathrm{d}z) \\
&\quad = \int_{\mathrm{T}^\star\mathsf{M}\times\mathrm{T}^\star\mathsf{M}} f(z,z')(s_\# \mathrm{Q}_3)(z,\mathrm{d}z')\bar{\pi}(\mathrm{d}z) \\
&\quad = \int_{\mathrm{T}^\star\mathsf{M}\times\mathrm{T}^\star\mathsf{M}} f(z,s(z'))\mathrm{Q}_3(z,\mathrm{d}z')\bar{\pi}(\mathrm{d}z) && \text{(momentum reversal on } z') \\
&\quad = \int_{\mathrm{T}^\star\mathsf{M}\times\mathrm{T}^\star\mathsf{M}} f(z',s(z))\mathrm{Q}_3(z,\mathrm{d}z')\bar{\pi}(\mathrm{d}z) && \text{(using (A.64))} \\
&\quad = \int_{\mathrm{T}^\star\mathsf{M}\times\mathrm{T}^\star\mathsf{M}} f(s(z'),s(z))(s_\# \mathrm{Q}_3)(z,\mathrm{d}z')\bar{\pi}(\mathrm{d}z) && \text{(momentum reversal on } z') \\
&\quad = \int_{\mathrm{T}^\star\mathsf{M}\times\mathrm{T}^\star\mathsf{M}} f(s(z'),s(z))\mathrm{Q}_1(z,\mathrm{d}z')\bar{\pi}(\mathrm{d}z)\ ,
\end{aligned}$$

which concludes the proof.

□

We are now ready to prove Theorem 3, which states that Q is reversible up to momentum reversal with respect to $\bar{\pi}$.

Proof of Theorem 3. Assume **A**0, **A**1, **A**2. This proof is very similar to the proof of Theorem 36 (see Appendix A.5.2). Let $f : \mathrm{T}^\star\mathsf{M} \times \mathrm{T}^\star\mathsf{M} \to \mathbb{R}$ be a continuous function with compact support. We have

$$\begin{aligned}
&\int_{\mathrm{T}^\star\mathsf{M}\times\mathrm{T}^\star\mathsf{M}} f(z,z')\mathrm{Q}(z,\mathrm{d}z')\bar{\pi}(\mathrm{d}z) \\
&\quad = \int_{(\mathrm{T}^\star\mathsf{M})^4} f(z,z')\mathrm{Q}_0(z,\mathrm{d}z_1)\mathrm{Q}_1(z_1,\mathrm{d}z_2)\mathrm{Q}_0(z_2,\mathrm{d}z')\bar{\pi}(\mathrm{d}z) && \text{(see (A.57))} \\
&\quad = \int_{(\mathrm{T}^\star\mathsf{M})^4} f(s(z_1),z')\mathrm{Q}_0(z,\mathrm{d}z_1)\mathrm{Q}_1(s(z),\mathrm{d}z_2)\mathrm{Q}_0(z_2,\mathrm{d}z')\bar{\pi}(\mathrm{d}z) && \text{(Lemma 38)} \\
&\quad = \int_{(\mathrm{T}^\star\mathsf{M})^4} f(s(z_1),z')\mathrm{Q}_0(s(z),\mathrm{d}z_1)\mathrm{Q}_1(z,\mathrm{d}z_2)\mathrm{Q}_0(z_2,\mathrm{d}z')\bar{\pi}(\mathrm{d}z) && \text{(momentum reversal on } z) \\
&\quad = \int_{(\mathrm{T}^\star\mathsf{M})^4} f(s(z_1),z')\mathrm{Q}_0(z_2,\mathrm{d}z_1)\mathrm{Q}_1(z,\mathrm{d}z_2)\mathrm{Q}_0(s(z),\mathrm{d}z')\bar{\pi}(\mathrm{d}z) && \text{(Lemma 51)} \\
&\quad = \int_{(\mathrm{T}^\star\mathsf{M})^4} f(s(z_1),z')\mathrm{Q}_0(z_2,\mathrm{d}z_1)\mathrm{Q}_1(s(z),\mathrm{d}z_2)\mathrm{Q}_0(z,\mathrm{d}z')\bar{\pi}(\mathrm{d}z) && \text{(momentum reversal on } z) \\
&\quad = \int_{(\mathrm{T}^\star\mathsf{M})^4} f(s(z_1),s(z))\mathrm{Q}_0(z_2,\mathrm{d}z_1)\mathrm{Q}_1(z',\mathrm{d}z_2)\mathrm{Q}_0(z,\mathrm{d}z')\bar{\pi}(\mathrm{d}z) && \text{(Lemma 38)} \\
&\quad = \int_{\mathrm{T}^\star\mathsf{M}\times\mathrm{T}^\star\mathsf{M}} f(s(z'),s(z))\mathrm{Q}(z,\mathrm{d}z')\bar{\pi}(\mathrm{d}z)\ .
\end{aligned}$$

Moreover, $s_\#\bar{\pi} = \bar{\pi}$. Hence, by combining Definition 25 and Lemma 26, we obtain the result of Theorem 3. □

A.10 Comparison with Kook et al. [2022].

In this section, we provide a precise comparison between our work and the results stated in Kook et al. [2022]. We first dwell on the differences related to the general setting and the algorithms, and then explain how our methodology may solve the limitations of Kook et al. [2022].

A.10.1 General framework

Given a convex body $\mathsf{K} \subset \mathbb{R}^d$ and a function $c : \mathbb{R}^d \to \mathbb{R}^m$, consider

$$\mathsf{M} = \{x \in \mathsf{K} : c(x) = 0\}\ . \tag{A.65}$$

In their paper, Kook et al. [2022] aim at sampling from a target distribution π, with density given for $x \in \mathsf{M}$ by

$$\mathrm{d}\pi(x)/\mathrm{d}x = \exp[-V(x)]/Z\ , \tag{A.66}$$

where $V \in \mathrm{C}^2(\mathsf{M},\mathbb{R})$, $Z = \int_{\mathsf{M}} \exp[-V(x)]\mathrm{d}x$. They make the following assumptions:

(a) K is provided with a self-concordant barrier ϕ.

(b) The differential of c at x, $Dc(x)$, is full-ranked for any $x \in \mathsf{M}$.

Although this setting might be more general than ours (consider for instance the case where K is a polytope and c is a non-trivial function), Kook et al. [2022] focus on the case

$$\mathsf{K} = \{x \in \mathbb{R}^d : \ell \le x \le u\}, \quad c(x) = Ax - b \tag{A.67}$$

where $\ell, u \in \mathbb{R}^d$ with $\ell < u$, $b \in \mathbb{R}^m$, and $A \in \mathbb{R}^{d \times m}$. Combined with assumptions (a) and (b), A must have independent rows and the Hessian of ϕ at x given by $\mathfrak{g}(x) = D^2\phi(x)$ is a diagonal positive matrix for any x. Moreover, M defined by (A.65)-(A.67) is a polytope. However, not any polytope can be rewritten in a manner akin to (A.67), and therefore, the setting chosen by Kook et al. [2022] can actually be considered as *a special case of our setting*, see Section 2.2.

Then, they consider the extended state-space $\bar{\mathsf{M}} = \{(x,p) \in \mathbb{R}^d \times \mathbb{R}^d : x \in \mathsf{M}, p \in \mathrm{Ker}(Dc(x))\}$ and define on $\bar{\mathsf{M}}$ a *constrained* Hamiltonian H, given by

$$H(x,p) = \bar{H}(x,p) + \lambda(x,p)^\top c(x) \,, \tag{A.68}$$

$$\text{with } \bar{H}(x,p) = V(x) + \frac{1}{2}\log \mathrm{pdet}\, M(x) + \frac{1}{2} p^\top M(x)^\dagger p \,, \tag{A.69}$$

$$\text{and } M(x) = Q(x)^\top \mathfrak{g}(x) Q(x) \,, \tag{A.70}$$

where Q is the orthogonal projection into $\mathrm{Ker}(Dc(x))$, implying that M is semi-definite positive, and $\lambda(x,p)$ is a Lagrangian term given by

$$\lambda(x,p) = (Dc(x)Dc(x)^\top)^{-1}\{D^2c(x)[p, \mathrm{d}x/\mathrm{d}t] - Dc(x)\partial \bar{H}(x,p)/\partial x\} \,. \tag{A.71}$$

Remark that λ simplifies when c is chosen as in (A.67), but may not be well suited for higher order constraints. Then, Kook et al. [2022] provide simplifications of H for the setting (A.67) based on formulas for $\mathrm{pdet}\, M$ and $M^\dagger$.

Finally, they consider the joint distribution $\bar{\pi}$ given for any $(x,p) \in \bar{\mathsf{M}}$ by

$$\mathrm{d}\bar{\pi} = (1/\bar{Z}) \exp[-H(x,p)] \mathrm{d}x\mathrm{d}p, \tag{A.72}$$

where $\bar{Z} = \int_{\bar{\mathsf{M}}} \exp[-H(x,p)]\mathrm{d}x\mathrm{d}p$, for which the first marginal is π. They naturally propose to sample from $\bar{\pi}$ by implementing a version of RMHMC relying on the *constrained* Hamiltonian H, which they call CRHMC.

Similarly to our approach, they consider two cases.

(a) First, Kook et al. [2022] assume that the Hamiltonian dynamics given by H can be explicitly computed in continuous time (which is not the case in practice). They present the continuous version of RMHMC which incorporates the simplified *constrained* Hamiltonian H, see Algorithm 1. We refer to this algorithm as c-CRHMC. This algorithm is identical to our algorithm c-BHMC provided in Appendix A.4, apart from the involution checking step (see Line 4 in Algorithm 2). They state in Theorem 6 that the Markov kernel corresponding to one iteration of c-CRHMC satisfies detailed balance with respect to $\bar{\pi}$, thus ensuring that it preserves $\bar{\pi}$.

(b) Then, Kook et al. [2022] provide a practical implementation of CRHMC, which is similar to n-BHMC, based on a time-discretization of the Hamiltonian dynamics for a given time-step $h > 0$. They first split the simplified *constrained* Hamiltonian H into H_1 and H_2, by leveraging the non-separable aspect of H in H_2, as we do in Section 2.3.1. Then, they propose to use the first-order Euler method to solve the discretized ODE associated to H_1, as we also suggest, and the *Implicit Midpoint Integrator* (IMI) to solve the discretized ODE associated with H_2, while we implement the *Generalized Leapfrog Integrator* (GLI). We insist on the fact that these two second-order methods share the same theoretical properties [Hairer et al., 2006]. Since IMI is implicit, they propose to approximate it with a numerical integrator Φ_h, which is computed with a fixed-point method (as we do) detailed in Algorithm 3. Finally, they implement CRHMC which contains the same steps as n-BHMC *apart from the involution checking step* (see Line 8 in Algorithm 1): (i) refreshing the momentum with a symplectic scheme (Step 1 in both algorithms), (ii) solving the discretized ODE associated to H in three steps according to the splitting given by H_1 and H_2 (Step 2 in both algorithms), (iii) applying a Metropolis-Hastings filter (Step 3 in both algorithms), and (iv) operating a final momentum reversal (Step 4 in n-BHMC). They state in Theorem 8 that the Markov kernel corresponding to one iteration of CRHMC satisfies detailed balance with respect to $\bar{\pi}$, thus ensuring that it preserves $\bar{\pi}$.

A.10.2 Theoretical gaps in the reversibility of CRHMC Kook et al. [2022]

First, we call into question the proof of the reversibility of c-CRHMC stated in [Kook et al., 2022, Theorem 6]. Indeed, Kook et al. [2022] implicitly assume that the time-continuous Hamiltonian dynamics have a solution at any time, but do not provide any proof of this result. We emphasize here that this result is not trivial and possibly may not hold due to the pathological behaviour of the barrier at the border of M, see Proposition 35.

In contrast, the involution checking step implemented in c-BHMC verifies this condition for the *forward* and the *reversed* dynamics after momentum reversal, which then allows us to derive the reversibility of c-BHMC, see Theorem 36.

Finally, we report some gaps in the proof of reversibility of CRHMC provided in [Kook et al., 2022, Theorem 8]. Indeed, while Kook et al. [2022] claim that [Hairer et al., 2006, Theorem VI.3.5.] applies to CRHMC which contains the *numerical* integrator detailed in [Kook et al., 2022, Algorithm 3], in fact [Hairer et al., 2006, Theorem VI.3.5.] only applies to the *implicit* integrator (IMI). This is a fundamental limitation of their theory. This theoretical confusion thus leads to numerical errors, as we detail in Section 2.6. In contrast, we present in Section 2.4 theoretical results on the implicit integrator (GLI), which allows us to derive reasonable assumptions on the numerical integrator Φ_h to obtain reversibility of n-BHMC in Theorem 3. Contrary to Kook et al. [2022], our proof relies on the properties of Φ_h and highlights the critical role of the involution checking step.

A.11 Experimental details

The numerical experiments presented in Section 2.6 are based on the MATLAB implementation of CRHMC provided by Kook et al. [2022]. We adapt this code for n-BHMC for sake of fairness. In particular, our implementation differs from CRHMC by the use of the Generalized Leapfrog integrator and the "involution checking" mechanism.

Details on experiments with synthetic data. In these experiments, the hypercube refers to the set $[-1/2, 1/2]^d$ where $d \in \{5, 10\}$. We recall that the ground truth on the quantities we estimate is given by the Metropolis Adjusted Langevin Algorithm (MALA) [Roberts and Stramer, 2002] for the hypercube and the Independent Metropolis Hastings (IMH) sampler [Liu, 1996] for the simplex. We now give details on how the parameters of these two algorithms are chosen. For MALA, we use a constant step-size $h = 0.05$ for both dimensions, which results in an average acceptance probability equal to 0.55 for $d = 5$ and 0.44 for $d = 10$. For IMH, we use as proposal the uniform distribution, which is simply a Dirichlet distribution, resulting in an average acceptance probability of order 0.36. We recall that we use an adaptive step-size h in CRHMC and n-BHMC such that we obtain an average acceptance probability of order 0.5 in the MH filter. We now discuss the setting of the tolerance parameter η in n-BHMC. We choose η so that (i) the step-size h is at least greater than 10^{-2} over the iterations of the algorithm and (ii) the average acceptance probability is roundly equal to 0.5. This heuristic results in defining (a) for the hypercube, $\eta = 5$ if $d = 5$ and $\eta = 10$ if $d = 10$ and (b) for the simplex, $\eta = 10$ if $d = 5$ and $\eta = 200$ if $d = 10$.

Details on experiments with real-world data. For these experiments, we consider the exact same setting as in Kook et al. [2022], using the same adaptation strategies to update the step-size and the barrier functions.

Influence of the norm in (2.18). The norm chosen for the "involution checking step" is arbitrary and many options are available. In particular, one could design $\|z^{(0)} - \Phi_h(z^{(1)})\|_2 > \eta$, and thus pick the Euclidean norm in (2.18). However, while this approach should theoretically also reduce the bias in the method, we remark that the obtained Markov chains have very poor mixing time. This is due to the fact that in practice the update on the momentum is of order $\|\mathfrak{g}(x)\|_2^{1/2}$ while the update on the position is of order $\|\mathfrak{g}(x)^{-1}\|_2^{1/2}$. Hence the Euclidean norm is ill-suited for the "involution checking step" and the proposal states are rejected a lot more near the boundaries, see Figure A.1.

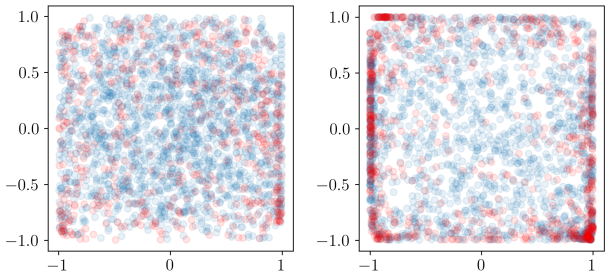


Figure A.1: Outputs of n-BHMC after 25k iterations to sample from the uniform distribution over $[-1, 1]^2$ with $\eta = 10^{-3}$, $h = 0.8$, using in (2.18) the "self-concordant" norm (left) or the Euclidean norm (right). Red samples are rejected and blue samples are accepted.

Appendix B

Appendix of Chapter 3

B.1 Details on considered samplers

In the following, we present a brief description of the samplers presented in Section 3.3, and used in our numeric evaluation. Throughout this section, we will consider an arbitrary test function $f : \mathbb{R}^d \to \mathbb{R}$, for which we would like to evaluate the expectation $\mathbb{E}_\pi[f] := \int_{\mathbb{R}^d} f(x)\mathrm{d}\pi(x)$, and will explain how to compute Monte Carlo estimations $\mathbb{E}_\pi[f]$ for each sampler. Note that in our numerical experiment, we will take $f = \mathbb{1}_{S_1}$. Below, p^{init} will refer to an easy-to-sample distribution, with tractable density and normalization constant (basically, a Gaussian distribution).

MALA sampler. This algorithm is an extension of the largely used Unadjusted Langevin Algorithm (ULA) [Roberts and Tweedie, 1996], which builds a Markov chain $(X^\ell)_{\ell=0}^L$ with $(L+1)$ states ($L \geq 1$), defined by $X^0 \sim p^{\text{init}}$ and for any $\ell \in \{0, \dots, L-1\}$,

$$X^{\ell+1} = X^\ell + \lambda \nabla \log \gamma(X^\ell) + \sqrt{2\lambda} Z^\ell \ , \ Z^\ell \sim \mathrm{N}(0, \mathrm{I}_d) \ , \tag{B.1}$$

which reduces to a time discretization of the Langevin diffusion that admits π as invariant distribution. Building upon this recursion, MALA [Roberts and Tweedie, 1996] addresses the time discretization bias by adding a Metropolis-Hastings (MH) acceptance/rejection step at each iteration ℓ based on the acceptance probability

$$\alpha(x, x') = \min\left(1, \frac{\gamma(x')\mathrm{N}(x; x' + \lambda\nabla\log\gamma(x'), 2\lambda\mathrm{I}_d)}{\gamma(x)\mathrm{N}(x'; x + \lambda\nabla\log\gamma(x), 2\lambda\mathrm{I}_d)}\right) \ . \tag{B.2}$$

In this case, the corresponding Monte Carlo estimation of $\mathbb{E}_\pi[f]$ will be given by $L^{-1}\sum_{\ell=1}^L f(X^\ell)$.

Importance Sampling. Given an arbitrary proposal probability distribution ν with tractable density q known up to a normalizing constant such that $\pi \ll \nu$, Importance Sampling (IS) relies on the following equality

$$\mathbb{E}_\pi[f] = \mathbb{E}_\nu\left[\frac{\mathrm{d}\pi}{\mathrm{d}\nu} f\right] = \frac{1}{\mathcal{Z}}\mathbb{E}_\nu\left[\frac{\gamma}{q} f\right] \ . \tag{B.3}$$

Hence, the IS Monte Carlo estimation of $\mathbb{E}_\pi[f]$ is given by the weighted mean $N^{-1}\sum_{n=1}^N w_n(X^n) f(X^n)$, where $(X^n)_{n=1}^N$ are samples from ν and $\{w_n(X^n)\}_{n=1}^N$ are self-normalized importance weights defined by

$$w_n(X^n) = \frac{\tilde{w}_n(X^n)}{\sum_{m=1}^N \tilde{w}_m(X^m)} \ , \ \tilde{w}_n(X^n) = \frac{\gamma(X^n)}{q(X^n)} \ . \tag{B.4}$$

In the special case where the proposal is defined by a normalizing flow $\mathrm{T}_\theta : \mathbb{R}^d \to \mathbb{R}^d$ with base distribution p^{init}, the corresponding parametric density is given by

$$q_\theta(x) = p^{\text{init}}(\mathrm{T}_\theta^{-1}(x)) \left| J_{\mathrm{T}_\theta^{-1}}(x) \right| \ . \tag{B.5}$$

where $J_{\mathrm{T}_\theta^{-1}}$ denotes the Jacobian of T_θ^{-1}.

Variational Inference. Given a family of parametric distributions $\{\nu_\theta\}_{\theta\in\Theta}$ such that for any $\theta \in \Theta$, (i) ν_θ admits a tractable density q^θ known up to a normalizing constant and (ii) ν_θ is easy to sample from, VI aims to solve the following optimization problem

$$\theta^\star \in \arg\min_{\theta\in\Theta} \mathrm{D}(\nu_\theta|\pi) \ , \tag{B.6}$$

where D is a divergence[1] operating on the space of probability distributions. In most cases, D is defined as the reverse Kullback-Leibler divergence, i.e., we have $\mathrm{D}(\nu_\theta|\pi) = \mathrm{KL}(\nu_\theta|\pi) = \int_{\mathbb{R}^d} \log(q_\theta(x)/\gamma(x))\mathrm{d}\nu_\theta(x)$, where the equality holds up to constants independent from θ. In practice, this objective function is minimized by performing Stochastic Gradient Descent (SGD), based on samples from the variational distribution being trained. Assuming that $\hat{\theta}$ is the output of this numerical procedure, the corresponding VI Monte Carlo estimation of $\mathbb{E}_\pi[f]$ is given by $N^{-1}\sum_{n=1}^N f(X_{\hat{\theta}}^n)$, where $(X_{\hat{\theta}}^n)_{n=1}^N$ are samples from $\nu_{\hat{\theta}}$.

Annealed sampling. Given $K \geq 1$ the number of intermediate steps, the standard geometric density interpolation defines a sequence of target densities $\{p_k\}_{k=0}^K$ up to a normalizing constant by $p_k \propto (p^{\mathrm{init}})^{1-\beta_k}\gamma^{\beta_k}$, where $k \in \{0,\dots,K\}$ and $\{\beta\}_{k=0}^K$ is an increasing sequence with $\beta_0 = 0$ and $\beta_K = 1$. In the literature, $\{\beta\}_{k=0}^K$ is often referred to as the tempering schedule.

Sequential Monte Carlo. Given $N \geq 1$, SMC generates N chains of size $K+1$, referred to as particles, such that the k-th elements of all chains are expected to be samples from p_k. To do so, SMC relies *locally* on MALA steps performed independently on each chain and *globally* on resampling steps based on accumulated importance weights, as defined next.

Let $k \in \{0,\dots,K-1\}$, $(\tilde{w}_k^n)_{n=1}^N$ be unnormalized importance weights accumulated on each chain up to step k ($\tilde{w}_k^n = 1/N$ if $k=0$), and $(x_k^n)_{n=1}^N$ be samples from p_k obtained independently as the last samples of N parallel MALA runs (or exactly obtained if $k=0$). Then, the SMC importance weights designed to sample from p_{k+1}, denoted by $(\tilde{w}_{k+1}^n)_{n=1}^N$, and their renormalized versions, denoted by $(w_{k+1}^n)_{n=1}^N$, are defined as

$$\tilde{w}_{k+1}^n = \frac{p_{k+1}(x_k^n)}{p_k(x_k^n)}\tilde{w}_k^n, \quad w_{k+1}^n = \frac{\tilde{w}_{k+1}^n}{\sum_{j=1}^N \tilde{w}_{k+1}^j} \ . \tag{B.7}$$

The SMC resampling step then consists in obtaining N new samples $(y_{k+1}^n)_{n=1}^N$ by random selection among the original samples $(x_k^n)_{n=1}^N$ via a multinomial distribution (with replacement) defined by the normalized weights $(w_{k+1}^n)_{n=1}^N$. These novel samples will then be used as starting points for running MALA on the target p_{k+1}. In our implementation, we follow the recommendation of Chopin et al. [2020] and build the sequence $\{\beta_k\}_{k=0}^K$ on-the-fly. This is done by solving, at each step k, the following optimization problem

$$\delta_k^\star = \min\{\delta \in [0, 1-\beta_k] : \mathrm{ESS}(\delta) = \mathrm{ESS}_{\min}\} \ , \tag{B.8}$$

$$\text{where } \mathrm{ESS}(\delta) = \left[\left(\sum_{n=1}^N \tilde{w}_{\delta,k}^n\right)^2 / \sum_{n=1}^N (\tilde{w}_{\delta,k}^n)^2\right] \quad \tilde{w}_{\delta,k}^n = \frac{(p^{\mathrm{init}})^{1-(\beta_k+\delta)}(x_k^n)\gamma^{\beta_k+\delta}(x_k^n)}{p_k(x_k^n)} \ , \tag{B.9}$$

and $\mathrm{ESS}_{\min} = \alpha N$, with $\alpha \in (0,1)$ being a user-defined threshold. Then, the schedule update is given by $\beta_{k+1} = \beta_k + \delta_k^\star$. By doing so, this procedure enables to prevent the degeneracy of the particles at step $k+1$ (which occurs when the corresponding ESS is too low). In practice, we minimize this quantity using Brent's method. In the end, the SMC estimator of $\mathbb{E}_\pi[f]$ is given by the weighted mean $N^{-1}\sum_{n=1}^N w_K^n f(X_K^n)$, where $(X_K^n)_{n=1}^N$ are the last MALA samples obtained at step K.

Replica Exchange. The RE algorithm aims at sampling from the annealed distributions $\{p_k\}_{k=0}^K$ with $K+1$ parallel Markov chains, i.e., the k-th chain targets p_k. For the sake of pedagogy, we assume here that K is even.

At each level k, RE sampling is done via a local MCMC sampler, such as MALA, which is expected to have poor performance for large k (as p_k will be highly multi-modal). To alleviate this issue, every L MALA steps, RE randomly performs a *swapping* between Markov chains. More specifically, each consecutive pair $(k, k+1)$ of Markov chains (i.e., either $(k,k+1) \in \{(0,1),\dots,(K-2,K-1)\}$ or $(k,k+1) \in \{(1,2),\dots,(K-1,K)\}$), with respective current states X_k and X_{k+1}, is swapped with probability

$$p_{k,k+1} = \min\left(1, \frac{p_k(X_{k+1})p_{k+1}(X_k)}{p_k(X_k)p_{k+1}(X_{k+1})}\right) \ . \tag{B.10}$$

[1] Commonly, a divergence D verifies $\mathrm{D}(\nu \mid \pi) \geq 0$ and $\mathrm{D}(\nu \mid \pi) = 0$ if and only if $\nu = \pi$.

After running the Markov chains for N steps, we obtain a Monte Carlo estimation of $\mathbb{E}_\pi[f]$ by considering $N^{-1}\sum_{n=1}^N f(X_K^n)$, where $(X_K^n)_{n=1}^N$ are the visited states of the K-th chain (corresponding to the target distribution).

SLIPS. Grenioux et al. [2024] proposed a diffusion-based sampling method extending the stochastic localization framework from Montanari [2023], which can be seen as an analog of denoising diffusion models. More precisely, given a horizon time $T > 0$, they consider a stochastic process $(Y_t)_{t\in[0,T]}$, called observation process, defined by

$$Y_t = \alpha(t)X + \sigma W_t \ , \ X \sim \pi \ , \tag{B.11}$$

where $\sigma \in (0,\infty)$, $\alpha : [0,T) \to [0,\infty)$ is an increasing function of time verifying $\alpha(0) = 0$ and $\alpha(t)/\sqrt{t} \to \infty$ as $t \to T$, and $(W_t)_{t\geq 0}$ is a d-dimensional Brownian motion. A particular choice of interest is given by $\alpha(t) = \sqrt{t/(1-t)}$, referred to as Geom$(1,1)$, with $T = 1$.

By definition of α, this observation process *localizes* on the target distribution as time increases since $Y_t/\alpha(t) \to X$ as $t \to T$. Hence, if we were able to simulate the stochastic process $(Y_t)_{t\in[0,T]}$ up to time T (which is infeasible a priori since we do not have access to ground truth samples), we would then obtain approximate samples from π. To do so, the authors propose to simulate the Markov process induced by the following SDE, which is proved to have the same marginals as the observation process, denoted by $(p_t)_{t\in[0,T]}$,

$$\mathrm{d}Y_t = \dot{\alpha}(t)u_t(Y_t)\mathrm{d}t + \sigma \mathrm{d}B_t \ , Y_0 = 0 \ , \tag{B.12}$$

where $(B_t)_{t\geq 0}$ is another d-dimensional Brownian motion and $u_t(y_t) = \mathbb{E}[X|Y_t = y_t]$ is the conditional expectation of the posterior distribution whose unnnormalized density is given by $q_t(x|y_t) : x \mapsto \mathrm{N}(y_t; \alpha(t)x, \sigma^2 t \mathrm{I}_d)\gamma(x)$. This drift being intractable, the authors propose to estimate it on-the-fly, i.e., while running a discretized version of the SDE (B.12), using MCMC methods to target the posterior q_t. They justify their method by highlighting the phenomenon *duality of log-concavity*, that is, while p_t gets increasingly multi-modal, q_t gets increasingly more log-concave and therefore, easier to sample from with standard MCMC techniques. Overall, the success of this method relies on tuning a critical hyperparameter $t_0 \in (0,T)$, which denotes the effective starting time of the generative SDE, such that both p_{t_0} and q_{t_0} are approximately log-concave. While some heuristics have been proposed by the authors to solve this issue, they may be not effective on arbitrary multi-modal distributions. For this algorithm, given a time-step t_1 close to T, the corresponding Monte Carlo estimation of $\mathbb{E}_\pi[f]$ is given by $N^{-1}\sum_{n=1}^N f(Y_{t_1}^n/\alpha(t_1))$, where $(Y_{t_1}^n)_{n=1}^N$ are obtained by running independently N times the approximated version of (B.12), up to time t_1.

DDS. This parameterized diffusion-based sampling method can be seen as the analog of the Variance-Preserving (VP) score-based generative model previously introduced by [Song et al., 2021b], in the setting where ground truth samples are not available. More precisely, given a time horizon $T > 0$ and $\sigma \in (0,\infty)$, DDS aims to sample from a *denoising diffusion process* $(Y_t)_{t\in[0,T]}$ defined as the time-reversal of the Ornstein-Uhlenbeck process $(X_t)_{t\in[0,T]}$ induced by the following so-called *noising* SDE

$$\mathrm{d}X_t = -\frac{\beta(t)}{2}X_t\mathrm{d}t + \sigma\sqrt{\beta(t)}\mathrm{d}W_t \ , X_0 \sim \pi \ , \tag{B.13}$$

where $(W_t)_{t\geq 0}$ is a d-dimensional Brownian motion and $\beta : [0,T] \to (0,\infty)$ is a noising schedule chosen such that it holds approximately $X_T \sim \mathrm{N}(0, \sigma^2\mathrm{I}_d)$. In particular, it holds that $Y_t = X_{T-t}$ in the distribution sense, and, under mild assumptions on π and β, $(Y_t)_{t\in[0,T]}$ is solution to the following non-linear SDE

$$\mathrm{d}Y_t = \frac{\beta(T-t)}{2}Y_t\mathrm{d}t + \sigma^2\beta(T-t)\nabla\log p_{T-t}(Y_t)\mathrm{d}t + \sigma\sqrt{\beta(t)}\mathrm{d}B_t \ , Y_0 \sim \mathrm{N}(0, \sigma^2\mathrm{I}_d) \ , \tag{B.14}$$

where $(B_t)_{t\geq 0}$ is another d-dimensional Brownian motion and p_t denotes the density of the marginal distribution of X_t, which is intractable. Hence, in the same spirit as SLIPS, if we were able to simulate the stochastic process $(Y_t)_{t\in[0,T]}$ up to time T (which is infeasible a priori since the scores $\nabla\log p_t$ are intractable), we would then obtain approximate samples from π. To do so, Vargas et al. [2023a] propose to learn the scores by solving the following Variational Inference problem, which does not require access to ground truth samples,

$$\theta^\star \in \arg\min_{\theta\in\Theta} \mathrm{KL}(\mathbb{P}^\theta|\mathbb{P}) \ , \tag{B.15}$$

where $\mathbb{P}$ is the path measure associated to the true reverse process, and for any $\theta \in \Theta$, $\mathbb{P}^\theta$ is the path measure associated to the neural SDE

$$\mathrm{d}Y_t = \frac{\beta(T-t)}{2} Y_t \mathrm{d}t + \sigma^2 \beta(T-t) s_\theta(T-t, Y_t)\mathrm{d}t + \sigma\sqrt{\beta(t)}\mathrm{d}B_t \ , Y_0 \sim \mathrm{N}(0, \sigma^2 \mathrm{I}_d) \ , \tag{B.16}$$

where $s_\theta : [0,T] \times \mathbb{R}^d \to \mathbb{R}^d$ is a neural network learnt such that $s_\theta(t,\cdot) \approx \nabla \log p_t$. To obtain a tractable objective function, Vargas et al. [2023a] propose to parameterize s_θ via a linear term as $s_\theta(t,x) = -\frac{x}{\sigma^2} + u_\theta(t,x)$, where u_θ can be interpreted as a guidance. After time discretization, the optimization procedure to learn $\theta^\star$ amounts to perform VI on joint densities. Note that the reverse KL divergence may be replaced by the Log-Variance divergence for increased performance Richter et al. [2023]. Assuming that $\hat{\theta}$ is the output of this numerical procedure, the corresponding DDS Monte Carlo estimation of $\mathbb{E}_\pi[f]$ is given by $N^{-1}\sum_{n=1}^N f(Y^n_{\hat{\theta},T})$ where $(Y^n_{\hat{\theta},T})_{n=1}^N$ are obtained by running independently N times the neural SDE (B.16) with control term $s_{\hat{\theta}}$, up to time T.

B.2 Implementation details

Details on the multi-modal target distribution. In this paper, we consider the Gaussian mixture with density defined by

$$\gamma : x \in \mathbb{R}^d \mapsto \frac{2}{3}\overbrace{\mathrm{N}(x; -a\mathbf{1}_d, \Sigma_1)}^{q_1(x)} + \frac{1}{3}\underbrace{\mathrm{N}(x; a\mathbf{1}_d, \Sigma_2)}_{q_2(x)} \ , \tag{B.17}$$

where $\Sigma_1 \in \mathbb{R}^{d\times d}$ and $\Sigma_2 \in \mathbb{R}^{d\times d}$ are both diagonal matrices respectively defined by $(\Sigma_1)_{i,i} = \frac{i}{d}\sigma^2_{\max} + \frac{d-i}{d}\sigma^2_{\min}$ for $i \in \{1,\dots,d\}$ and $(\Sigma_2)_{i,i} = (\Sigma_1)_{d-i,d-i}$ for $\sigma^2_{\max} = 0.2$ and $\sigma^2_{\min} = 0.01$. In our numerics, we considered the following range of hyperparameters: $a \in \{0.5, 2.875, 5.25, 7.625, 10\}$ and $d \in \{4, 8, 16, 32, 64\}$ (and the additional settings $d \in \{128, 256\}$ for SMC, RE and SLIPS). Below, we will consider the Gaussian approximation of π defined by $\mathrm{N}(\mathbf{m}_\pi, \Sigma_\pi)$, where $\mathbf{m}_\pi$ is the (tractable) mean of π and Σ_π is the diagonal positive matrix such that for any $i \in \{1,\dots,d\}$, $\mathrm{Diag}(\Sigma_\pi)_i$ is the marginal variance of π on the i-th axis (which is also tractable).

Setting of MALA. To obtain one Monte Carlo estimation, we consider 32 independent chains randomly initialized in the strongest mode of π (i.e., in S_1). While running the Markov chains, we automatically adapt the step-size λ (initialized at 10^{-4}) by targeting a specific MH acceptance ratio fixed to 0.75. For each chain, we first run 4096 warm-up steps, and then run MALA for 8192 steps to obtain as many samples. This procedure produces 8192×32 samples which will be used to compute the mode weight estimation. We repeat this process 48 times to obtain aggregate results.

Setting of IS. To ensure an ideal setting for IS estimation, we set the proposal ν to be closely adapted to π, while avoiding giving access to the knowledge of the true mode proportions. This is done by minimizing the forward Kullback-Leibler divergence between the *equilibrated* version of π, defined by the mixture $x \mapsto \frac{1}{2}q_1(x) + \frac{1}{2}q_2(x)$, and the chosen parametric family of distributions (namely, Gaussian distributions or normalizing flows). Note that sampling from the equilibrated mixture is highly similar to directly sampling from the target distribution with an MCMC procedure, where the location of the modes is given. Regarding the parametric family used:

- Gaussian distributions: we compute maximum likelihood estimators of the ideal parameters, based on 16384 samples of the equilibrated target;
- Normalizing flows (NFs): we restrict our choice of parametrization to the RealNVP architecture [Dinh et al., 2017], which is composed of 32 affine coupling blocks with scale and shifts parametrized as Multi-Layer Perceptrons (MLPs) with 3 hidden layers of size 64. The base of the NF is a standard centered Gaussian distribution at temperature 0.5. The NF is trained on a dataset of 16384 samples for 100 epochs with a batch size of 1024 samples. We follow the implementation from Stimper et al. [2023].

Once training the proposal is done, we compute for each setting 48 IS estimators of the mode weight, using 8192 particles each time.

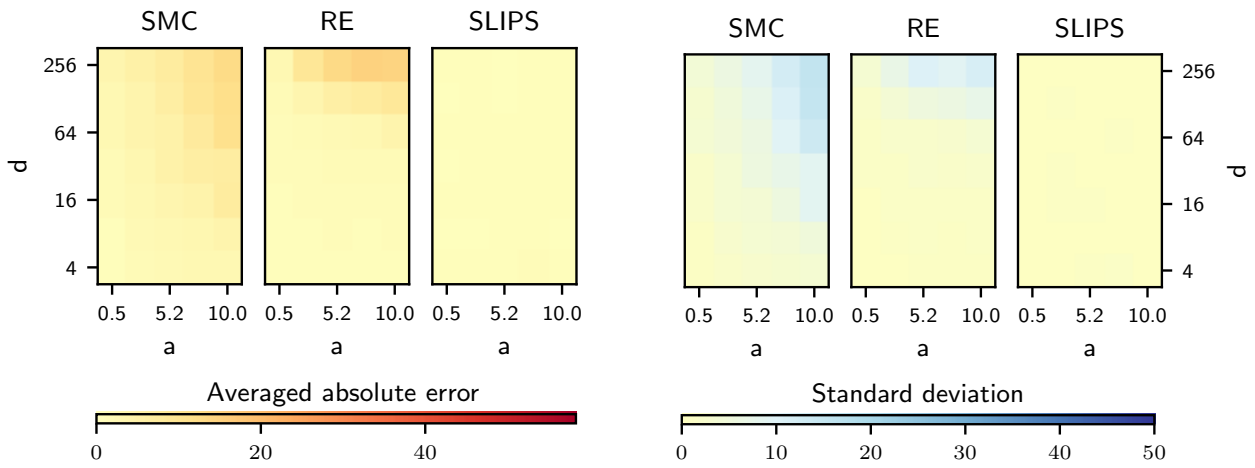


Figure B.1: Same as Figure 3.2 for the best performing SMC, RE and SLIPS in higher-dimensions. Note that the color scale is the same as in Figure 3.2. **(Left):** Averaged absolute error of the estimation with respect to the true mode weight $w \approx 66.7\%$. **(Right):** Standard deviation of the estimation.

Setting of VI. We consider two different variational families :

• Gaussian distributions: we restrict our choice to distributions with diagonal covariances. The initial parameters of the VI procedure are taken as the exact mean of π and the diagonal of the exact covariance of π. In practice, we parameterize the log-variances instead of the variances;

• Normalizing flows (NFs): we restrict our choice of parametrization to the RealNVP architecture [Dinh et al., 2017], which is composed of 32 affine coupling blocks with scale and shifts parametrized as MLPs with 3 hidden layers of size 64. The base of the NF is a standard centered Gaussian distribution. We follow the implementation from Stimper et al. [2023].

In both cases, we minimize the reverse Kullback-Leibler divergence for 2048 SGD iterations with a batch size of 2048. Once the optimization is done, we sample 8192 from the obtained optimal parametric distribution to compute the mode weight estimation (this is done 48 times).

Setting of SMC. The base distribution p^{init} is given by $\mathrm{N}(\mathbf{m}_\pi, \Sigma_\pi)$. We set the maximum number of intermediate steps as 512 and $\mathrm{ESS}_{\min} = N/2$ (i.e., $\alpha = 1/2$) as per Chopin et al. [2020]. Locally, we perform MALA sampling, with adaptive step-size (initialized at 10^{-2}), for 96 steps. To ensure efficiency over the sequential procedure, we initialize the step-size of the $(k+1)$-th sampler with the last value of the step-size from the k-th sampler. Overall, we run SMC 48 times, with $N = 8192$ particles each time to compute the mode weight.

Setting of RE. The base distribution p^{init} is given by $\mathrm{N}(\mathbf{m}_\pi, \Sigma_\pi)$. We consider $K = 64$ intermediate steps and the schedule $\{\beta_k\}_{k=0}^K$ to be logarithmically spaced between $\varepsilon = 10^{-5}$ and 1, i.e., $\beta_k = 1 - \varepsilon^{(k/K)}$ for $k \in \{0, \dots, K\}$. In our experiments, we observed that this particular choice of schedule (unlike the linear schedule) has benefits to prevent the issue of mode switching in geometric density interpolations [Máté and Fleuret, 2023]. Such as in SMC, we locally perform MALA sampling, with adaptive step-size (initialized at 10^{-2}). In particular, we enable the swapping every 8 MCMC steps. Overall, we perform 16384 warm-up steps at each annealing level and then 32768 steps to generate the samples, for 16 parallel instances of the algorithm. In the end, we perform a thinning by only selecting one sample every 8 steps in the K-th chain (corresponding to the target distribution), and randomly sub-sample 8192 samples out of all the obtained samples to compute the mode weight estimation. We repeat this procedure 48 times.

Setting of SLIPS. We run the SLIPS algorithm with the same setting as in the original paper, see Grenioux et al. [2024], but we rather use 512 discretization timesteps. Moreover, the Gaussian initialization procedure of SLIPS at time t_0 is preserved, but with the additional information of the true mean of the target distribution. As in the original paper, we tune the t_0 hyperparameter by leveraging access to ground truth samples. Overall, to obtain the Monte Carlo estimations, we run SLIPS 48×8192 times.

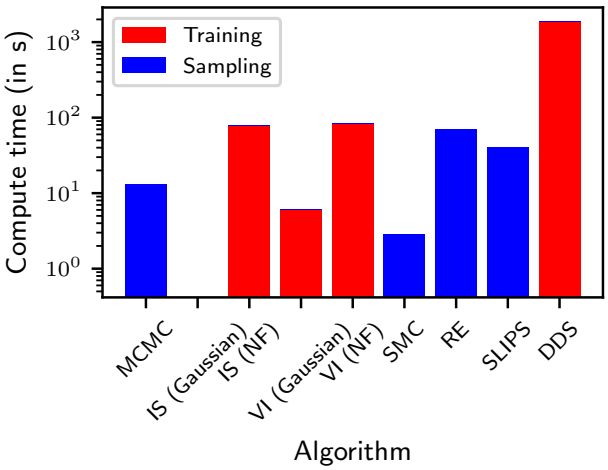


Figure B.2: Average wall-clock computing time of each sampling algorithm. The results were averaged on 48×25 runs (48 Monte Carlo estimations for each of the 25 different mode separation/dimension settings).

Setting of DDS. We run the DDS sampler using the code from Berner et al. [2023], using 100 integration steps uniformly distributed between 0 and T. Following the guidelines from Vargas et al. [2023a], we use a target-informed parametrization of the neural network u^θ. Moreover, the hyperparameter σ is set according to the recommendations of Noble et al. [2025]

$$\sigma^2 = \frac{1}{d}\left(m^\top m + \sum_{i=1}^{d} \Gamma_i^2\right) \tag{B.18}$$

where $m = \mathbb{E}_\pi[X] = a(2-w)\mathbf{1}_d$ and for $i \in \{1, \ldots, d\}$,

$$\Gamma_i^2 = w\{(\Sigma_1)_{ii} + (a+m_i)^2\} + (1-w)\{(\Sigma_2)_{ii} + (a-m_i)^2\} \,. \tag{B.19}$$

Besides this, we train 16 times the DDS log-variance objective for 4096 gradient steps using a batch size of 1024 samples, and for each training, and compute, for each training, 3 Monte Carlo estimations of the mode weight based on 8192 generated samples (hence, overall 48 MC estimations).

Computational footprint. Figure B.2 reports the average computing time of each algorithm. Note that, because of the on-the-fly discretization, the computational budget of SMC is stochastic. By way of indication, the budget of RE, SLIPS as well as the NF approaches would align with a 512 steps SMC algorithm.

Appendix C

Appendix of Chapter 4

Organization of the supplementary

The appendix is organized as follows. Appendix C.1 summarizes general facts that will be useful for proofs. Appendix C.2 provides details on the Stochastic Localization (SL) frameworks introduced in Section 4.2 and Section 4.3. In Appendix C.3, we derive detailed computations on the application of SL to Gaussian mixtures. Appendix C.4 presents an implementation of SL for sampling in the case where the score of the observation process is available. Appendix C.5 dispenses theoretical results on the phenomenon of "duality of log-concavity" highlighted in Section 4.4.2 and Section 4.4.3. In Appendix C.6, we detail the implementation of our sampling algorithm called `SLIPS` and provide an ablation study of its hyper-parameters. A detailed comparison between our approach and works from Huang et al. [2024a] and Saremi et al. [2024] is given in Appendix C.7. Finally, Appendix C.8 provides more details on the numerical experiments presented in Section 4.6.

Notation. To alleviate the computations derived below, for a probability distribution μ defined on $\mathbb{R}^d$, and any measurable function $\varphi : \mathbb{R}^d \to \mathbb{R}^d$, we will denote the expectation $\int_{\mathbb{R}^d} \varphi(x) \mathrm{d}\mu(x) \in \mathbb{R}^d$ by $\mathbb{E}[\varphi(X)]$ and the covariance $\int_{\mathbb{R}^d} (\varphi(x) - \mu(\varphi))(\varphi(x) - \mu(\varphi))^\top \mathrm{d}\mu(x) \in \mathbb{R}^{d \times d}$ by $\mathrm{Cov}[\varphi(X)]$, where X is a random vector distributed according to μ. For any $T \in (0, +\infty]$, two functions f and g defined on $[0, T)$ are said to be asymptotically equivalent if $f(t)/g(t) \to 1$ and $g(t)/f(t) \to 1$ as $t \to T$. This is denoted by $f(t) \sim g(t)$.

C.1 Preliminaries

Metrics. We recall that the 2-Wasserstein distance between two distributions μ and ν is given by

$$W_2(\mu, \nu) = \inf\{\textstyle\int_{\mathbb{R}^d \times \mathbb{R}^d} \|x_1 - x_0\|^2 \mathrm{d}\pi(x_0, x_1) : \pi_0 = \mu, \pi_1 = \nu\}^{1/2} , \tag{C.1}$$

where π_i denotes the i-th marginal of π for $i \in \{0, 1\}$. While it is not tractable in general, we have an explicit expression when comparing two Gaussian distributions. Consider $\mu = \mathrm{N}(\mathbf{m}_1, \gamma_1^2 \, \mathrm{I}_d)$ and $\nu = \mathrm{N}(\mathbf{m}_2, \gamma_2^2 \, \mathrm{I}_d)$, where $(\mathbf{m}_1, \mathbf{m}_2) \in \mathbb{R}^d \times \mathbb{R}^d$ and $(\gamma_1, \gamma_2) \in (0, \infty)^2$. Then, following [Peyré et al., 2019, Equation 2.41], we have

$$W_2(\mu, \nu)^2 = \|\mathbf{m}_1 - \mathbf{m}_2\|^2 + d(\gamma_1 - \gamma_2)^2 . \tag{C.2}$$

We also recall that the entropic regularized 2-Wasserstein distance between two distributions μ and ν is defined by

$$W_{2,\varepsilon}(\mu, \nu) = \inf\{\textstyle\int_{\mathbb{R}^d \times \mathbb{R}^d} \|x_1 - x_0\|^2 \mathrm{d}\pi(x_0, x_1) - \mathscr{H}(\pi) : \pi_0 = \mu, \pi_1 = \nu\}^{1/2} , \tag{C.3}$$

where $\varepsilon > 0$ is a regularization hyper-parameter and $\mathscr{H}(\pi) = -\int_{\mathbb{R}^d \times \mathbb{R}^d} \log \pi(x_0, x_1) \mathrm{d}\pi(x_0, x_1)$ refers to as the entropy of π.

Finally, we recall that the Kolmogorov-Smirnov distance between two distributions μ and ν is defined by

$$\mathrm{KS}(\mu, \nu) = \sup_{x \in \mathbb{R}^d} |F_\mu(x) - F_\nu(x)| , \tag{C.4}$$

where F_μ (respectively F_ν) denotes the cumulative distribution function of μ (respectively ν).

Below, we provide a first result, known as the Tweedie's formula, which gives the expression of the score of any distribution that writes as a marginalization, and its derivative.

Lemma 52 (Tweedie's formula and extension)**.** *Let p be a positive probability density on $\mathbb{R}^d$ such that for any $y \in \mathbb{R}^d$ $p(y) = \int_{\mathbb{R}^d} p(y|x)\mathrm{d}\mu(x)$, where $(x, y) \mapsto p(y|x)$ is a positive transition density on $\mathbb{R}^d \times \mathbb{R}^d$, and μ is a probalility mesure on $\mathbb{R}^d$. Assume that for any $x \in \mathbb{R}^d$, $y \mapsto \log p(y|x)$ is twice continuously differentiable and that, for any $k \in \{1, 2\}$, there exists $\varphi_k : \mathbb{R}^d \to \mathbb{R}_+$ such that $\int_{\mathbb{R}^d} \varphi_k(x)\mathrm{d}\mu(x) < \infty$ and for any $(x, y) \in \mathbb{R}^d \times \mathbb{R}^d$, we have $\left\|\nabla_y^k p(y|x)\right\| \leq \varphi_k(x)$.*

For any $y \in \mathbb{R}^d$, define the 'posterior' density $x \mapsto p(x|y)$ by $p(x|y) = p(y|x)p(x)/p(y)$.

Then, we have for any $y \in \mathbb{R}^d$

$$\nabla_y \log p(y) = \int_{\mathbb{R}^d} \nabla_y \log p(y|x) p(x|y)\mathrm{d}x \; . \tag{C.5}$$

This is often referred to as the Tweedie's formula Robbins [1992]. We also have for any $y \in \mathbb{R}^d$

$$\nabla_y^2 \log p(y) = \int_{\mathbb{R}^d} \nabla_y^2 \log p(y|x) p(x|y)\mathrm{d}x + \int_{\mathbb{R}^d} \nabla_y \log p(y|x) \{\nabla_y \log p(y|x)\}^\top p(x|y)\mathrm{d}x \tag{C.6}$$

$$- \left\{ \int_{\mathbb{R}^d} \nabla_y \log p(y|x) p(x|y)\mathrm{d}x \right\} \left\{ \int_{\mathbb{R}^d} \nabla_y \log p(y|x) p(x|y)\mathrm{d}x \right\}^\top \; . \tag{C.7}$$

For any $y \in \mathbb{R}^d$, these two results can be reformulated as

$$\nabla_y \log p(y) = \mathbb{E}[\nabla_y \log p(y|X)] \; , \; \nabla_y^2 \log p(y) = \mathbb{E}[\nabla_y^2 \log p(y|X)] + \mathrm{Cov}[\nabla_y \log p(y|X)] \; , \tag{C.8}$$

where X is a random vector distributed according to the posterior distribution defined by $\mathrm{d}\mu^y(x) = p(x|y)\mathrm{d}x$.

Proof. First observe that for any $x \in \mathbb{R}^d$, $y \mapsto p(y|x)$ is twice continuously differentiable since $p(y|x) = \exp(\log p(y|x))$, where $y \mapsto \log p(y|x)$ is twice continuously differentiable.

We begin with the proof of (C.5). By combining the assumptions of the lemma with the dominated convergence theorem, we obtain that $p \in \mathrm{C}^2(\mathbb{R}^d, (0, \infty))$, and that for any $y \in \mathbb{R}^d$, we have

$$\nabla_y \log p(y) = \frac{\nabla_y p(y)}{p(y)} = \frac{\nabla_y \int_{\mathbb{R}^d} p(y|x)\mathrm{d}\mu(x)}{\int_{\mathbb{R}^d} p(y|x)\mathrm{d}\mu(x)} = \frac{\int_{\mathbb{R}^d} \nabla_y \log p(y|x) p(y|x)\mathrm{d}\mu(x)}{\int_{\mathbb{R}^d} p(y|x)\mathrm{d}\mu(x)} = \int_{\mathbb{R}^d} \nabla_y \log p(y|x) p(x|y)\mathrm{d}x \; , \tag{C.9}$$

We now turn to the proof of (C.6). Note that we have $\nabla_y \log p(y) = p(y)^{-1} \int_{\mathbb{R}^d} \nabla_y \log p(y|x) p(y|x)\mathrm{d}\mu(x)$. Then, using once again the dominated convergence theorem, we obtain that for any $y \in \mathbb{R}^d$,

$$\begin{aligned}
\nabla_y^2 \log p(y) &= \frac{1}{p(y)} \int_{\mathbb{R}^d} \nabla_y^2 \log p(y|x) p(y|x)\mathrm{d}\mu(x) + \frac{1}{p(y)} \int_{\mathbb{R}^d} \nabla_y \log p(y|x) \{\nabla_y \log p(y|x)\}^\top p(y|x)\mathrm{d}\mu(x) \\
&\quad - \frac{\nabla_y p(y)}{p(y)^2} \left\{ \int_{\mathbb{R}^d} \nabla_y \log p(y|x) p(y|x)\mathrm{d}\mu(x) \right\}^\top \\
&= \int_{\mathbb{R}^d} \nabla_y^2 \log p(y|x) p(x|y)\mathrm{d}x + \int_{\mathbb{R}^d} \nabla_y \log p(y|x) \{\nabla_y \log p(y|x)\}^\top p(x|y)\mathrm{d}x \\
&\quad - \nabla_y \log p(y) \left\{ \frac{1}{p(y)} \int_{\mathbb{R}^d} \nabla_y \log p(y|x) p(y|x)\mathrm{d}\mu(x) \right\}^\top \; ,
\end{aligned}$$

which gives the result. □

We now dispense a useful lemma to compute exact integration in SDEs with linear drift.

Lemma 53. *Let $T > 0$. Consider the SDE defined on $[0, T]$ by $\mathrm{d}Y_t = \beta_t(Y_t + b)\mathrm{d}t + \sigma \mathrm{d}B_t$, where β is integrable on $[0, T]$, $b \in \mathbb{R}^d$, and $\sigma > 0$. For any $T \geq t > s \geq 0$, the conditional density of Y_t given $Y_s = y_s$, denoted by $p_{t|s}(\cdot|y_s)$, verifies*

$$p_{t|s}(y_t|y_s) = \mathrm{N}(y_t; \exp(\textstyle\int_s^t \beta_u \mathrm{d}u) y_s + (\exp(\int_s^t \beta_u \mathrm{d}u) - 1)b, \sigma^2 \int_s^t \exp(2 \int_u^t \beta_r \mathrm{d}r)\mathrm{d}u \, \mathrm{I}_d) \; . \tag{C.10}$$

Proof. Define $\gamma(t) = \exp(-\int_0^t \beta_u \mathrm{d}u)$. Consider the stochastic process $Z_t = \gamma(t)Y_t$. By Îto's formula, we have $\mathrm{d}Z_t = \beta_t\gamma(t)b\mathrm{d}t + \gamma(t)\sigma\mathrm{d}B_t = -\dot{\gamma}(t)b\mathrm{d}t + \gamma(t)\sigma\mathrm{d}B_t$. By integrating this SDE between s and t, we have

$$\gamma(t)Y_t - \gamma(s)Y_s = \{\gamma(s) - \gamma(t)\}b + \sigma\int_s^t \gamma(u)\mathrm{d}B_u \ ,$$

and then

$$Y_t = \exp(\int_s^t \beta_u \mathrm{d}u)Y_s + (\exp(\int_s^t \beta_u \mathrm{d}u) - 1)b + \sigma\int_s^t \exp(\int_u^t \beta_r \mathrm{d}r)\mathrm{d}B_u \ ,$$

which gives the result using Îto's isometry and the fact that Y_s is independent from $(B_t - B_s)_{t\in[s,T]}$. □

C.2 Details on Stochastic Localization and our extension

Vocabulary of Stochastic Localization (SL). For sake of clarity, we precise below some terms employed in our paper:

- Y_t^α: observation process,
- g: denoising schedule,
- u_t^α : denoiser function,
- $u_t^\alpha(Y_t^\alpha)$: denoiser or Bayes estimator,
- $q_t^\alpha(\cdot|Y_t^\alpha)$: random posterior density. In the geometric measure theory literature, this is referred to as the SL process.

C.2.1 Reminders and intuition on generalized stochastic localization

Connection between standard stochastic localization and diffusion models. Stochastic localization as defined in (4.1) is equivalent to Variance-Preserving (VP) diffusion models Song et al. [2021b] under change of time. Indeed, if we define $\Phi(t) = \log(1 + \sigma^2 t^{-1})/2$ for any $t \geq 0$, with the convention that $\Phi(0) = \infty$, and consider the Ornstein-Uhlenbeck process $(X_t)_{t\geq 0}$ solving the SDE

$$\mathrm{d}X_s = -X_s\mathrm{d}s + \sqrt{2}\mathrm{d}B_s \ , \quad X_0 \sim \pi \ , \tag{C.11}$$

then, $(Y_t)_{t\geq 0}$ and $(\sqrt{t(t+\sigma^2)}X_{\Phi(t)})_{t\geq 0}$ have the same distribution by application of Dubins-Schwartz theorem Montanari [2023]. In other words, the localization process (4.1) can be identified as a *non-linear* time-reversal transformation of the noising VP process (C.11).

Comments on our framework. We recall that $\alpha(t) = t^{1/2}g(t)$. We now justify the assumptions given on g in Section 4.3.1.

- Rationale behind (a) and (b): together, these requirements ensure that (i) α is continuously derivable at $t = 0$, without imposing this same condition on g and that (ii) we have $\alpha(0) = 0$. With (i), the SDE defined in (4.8) does not have a singularity at time $t = 0$. With (ii), it allows to fix Y_0^α to a deterministic value. This is equivalent to full independence with X, *i.e.*, full noise looking at the expression of the SNR in (4.14), which is natural to start the denoising process. Note that taking $-g$ instead of g would be equivalent in the denoising procedure, since the SNR considers $g(t)^2$. Nonetheless, considering g with positive values is arbitrary and allows us to simplify our framework.
- Rationale behind (c): as $t \to T_{\text{gen}}$, we would like to denoise increasingly the observation process, *i.e.*, gaining progressively information about X, such that we obtain complete denoising at the end of the process. Looking at the expression of the SNR in (4.14), this requires to naturally take $g(t) \to \infty$ as $t \to T_{\text{gen}}$ and g strictly increasing, recalling that g takes non negative values.

Note that it includes the case of standard stochastic localization by taking $g(t) = t^{1/2}$. In this case, all properties are verified: in particular, consider $C = 1$ and $\beta = 1$ in (b).

C.2.2 Theoretical results.

Here, we provide details on the theoretical claims made in Section 4.3.1 on our extended framework of stochastic localization.

Link between score and denoiser function. The formula given in (4.12) is an immediate corollary of Lemma 52 applied to the observation process defined in (4.6). Indeed, the conditional distribution of Y_t^α given $X = x \in \mathbb{R}^d$ is $p_t^\alpha(\cdot|x) = \mathrm{N}(\alpha(t)x, \sigma^2 t\, \mathrm{I}_d)$. Then, it comes that $\nabla_y \log p_t^\alpha(y|x) = \{\alpha(t)x - y\}/\sigma^2 t$, and we obtain the result using Lemma 52-(C.5).

Localization rate. We begin with the localization rate given in Equation (4.7). We define the following quantity $\mathrm{LocR}(t) = \sigma\sqrt{d}/g(t)$. We emphasize that the assumption of finite second order moment on the target distribution, that is used below, guarantees the proper definition of the 2-Wasserstein distance.

Proposition 54. *Assume that π has finite second order moment. Denote by π_t^α the probability distribution of $Y_t^\alpha/\alpha(t)$ where $(Y_t^\alpha)_{t\in[0,T_{gen})}$ is the stochastic observation process defined in (4.6). Then, for any $t \in (0, T_{gen})$, we have*

$$W_2(\pi, \pi_t^\alpha) \leq \mathrm{LocR}(t)\ . \tag{C.12}$$

Proof. Let $t \in (0, T_{\mathrm{gen}})$. Consider the following coupling $(X, \hat{Y}_t^\alpha)$: (i) $X \sim \pi$, (ii) $\hat{Y}_t^\alpha = \alpha(t)X + \sigma\sqrt{t}Z$, where $Z \sim \mathrm{N}(0, \mathrm{I}_d)$. Denote $X_t^\alpha = \hat{Y}_t^\alpha/\alpha(t)$. We have $X_t^\alpha = X + \{\sigma/g(t)\}\, Z$ and $\hat{X}_t^\alpha \sim \pi_t^\alpha$. Therefore, it comes that

$$W_2(\pi, \pi_t^\alpha)^2 \leq \mathbb{E}[\|\sigma/g(t)\, Z\|^2] = \sigma^2 d/g(t)^2\ , \tag{C.13}$$

which gives the result. □

In particular, we recover the localization rate of the standard setting given in Section 4.2 with $g(t) = t^{1/2}$ and $T_{\mathrm{gen}} = \infty$. Note that this upper bound applies on general distributions that may be non log-concave, as considered in **A**3, and may be improved by assuming further regularity. We notably show a strong refinement in the Gaussian case by a factor $O(g(t))$.

Proposition 55. *Consider the target distribution given by $\pi = \mathrm{N}(\mathbf{m}, \gamma^2\, \mathrm{I}_d)$ where $\mathbf{m} \in \mathbb{R}^d$ and $\gamma > 0$. Then, as $t \to T_{gen}$, we have*

$$W_2(\pi, \pi_t^\alpha) = \gamma \left| 1 - \left(1 + \frac{\sigma^2}{\gamma^2 g(t)^2} \right)^{1/2} \right| \sqrt{d} \sim \frac{\sigma}{2\gamma g(t)}\, \mathrm{LocR}(t)\ . \tag{C.14}$$

Proof. In this setting, the distribution of the observation process at time t is tractable, and we have

$$p_t^\alpha = \mathrm{N}(\alpha(t)\mathbf{m}, \{\alpha(t)^2\gamma^2 + \sigma^2 t\}\ \mathrm{I}_d)\ . \tag{C.15}$$

Then, we get that

$$\pi_t^\alpha = \mathrm{N}(\mathbf{m}, \{\gamma^2 + \sigma^2/g(t)^2\}\, \mathrm{I}_d)\ . \tag{C.16}$$

In this particular, using (C.2), the 2-Wasserstein distance is given by

$$W_2(\pi, \pi_t^\alpha) = \gamma \left| 1 - \left(1 + \frac{\sigma^2}{\gamma^2 g(t)^2} \right)^{1/2} \right| \sqrt{d}\ , \tag{C.17}$$

from which we deduce the result by a simple asymptotic equivalent. □

Interestingly, the same bound applies on the denoiser $u_t^\alpha(Y_t^\alpha) = \int_{\mathbb{R}^d} x q_t^\alpha(x|Y_t^\alpha)$. The proof relies on the same structure.

Proposition 56. *Assume that π has finite second order moment. Denote by $\tilde{\pi}_t^\alpha$ the probability distribution of $u_t^\alpha(Y_t^\alpha)$ where $(Y_t^\alpha)_{t\in[0,T_{gen})}$ is the stochastic observation process defined in (4.6). Then, for any $t \in (0, T_{gen})$, we have*

$$W_2(\pi, \tilde{\pi}_t^\alpha) \leq \mathrm{LocR}(t)\ . \tag{C.18}$$

Proof. Let $t \in (0, T_{\mathrm{gen}})$. Consider the following coupling $(X, \hat{Y}_t^\alpha)$: (i) $X \sim \pi$, (ii) $\hat{Y}_t^\alpha = \alpha(t)X + \sigma\sqrt{t}Z$, where $Z \sim \mathrm{N}(0, \mathrm{I}_d)$. Denote $X_t^\alpha = u_t^\alpha(Y_t^\alpha)$. We have $X_t^\alpha \sim \tilde{\pi}_t^\alpha$. Therefore, it comes that

$$W_2(\pi, \tilde{\pi}_t^\alpha)^2 \leq \mathbb{E}_{X,X_t^\alpha}[\|X_t^\alpha - X\|^2] = \mathbb{E}_{X,\hat{Y}_t^\alpha}[\|\mathbb{E}[X|\hat{Y}_t^\alpha] - X\|^2] = \mathbb{E}_{X,\hat{Y}_t^\alpha}[\|\mathbb{E}[X|\hat{Y}_t^\alpha/\alpha(t)] - X\|^2]\ . \tag{C.19}$$

Since conditional expectations are orthogonal projections in L^2, we have

$$W_2(\pi, \tilde{\pi}_t^\alpha)^2 \leq \mathbb{E}_{X, \hat{Y}_t^\alpha}[\|\hat{Y}_t^\alpha/\alpha(t) - X\|^2] = \mathbb{E}[\|\sigma/g(t)\, Z\|^2] = \sigma^2 d/g(t)^2 \;, \tag{C.20}$$

which gives the result. □

Similarly, we also show a strong refinement of Proposition 56 by a factor $O(g(t))$ in the Gaussian case.

Proposition 57. *Consider the target distribution given by $\pi = \mathrm{N}(\mathbf{m}, \gamma^2 \mathrm{I}_d)$ where $\mathbf{m} \in \mathbb{R}^d$ and $\gamma > 0$. Then, as $t \to T_{gen}$, we have*

$$W_2(\pi, \tilde{\pi}_t^\alpha) = \gamma \left| 1 - \left(1 + \frac{\sigma^2}{\gamma^2 g(t)^2}\right)^{-1/2} \right| \sqrt{d} \sim \frac{\sigma}{2\gamma g(t)} \operatorname{LocR}(t) \;. \tag{C.21}$$

Proof. Here again, the distribution of the observation process at time t is tractable, we have

$$p_t^\alpha = \mathrm{N}(\alpha(t)\mathbf{m}, \{\alpha(t)^2\gamma^2 + \sigma^2 t\}\, \mathrm{I}_d) \;. \tag{C.22}$$

In particular, the score is tractable and given by

$$\nabla_y \log p_t^\alpha(y) = \frac{\alpha(t)\mathbf{m} - y}{\alpha(t)^2\gamma^2 + \sigma^2 t} \;. \tag{C.23}$$

Using relation (4.12), we thus have

$$u_t^\alpha(y) = \frac{\alpha(t)\gamma^2}{\alpha(t)^2\gamma^2 + \sigma^2 t} y + \frac{\sigma^2 t}{\alpha(t)^2\gamma^2 + \sigma^2 t} \mathbf{m} \;. \tag{C.24}$$

Recalling that $\tilde{\pi}_t^\alpha$ is the distribution of $u_t^\alpha(Y_t^\alpha)$, by combining (C.22) and (C.24), it comes that

$$\tilde{\pi}_t^\alpha = \mathrm{N}\left(\mathbf{m}, \gamma^2 \left\{1 + \tfrac{\sigma^2}{\gamma^2 g(t)^2}\right\}^{-1} \mathrm{I}_d\right) \;. \tag{C.25}$$

In this particular, using (C.2), the 2-Wasserstein distance is given by

$$W_2(\pi, \tilde{\pi}_t^\alpha) = \gamma \left| 1 - \left(1 + \frac{\sigma^2}{\gamma^2 g(t)^2}\right)^{-1/2} \right| \sqrt{d} \tag{C.26}$$

from which we deduce the result with a simple asymptotic equivalent. □

General remark on denoising in Stochastic Localization. In our presentation of standard stochastic localization where the observation process is defined in (4.1), we take Y_T/T as approximate sample from π, where Y_T is obtained by running the (discretized) SDE (4.2) up to time T. Yet, in standard stochastic localization, Montanari [2023] proposed to take the last denoiser of the procedure $u_T(Y_T) = \int_{\mathbb{R}^d} x q_T(x|Y_T)\mathrm{d}x$. We highlight here that this is equivalent asymptotically.

- The two approaches have the same upper bound on their rate of convergence, see Proposition 54 and Proposition 56 in the standard setting. Note that we also obtain the same rate asymptotically in the Gaussian case, see Proposition 55 and Proposition 57.
- When T is large, the Gaussian term dominates the target term in the expression of the posterior q_t given in (4.3). Noting that $\sigma^2/T \ll 1$, it comes that $q_T(x|Y_T) \approx \delta_{Y_T/T}$, and therefore $u_T(Y_T) \approx Y_T/T$.

As shown above, this equivalence also occurs in generalized stochastic localization. In addition to the similarity of results in Proposition 54 and Proposition 56, we observe that, as T is large if $T_{\mathrm{gen}} = \infty$ or as T is close to T_{gen} otherwise, the Gaussian term dominates the target term in the expression of the posterior q_t given in (4.10). Since $\sigma^2/g(T)^2 \ll 1$, it comes that $q_T^\alpha(x|Y_T^\alpha) \approx \delta_{Y_T^\alpha/\alpha(T)}$, and therefore, we have again $u_T^\alpha(Y_T^\alpha) \approx Y_T^\alpha/\alpha(T)$.

In the main document of our paper, we choose to consider the distribution of $Y_T^\alpha/\alpha(T)$ rather than the the distribution of the denoiser $u_T^\alpha(Y_T^\alpha)$ as an approximation of the target distribution. This choice seemed to us to be easier to understand for non-expert readers. In practice, we however align with the original methodology and compute the denoiser $u_T^\alpha(Y_T^\alpha)$ instead of $Y_T^\alpha/\alpha(T)$. This allows to us to have a fair comparison with the approach from Saremi et al. [2024], who also consider such Bayes estimator for sampling.

Markovian projection of the observation process. Following Liptser and Shiryaev [1977] and Brunick and Shreve [2013], we now turn to existence and uniqueness results on the SDE (4.8), under assumptions on π. These are corollaries of the general results: [Brunick and Shreve, 2013, Corollary 3.7], restated in Proposition 58, and [Liptser and Shiryaev, 1977, Theorem 7.6], restated in Proposition 60.

Proposition 58 (Corollary 3.7 in Brunick and Shreve [2013])**.** *Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a complete probability space, and let W be an $\mathbb{R}^r$-valued Brownian motion defined on this space. Fix a time horizon $T > 0$, and consider the $\mathbb{R}^d$-valued stochastic process $(X_t)_{t\in[0,T]}$ defined by*

$$X_t = X_0 + \int_0^t b_s \mathrm{d}s + \int_0^t \sigma_s \mathrm{d}W_s, \quad 0 \le t \le T \,,$$

where $b = (b_t)_{t\in[0,T]}$ is an $\mathbb{R}^d$-valued process adapted to a filtration with respect to which W is a Brownian motion, $\sigma = (\sigma_t)_{t\in[0,T]}$ is a $d\times r$-valued adapted process (with respect to the same filtration), and the integrability condition

$$\mathbb{E}\left[\int_0^t \left(\|b_s\| + \left\|\sigma_s\sigma_s^\top\right\|\right) \mathrm{d}s\right] < \infty, \quad \text{for all } t \in [0,T] \,,$$

is satisfied. Then there exist measurable functions $\hat{b} : [0,T) \times \mathbb{R}^d \to \mathbb{R}^d$, and $\hat{\sigma} : [0,T) \times \mathbb{R}^d \to \mathbb{R}^{d\times d}$, such that for each $t \in [0,T)$ and $x_t \in \mathbb{R}^d$,

$$\hat{b}(t,x_t) = \mathbb{E}\left[b_t \mid X_t = x_t\right], \quad \hat{\sigma}(t,x_t)\hat{\sigma}(t,x_t)^\top = \mathbb{E}\left[\sigma_t\sigma_t^\top \mid X_t = x_t\right] \,.$$

Moreover, there exists a complete probability space $(\hat{\Omega}, \hat{\mathcal{F}}, \hat{\mathbb{P}})$ supporting a continuous, adapted $\mathbb{R}^d$-valued process $(\hat{X}_t)_{t\in[0,T]}$, and a d-dimensional Brownian motion $(\hat{W}_t)_{t\in[0,T]}$, such that

$$\hat{X}_t = \hat{X}_0 + \int_0^t \hat{b}(s,\hat{X}_s)\mathrm{d}s + \int_0^t \hat{\sigma}(s,\hat{X}_s)\mathrm{d}\hat{W}_s, \quad 0 \le t \le T \,,$$

and for each $t \in [0,T]$, the law of $\hat{X}_t$ under $\hat{\mathbb{P}}$ coincides with the law of X_t under $\mathbb{P}$.

Corollary 59. *Assume that π has a finite first moment. Let $Y^\alpha = (Y_t^\alpha)_{t\in[0,T_{gen}]}$ be the observation process defined on a complete probability space $(\Omega, \mathcal{F}, \mathbb{P})$ by*

$$Y_t^\alpha = \alpha(t)X + \sigma B_t, \quad Y_0^\alpha = 0 \,, \tag{C.27}$$

where $\alpha : [0, T_{gen}] \to \mathbb{R}_+$ is a differentiable, strictly increasing function, $\sigma > 0$, $X \sim \pi$, and B is a standard Brownian motion defined on this space and independent of X.

Then, there exists a complete probability space and a process $\hat{Y}^\alpha = (\hat{Y}_t^\alpha)_{t\in[0,T_{gen}]}$ defined on it, which satisfies the stochastic differential equation

$$\mathrm{d}\hat{Y}_t^\alpha = \dot{\alpha}(t)u_t^\alpha(\hat{Y}_t^\alpha)\mathrm{d}t + \sigma\mathrm{d}B_t, \quad \hat{Y}_0^\alpha = 0 \,, \tag{C.28}$$

where $u_t^\alpha(y) = \mathbb{E}[X|Y_t^\alpha = y]$, and such that for every $t \in [0, T_{gen}]$, the marginal law of $\hat{Y}_t^\alpha$ coincides with that of Y_t^α.

Proof. By construction,

$$Y_t^\alpha = \alpha(t)X + \sigma B_t \,,$$

so Y^α is an Itô process with drift $b_t = \dot{\alpha}(t)X$ and constant diffusion σ. It satisfies

$$Y_t^\alpha = \int_0^t b_s \mathrm{d}s + \sigma B_t \,.$$

To apply Proposition 58, we verify the integrability condition

$$\begin{aligned}\mathbb{E}\left[\int_0^t \left(\|b_s\| + \sigma^2\right) \mathrm{d}s\right] &= \mathbb{E}[\|X\|] \int_0^t \dot{\alpha}(s)\mathrm{d}s + \sigma^2 t \\ &= \mathbb{E}[\|X\|](\alpha(t) - \alpha(0)) + \sigma^2 t < \infty \,,\end{aligned}$$

for all $t \in [0, T_{\text{gen}}]$, since $\mathbb{E}[\|X\|] < \infty$ by assumption. By Proposition 58, there exists a measurable function $\hat{b} : [0, T) \times \mathbb{R}^d \to \mathbb{R}^d$ such that

$$\hat{b}(t, y) = \mathbb{E}[b_t \mid Y_t^\alpha = y] = \dot{\alpha}(t)\mathbb{E}[X \mid Y_t^\alpha = y] = \dot{\alpha}(t) u_t^\alpha(y) \, .$$

Thus, there exists a process $\hat{Y}^\alpha$, defined on a possibly different probability space, satisfying the SDE

$$\mathrm{d}\hat{Y}_t^\alpha = \dot{\alpha}(t) u_t^\alpha(\hat{Y}_t^\alpha)\mathrm{d}t + \sigma \mathrm{d}\hat{B}_t, \quad \hat{Y}_0^\alpha = 0 \, ,$$

whose law at each time t matches that of Y_t^α. □

Proposition 60 (Theorem 7.6 & Remark 7.2.7. in Liptser and Shiryaev [1977])**.** *Let $T > 0$, $\sigma > 0$. Consider the following SDE*

$$\mathrm{d}Y_t = h_t(Y_t)\mathrm{d}t + \sigma \mathrm{d}W_t, \quad Y_0 = 0 \, , \tag{C.29}$$

where $\mathbb{P}(\int_0^T \|h_t(Y_t)\|^2 \, \mathrm{d}t < \infty) = 1$. Then, (C.29) admits at most one weak solution.

The proof of this result lies in the fact that the distributions of solutions to the SDE (C.29) have the same density with respect to the distribution of the Brownian motion. We refer to Liptser and Shiryaev [1977] for the complete proof.

Corollary 61. *Assume that π has finite second order moment and that $t \to \dot{\alpha}(t)^2$ is integrable at time $t = 0$. Then, the SDE defined in (4.8) by*

$$\mathrm{d}Y_t^\alpha = \dot{\alpha}(t) u_t^\alpha(Y_t^\alpha)\mathrm{d}t + \sigma \mathrm{d}B_t \, , \; Y_0^\alpha = 0 \, , \tag{C.30}$$

has a unique weak solution.

Proof. Consider $T \in (0, T_{\text{gen}})$. Since π has its first order moment that is finite, we have existence of solutions by Corollary 59. Consider a solution Y^α and denote $h_t = \dot{\alpha}(t) u_t^\alpha$. We have $\|h_t(Y_t^\alpha)\|^2 = \dot{\alpha}(t)^2 \|\mathbb{E}[X|Y_t^\alpha]\|^2 \leq \dot{\alpha}(t)^2 \mathbb{E}[\|X\|^2 \, |Y_t^\alpha]$ by Jensen's inequality. Then, using this inequality, we obtain that

$$\mathbb{E}[\textstyle\int_0^T \|h_t(Y_t^\alpha)\|^2 \, \mathrm{d}t] = \int_0^T \mathbb{E}[\|h_t(Y_t^\alpha)\|^2]\mathrm{d}t \leq \int_0^T \dot{\alpha}(t)^2 \mathbb{E}[\mathbb{E}[\|X\|^2 \, |Y_t^\alpha]]\mathrm{d}t = \mathbb{E}[\|X\|^2] \int_0^T \dot{\alpha}(t)^2 \mathrm{d}t$$

Combining the assumptions on π and α, it comes that $\mathbb{E}[\int_0^T \|h_t(Y_t^\alpha)\|^2 \, \mathrm{d}t] < \infty$ using the inequality above, and therefore $\mathbb{P}(\int_0^T \|h_t(Y_t^\alpha)\|^2 \, \mathrm{d}t < \infty) = 1$. We finally obtain the result by applying Proposition 60. □

We emphasize that the extra assumption made on α in Corollary 61 is verified for the localization schemes Geom and Geom-∞ presented in Section 4.3.2.

C.2.3 Alternative denoising approach to SLIPS

We note here the score of the observation process can be expressed via an expectation over the posterior of the model, in a different manner than (4.12). Define $v_t^\alpha(y) = \int_{\mathbb{R}^d} \nabla_x \log \pi(x) q_t^\alpha(x|y)\mathrm{d}x$, where q_t^α is the posterior density given in (4.10).

Lemma 62. *Consider the observation process defined in (4.6) with marginal distribution at time t given by p_t^α. Assume that $\log \pi$ is continuously differentiable on $\mathbb{R}^d$ and that there exists $\varphi : \mathbb{R}^d \to \mathbb{R}_+$ such that $\int_{\mathbb{R}^d} \varphi(z)\mathrm{N}(z; 0, \sigma^2 t \, \mathrm{I}_d)\mathrm{d}z < \infty$ and for any $(z, y) \in \mathbb{R}^d \times \mathbb{R}^d$, we have $\|\nabla_y \pi(\{z + y\}/\alpha(t))\| \leq \varphi(z)$. Then, we have for any $y \in \mathbb{R}^d$*

$$\nabla_y \log p_t^\alpha(y) = v_t^\alpha(y)/\alpha(t) \, . \tag{C.31}$$

Proof. We recall that $p_t^\alpha(y) = \int_{\mathbb{R}^d} \mathrm{N}(y; \alpha(t)x, \sigma^2 t \, \mathrm{I}_d)\mathrm{d}\pi(x)$. Then, by change of variable $z = \alpha(t)x - y$, we have

$$p_t^\alpha(y) \propto \int_{\mathbb{R}^d} \pi(\{z + y\}/\alpha(t))\mathrm{N}(z; 0, \sigma^2 t \, \mathrm{I}_d)\mathrm{d}z \, , \tag{C.32}$$

where the multiplicative constant does not depend on y. For any $y \in \mathbb{R}^d$, denote by $\tilde{q}_t^\alpha(\cdot|y)$ the density defined up to a normalizing constant by $\tilde{q}_t^\alpha(z|y) \propto \pi(\{z+y\}/\alpha(t))\mathrm{N}(z; 0, \sigma^2 t\,\mathrm{I}_d)$. By combining the assumptions of the lemma with the result of Lemma 52-(C.5) for this new expression of p_t^α, we obtain that

$$\begin{aligned}\nabla_y \log p_t^\alpha(y) &= \int_{\mathbb{R}^d} \nabla_y \log \pi(\{z+y\}/\alpha(t))\tilde{q}_t^\alpha(z|y)\mathrm{d}z \\ &= \frac{1}{\alpha(t)} \int_{\mathbb{R}^d} \nabla \log \pi(\{z+y\}/\alpha(t))\tilde{q}_t^\alpha(z|y)\mathrm{d}z \\ &= \frac{1}{\alpha(t)} \int_{\mathbb{R}^d} \nabla_x \log \pi(x) q_t^\alpha(x|y)\mathrm{d}x \ ,\end{aligned}$$

where we re-applied the change of variable $z = \alpha(t)x - y$ in the last equality. □

Therefore, the SDE (4.8) is strictly equivalent to the SDE

$$\mathrm{d}Y_t^\alpha = \tfrac{\dot{\alpha}(t)}{\alpha(t)}\{Y_t^\alpha + \tfrac{\sigma^2 t}{\alpha(t)} v_t^\alpha(Y_t^\alpha)\}\mathrm{d}t + \sigma \mathrm{d}B_t \ , \ Y_0^\alpha = 0 \ . \tag{C.33}$$

Hence, to simulate from the observation process in a learning-free fashion, we can adopt a similar strategy to SLIPS, that rather involves the SDE derived above. Given a SNR-adapted discretization $(t_k)_{k=1}^K$ of a time interval $[t_0, T]$, where $t_0 > 0$ and $K \geq 1$, it amounts to consider a sequence $\{(\tilde{Y}_{t_k}^\alpha, V_{t_k}^\alpha)\}_{k=1}^K$, where $V_{t_k}^\alpha$ is a Monte Carlo estimator of $v_{t_k}^\alpha(\tilde{Y}_{t_k}^\alpha)$ and $\tilde{Y}_{t_k}^\alpha$ is obtained by solving the SDE

$$\mathrm{d}\tilde{Y}_t^\alpha = \tfrac{\dot{\alpha}(t)}{\alpha(t)}\{\tilde{Y}_t^\alpha + \tfrac{\sigma^2 t_k}{\alpha(t_k)} V_{t_k}\}\mathrm{d}t + \sigma \mathrm{d}B_t \ , t \in [t_k, t_{k+1}] \ . \tag{C.34}$$

In theory, this approach leads to the same level difficulty in the Monte Carlo estimation as SLIPS, since the involved random poster densities are the same. In particular, this formulation does not bypass the "duality of log-concavity" explained in Section 4.4.3. The main difference with the approach presented in the main of our paper however lies in the integration of the SDE (C.34), which is not tractable at first sight. By combining the result of Lemma 53 and the use of the stochastic Exponential Integrator scheme Durmus and Moulines [2015], we show in Appendix C.4.1 how to solve this SDE for our localization schemes. We emphasize that this implementation is also available in our code.

Unfortunately, we found in our early experiments that this approach suffered from numerical unstability and showed higher variance than SLIPS. We think that this is due to the evaluation of $\nabla \log \pi$ in our MC estimation for the early steps of the algorithm, and could be overcome by using the SLIPS recursion given in (4.18) instead. We leave this study for future work.

C.3 Detailed computations for Gaussian mixtures

In this section, we consider the special case where π is a mixture of N Gaussians with weights $(w_i)_{i=1}^N$, means $(\mathbf{m}_i)_{i=1}^N$ and covariance matrices $(\gamma_i^2\,\mathrm{I}_d)_{i=1}^N$. Under this assumption, p_t^α can be explicitly written for any $y \in \mathbb{R}^d$ and $t \in (0, T_{\mathrm{gen}})$ as

$$p_t^\alpha(y) = \sum_{i=1}^N w_i \mathrm{N}(y; \alpha(t)\mathbf{m}_i, t(g^2(t)\gamma_i^2 + \sigma^2)\,\mathrm{I}_d) \ . \tag{C.35}$$

This means that the distribution of the observation process is itself a mixture of Gaussians with the same weights as π but with means $(\alpha(t)\mathbf{m}_i)_{i=1}^N$ and covariance matrices $(t(g^2(t)\gamma_i^2 + \sigma^2)\,\mathrm{I}_d)_{i=1}^N$. This elementary result is obtained by applying the rule of linear combination of independent Gaussian random variables. The score of the observation process can also be computed by noticing that for all $y \in \mathbb{R}^d$

$$\nabla \log p_t^\alpha(y) = \frac{\nabla \log p_t^\alpha(y)}{p_t^\alpha(y)} = -\frac{\sum_{i=1}^N w_i t^{-1}(g^2(t)\gamma_i^2 + \sigma^2)^{-1}(y - \alpha(t)\mathbf{m}_i)\mathrm{N}(y; \alpha(t)\mathbf{m}_i, t(g^2(t)\gamma_i^2 + \sigma^2)\,\mathrm{I}_d)}{\sum_{i=1}^N w_i \mathrm{N}(y; \alpha(t)\mathbf{m}_i(g^2(t)\gamma_i^2 + \sigma^2)\,\mathrm{I}_d)} \ .$$

One can also compute the posterior distribution q_t^α for any $t \in (0, T_{\text{gen}})$ and $x, y \in \mathbb{R}^d$ as

$$\begin{aligned} q_t^\alpha(x|y) &\propto \sum_{i=1}^N w_i \mathrm{N}(x; \mathbf{m}_i, \gamma_i^2 \mathrm{I}_d) \mathrm{N}(y; \sqrt{t} g(t) x, \sigma^2 t \mathrm{I}_d) \\ &= \sum_{i=1}^N w_i (\sqrt{t} g(t))^{-d} \mathrm{N}(x; \mathbf{m}_i, \gamma_i^2 \mathrm{I}_d) \mathrm{N}\left(x; \frac{y}{\sqrt{t} g(t)}, \frac{\sigma^2}{g^2(t)} \mathrm{I}_d\right) \\ &= \sum_{i=1}^N \underbrace{w_i \alpha(t)^{-d} \mathrm{N}\left(\mathbf{m}_i; \frac{y}{\alpha(t)}, \left(\gamma_i^2 + \frac{\sigma^2}{g^2(t)}\right) \mathrm{I}_d\right)}_{=\tilde{w}_{t,y,i}^\alpha} \\ &\quad \times \mathrm{N}\left(x; \underbrace{\left(\frac{\gamma_i^2 \sigma^2}{\sigma^2 + g^2(t) \gamma_i^2}\right)\left(\frac{\mathbf{m}_i}{\gamma_i^2} + \frac{y g(t)}{\sqrt{t} \sigma^2}\right)}_{=\mathbf{m}_{t,y,i}^\alpha}, \underbrace{\left(\frac{\gamma_i^2 \sigma^2}{\sigma^2 + g^2(t) \gamma_i^2}\right)}_{=(\gamma_{t,y,i}^\alpha)^2} \mathrm{I}_d\right) . \end{aligned}$$

This shows that the posterior is itself a mixture of Gaussian distributions with weights $(w_{t,y,i}^\alpha)_{i=1}^N$ where $w_{t,y,i}^\alpha = \tilde{w}_{t,y,i}^\alpha / \sum_{j=1}^N \tilde{w}_{t,y,j}^\alpha$, means $(\mathbf{m}_{t,y,i}^\alpha)_{i=1}^N$ and covariance matrices $((\gamma_{t,y,i}^\alpha)^2 \mathrm{I}_d)_{i=1}^N$.

Additionally, we can derive tight expressions for the constants R and τ introduced in **A**3 in the case where π is a Gaussian mixture parameterized by $N = 2$, $w_1 = 1 - w_2 = w$, with $w \in (0, 1)$, $\mathbf{m}_2 = -\mathbf{m}_1 = a \mathbf{1}_d$[1], with $a > 0$, and $\gamma_1 = \gamma_2 = \gamma$, with $\gamma > 0$. In this case, π verifies **A**3, where (i) μ is a mixture of two Dirac masses at $-a \mathbf{1}_d$ and $+a \mathbf{1}_d$ with respective weights w and $1 - w$ and (ii) $\tau = \gamma$. Moreover, for any random vector $U \sim \mu$, it holds

$$\begin{aligned} \mathbb{E}[\|U - \mathbf{m}_\pi\|^2] &= \mathbb{E}[\|U - a(1 - 2w) \mathbf{1}_d\|^2] && \text{(C.36)} \\ &= w \|-a \mathbf{1}_d - a(1 - 2w) \mathbf{1}_d\|^2 + (a - w) \|a \mathbf{1}_d - a(1 - 2w) \mathbf{1}_d\|^2 && \text{(C.37)} \\ &= 4w(1 - w) a^2 d && \text{(C.38)} \end{aligned}$$

Therefore, we obtain that $R = 2\sqrt{w(1 - w)} a$ in **A**3, where the inequality holds tightly. Note also that $\bar{\pi}$, defined as the distribution of $X - \mathbf{m}_\pi$ where $X \sim \pi$, is still a Gaussian mixture that verifies **A**3 with same constants R and τ.

In the experiments conducted in Section 4.6, Appendix C.4 and Appendix C.6.2, we will consider a rolling example given by the target distribution $\bar{\pi}$ where $w = 1/3$, $a = 1.0$ and $\gamma^2 = 0.05$. In this case, we have $R^2 = 8/9$ and $\tau^2 = 0.05$, and the corresponding density is defined as

$$x \in \mathbb{R}^d \mapsto \tfrac{2}{3} \mathrm{N}\left(x; -\tfrac{2}{3} \mathbf{1}_d, 0.05 \mathrm{I}_d\right) + \tfrac{1}{3} \mathrm{N}\left(x; \tfrac{4}{3} \mathbf{1}_d, 0.05 \mathrm{I}_d\right) . \quad \text{(C.39)}$$

Due to the tightness of R, the scalar variance of this target distribution verifies $R_\pi^2 = d(R^2 + \tau^2)$.

C.4 Sampling via Stochastic Localization in an ideal setting

The goal of this section is to validate the claims from Section 4.4.1 about the minimization of the integration error. In this section, we work under the assumption that the score $\nabla \log p_t^\alpha$ is a known function and do not consider any MCMC method at all (even for the initialization). This setting removes entirely the estimation error that we deal with in Section 4.4.2 to solely focus on the integration error. For the initialization, we now consider an arbitrary $t_0 > 0$ with $Y_{t_0}^\alpha$ distributed as $\mathrm{N}(0, \sigma^2 t_0 \mathrm{I}_d)$ (the best estimation that we have at the beginning of the SDE). We will clarify 4 different points in this section.

(a) Exploring *Exponential Integration* (EI) based schemes for SDE discretization;

(b) Analyzing the impact of the SNR-adapted discretization on the integration error;

(c) Exploring the limits of the Gaussian approximation $Y_{t_0}^\alpha \sim \mathrm{N}(0, \sigma^2 t_0 \mathrm{I}_d)$;

(d) Exploring the impact of the computational budget.

[1] We recall that $\mathbf{1}_d$ stands for the d-dimensional vector with all components equal to 1.

In the numerical examples presented below, we consider the target distribution defined in (C.39) where $d = 10$. We compute the exact score using the analytical formulas from Appendix C.3. We choose to display two complementary results based on (i) the empirical Sliced Wasserstein distance Bonneel et al. [2015], Nadjahi et al. [2019], which tells how local information on π is recovered, and (ii) the error in estimating the weight of the first mode, which tells about global properties of the estimation. Moreover, for any SL scheme α, the values of t_0 and T will be taken so that the log-SNR evaluated at times t_0 and t_K has the same value (the initial log-SNR is taken as -4.0 and the last log-SNR is taken as 5.0). This enables fair comparison across different schedules.

C.4.1 Exploring new integration schemes

By exploiting the relation given in (4.12) between the denoiser function and the score of the observation process, which is now assumed to be tractable, the SDE (4.8) is equivalently defined on $[t_0, T]$ by

$$\mathrm{d}Y_t^\alpha = \tfrac{\dot\alpha(t)}{\alpha(t)}\{Y_t^\alpha + \sigma^2 t \nabla \log p_t^\alpha(Y_t^\alpha)\}\mathrm{d}t + \sigma \mathrm{d}B_t \ , \ Y_0^\alpha = 0 \ . \tag{C.40}$$

Through this formulation, we have fully removed the problem of score estimation, but the issue of discretization error still remains to sample from $(Y_t^\alpha)_{t\in[t_0,T]}$ with (C.40). Due to the divergence of the coefficient $t \mapsto \dot\alpha(t)/\alpha(t)$ at time 0 (and time T_{gen} in the finite-time setting), we propose to use the stochastic *Exponential Integrator* (EI) scheme Durmus and Moulines [2015] in this setting. Consider a time discretization of the interval $[t_0, T]$ defined by an increasing sequence of timesteps $(t_k)_{k=0}^K$ where $t_K = T$ and $K \geq 1$. Then, the EI scheme applied on the SDE (C.40) amounts to define a sequence of random variables $\{\tilde Y_{t_k}^\alpha\}_{k=0}^K$ obtained by *exactly* integrating the SDE defined for any $k \in \{0, \dots, K-1\}$ by

$$\mathrm{d}\tilde Y_t^\alpha = \tfrac{\dot\alpha(t)}{\alpha(t)}\{\tilde Y_t^\alpha + \sigma^2 t_k \nabla \log p_{t_k}^\alpha(\tilde Y_{t_k}^\alpha)\}\mathrm{d}t + \sigma \mathrm{d}B_t \ , \ t \in [t_k, t_{k+1}] \ . \tag{C.41}$$

Although the exact integration is not guaranteed for a general schedule α as defined in Section 4.3.1, our specific design of the denoising schedule $g(t) = t^{-1/2}\alpha(t)$ in the schemes Geom and Geom-∞ provides tractable computations following the result of Lemma 53. We treat these cases separately below.

EI scheme combined with Geom-∞ localization scheme. In this setting, we recall that $g(t) = t^{\alpha_1/2}$. Then, we have

$$\frac{\dot\alpha(t)}{\alpha(t)} = \frac{\alpha_1 + 1}{2t} \ . \tag{C.42}$$

Note that $\dot\alpha(t)/\alpha(t) \to 0$ as $t \to 0$. Following Lemma 53, the sequence $\{\tilde Y_{t_k}^\alpha\}_{k=0}^K$ obtained by the EI scheme is defined by the recursion

$$\tilde Y_{t_{k+1}}^\alpha = \left(\frac{t_{k+1}}{t_k}\right)^{\frac{\alpha_1+1}{2}} \tilde Y_{t_k}^\alpha + \left\{\left(\frac{t_{k+1}}{t_k}\right)^{\frac{\alpha_1+1}{2}} - 1\right\} \sigma^2 t_k \nabla \log p_{t_k}^\alpha(\tilde Y_{t_k}^\alpha) + \sigma \sqrt{\frac{t_{k+1}}{\alpha_1 t_k^{\alpha_1}}(t_{k+1}^{\alpha_1} - t_k^{\alpha_1})} Z_{k+1} \ .$$

where $(Z_k)_{k=1}^K$ is distributed according to the standard centered Gaussian distribution. In the *standard* case, *i.e.*, $\alpha_1 = 1$, this simplifies as

$$\tilde Y_{t_{k+1}}^\alpha = \tfrac{t_{k+1}}{t_k}\tilde Y_{t_k}^\alpha + \{\tfrac{t_{k+1}}{t_k} - 1\}\sigma^2 t_k \nabla \log p_{t_k}^\alpha(\tilde Y_{t_k}^\alpha) + \sigma \sqrt{\tfrac{t_{k+1}}{t_k}(t_{k+1} - t_k)} Z_{k+1} \ . \tag{C.43}$$

EI scheme combined with Geom localization scheme. In this second setting, we recall that $g(t) = t^{\alpha_1/2}(1-t)^{-\alpha_2/2}$ ($T_{\mathrm{gen}} = 1$). Then, we have

$$\frac{\dot\alpha(t)}{\alpha(t)} = \frac{\alpha_1 + 1}{2t} + \frac{\alpha_2}{2(1-t)} \ . \tag{C.44}$$

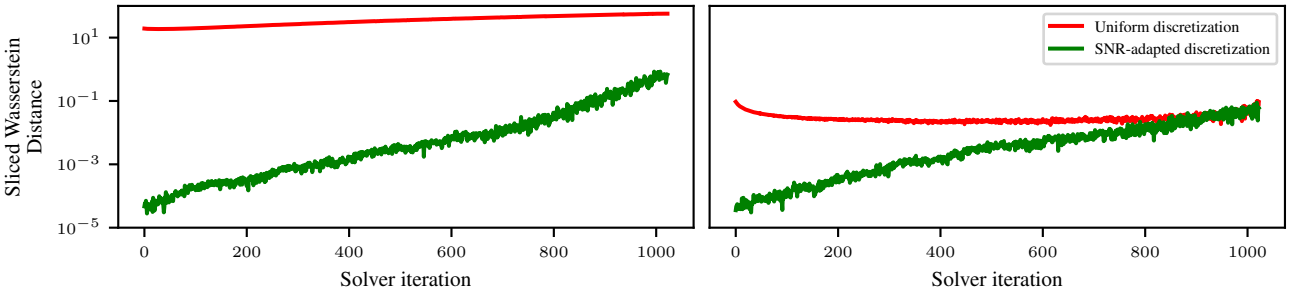


Figure C.1: **Sliced Wasserstein distance between the true samples from $p^{\alpha}_{t_k}$ and the empirical distribution of $\tilde{Y}^{\alpha}_{t_k}$** obtained by EI scheme combined with uniform time discretization (green) and SNR-adapted discretization (red) for the Standard (**Left**) and Geom(1,1) (**Right**) schemes.

Note that $\dot{\alpha}(t)/\alpha(t) \to 0$ as $t \to 0$ and $t \to 1$. Following Lemma 53, the sequence $\{\tilde{Y}^{\alpha}_{t_k}\}_{k=0}^{K}$ obtained by the EI scheme is defined by the recursion

$$\begin{aligned}\tilde{Y}^{\alpha}_{t_{k+1}} &= \left(\frac{t_{k+1}}{t_k}\right)^{\frac{\alpha_1+1}{2}} \left(\frac{1-t_k}{1-t_{k+1}}\right)^{\frac{\alpha_2}{2}} \tilde{Y}^{\alpha}_{t_k} \\ &+ \left\{\left(\frac{t_{k+1}}{t_k}\right)^{\frac{\alpha_1+1}{2}} \left(\frac{1-t_k}{1-t_{k+1}}\right)^{\frac{\alpha_2}{2}} - 1\right\} \sigma^2 t_k \nabla \log p^{\alpha}_{t_k}(\tilde{Y}^{\alpha}_{t_k}) \\ &+ \frac{\sigma}{\sqrt{\alpha_1}} \frac{t_{k+1}^{\frac{\alpha_1+1}{2}}}{(1-t_{k+1})^{\frac{\alpha_2}{2}}} \sqrt{{}_2\mathrm{F}_1(-\alpha_1,-\alpha_2,1-\alpha_1,t_k) t_k^{-\alpha_1} - {}_2\mathrm{F}_1(-\alpha_1,-\alpha_2,1-\alpha_1,t_{k+1}) t_{k+1}^{-\alpha_1}}\, Z_{k+1} \,,\end{aligned}$$

where ${}_2\mathrm{F}_1$ denotes the hypergeometric function, see [Olver, 2010, Chapter 15], and $(Z_k)_{k=1}^{K}$ is distributed according to the standard centered Gaussian distribution.

When $(\alpha_1, \alpha_2) = (1, 1)$, this simplifies as

$$\begin{aligned}\tilde{Y}^{\alpha}_{t_{k+1}} &= (\tfrac{t_{k+1}}{t_k})(\tfrac{1-t_k}{1-t_{k+1}})^{\frac{1}{2}} \tilde{Y}^{\alpha}_{t_k} \\ &+ \{(\tfrac{t_{k+1}}{t_k})(\tfrac{1-t_k}{1-t_{k+1}})^{\frac{1}{2}} - 1\} \sigma^2 t_k \nabla \log p^{\alpha}_{t_k}(\tilde{Y}^{\alpha}_{t_k}) \\ &+ \sigma \sqrt{\tfrac{t_{k+1}^2}{1-t_{k+1}} \log(\tfrac{t_k}{t_{k+1}}) + \tfrac{t_{k+1}(t_{k+1}-t_k)}{t_k(1-t_{k+1})}}\, Z_{k+1} \,.\end{aligned} \tag{C.45}$$

When $(\alpha_1, \alpha_2) = (2, 1)$, this simplifies as

$$\begin{aligned}\tilde{Y}^{\alpha}_{t_{k+1}} &= (\tfrac{t_{k+1}}{t_k})^{\frac{3}{2}}(\tfrac{1-t_k}{1-t_{k+1}})^{\frac{1}{2}} \tilde{Y}^{\alpha}_{t_k} \\ &+ \{(\tfrac{t_{k+1}}{t_k})^{\frac{3}{2}}(\tfrac{1-t_k}{1-t_{k+1}})^{\frac{1}{2}} - 1\} \sigma^2 t_k \nabla \log p^{\alpha}_{t_k}(\tilde{Y}^{\alpha}_{t_k}) \\ &+ \sigma t_{k+1} (\tfrac{t_{k+1}/t_k - 1}{1-t_{k+1}})^{\frac{1}{2}} \sqrt{\tfrac{t_k + t_{k+1}}{2 t_k t_{k+1}} - 1} Z_{k+1} \,.\end{aligned} \tag{C.46}$$

When $(\alpha_1, \alpha_2) = (1, 2)$, this simplifies as

$$\begin{aligned}\tilde{Y}^{\alpha}_{t_{k+1}} &= (\tfrac{t_{k+1}}{t_k})^{\frac{1}{2}}(\tfrac{1-t_k}{1-t_{k+1}})^{\frac{3}{2}} \tilde{Y}^{\alpha}_{t_k} \\ &+ \{(\tfrac{t_{k+1}}{t_k})^{\frac{1}{2}}(\tfrac{1-t_k}{1-t_{k+1}})^{\frac{3}{2}} - 1\} \sigma^2 t_k \nabla \log p^{\alpha}_{t_k}(\tilde{Y}^{\alpha}_{t_k}) \\ &+ \tfrac{\sigma}{1-t_{k+1}} \sqrt{(t_{k+1}-t_k)\{t_{k+1}^2 + \tfrac{t_{k+1}}{t_k}\} + 2t_{k+1}^2 \log(\tfrac{t_k}{t_{k+1}})\}}\, Z_{k+1} \,.\end{aligned} \tag{C.47}$$

C.4.2 Studying the impact of the SNR-adapted time discretization

In Section 4.4.1, we suggested to take a SNR-adapted time discretization by $(t_k)_{k=0}^{K}$ such that

$$\log \mathrm{SNR}(t_k) = \log \mathrm{SNR}(t_0) + \Delta_{\mathrm{SNR}} k \,, \tag{C.48}$$

with $t_0 > 0$ and $\Delta_{\mathrm{SNR}} = (\log \mathrm{SNR}(T) - \log \mathrm{SNR}(t_0))/K$. Figure C.1 shows that the SNR-adapted discretization efficiently reduces the integration error for both schedules Geom and Geom-∞. Note, the impact on the schedule Geom(1,1) is more moderated as the uniform initialization already splits the curve into moderated log-SNR increments (see Figure 4.2) because of a slower rate near t_0.

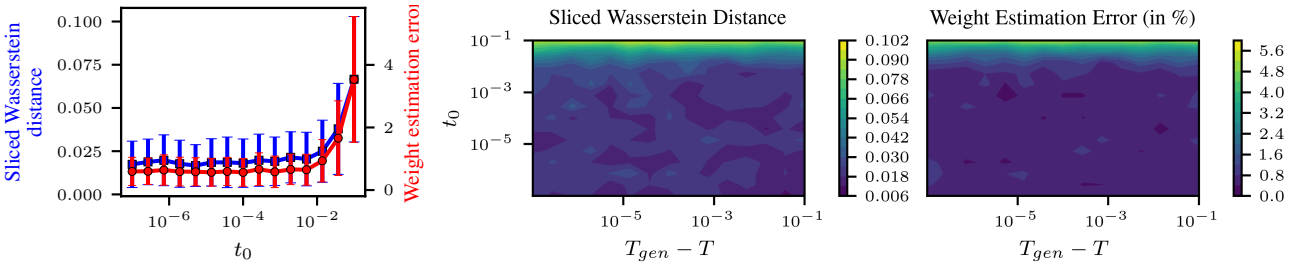


Figure C.2: **Impact of t_0 with different schedules measured with the relative weight estimation error and the sliced Wasserstein distance. Left**: Impact of t_0 in the standard scheme. **Right**: Impact of t_0 and $(T_{\text{gen}} - T)$ in the Geom(1,1) scheme.

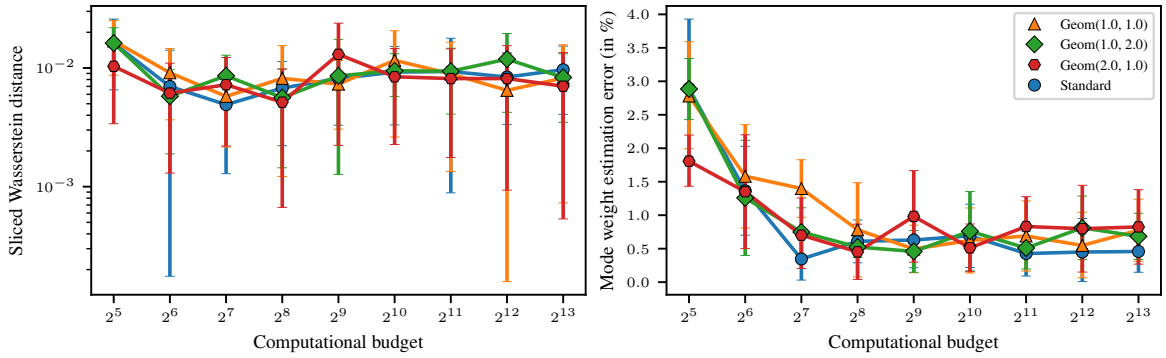


Figure C.3: **Impact of the computational budget K with different schemes.** The sampling error is computed either with the Sliced Wasserstein distance (*left*) or the relative weight error (*right*).

C.4.3 Studying the impact of the Gaussian approximation in the initialization

In this section only, we replace the target distribution considered in (C.39) by the mixture of two Gaussian distributions $\mathrm{N}(-3\,\mathbf{1}_{10}, \Sigma)$ and $\mathrm{N}(3\,\mathbf{1}_{10}, \Sigma)$, where $\Sigma = 0.05\,\mathrm{I}_{10}$, with weights respectively given by $2/3$ and $1/3$. Therefore, π has non-zero mean and high variance, which provides a challenging setting for our Gaussian approximation at initialization given by $Y^{\alpha}_{t_0} \sim \mathrm{N}(0, \sigma^2 t_0\,\mathrm{I}_d)$. Without information on the target, this is the best estimation that we have when t_0 is close to 0. Figure C.2 shows that only high values of t_0 (*i.e.* above 10^{-2}) degrades the performance in our localization schemes. This underlines the need of running the Langevin-within-Langevin correction procedure of this approximation as explained in Section 4.4.3. Additionally, note that Figure C.2 (*right*) shows that taking $(T_{\text{gen}} - T)$ low in the finite time setting does not degrade performance, which was not obvious from the log-SNR shape near T_{gen} in Figure 4.1.

C.4.4 Studying the impact of the number of discretization steps

The bottom row of Figure C.3 shows that, in the ideal case where the score is known, using the SNR-adapted discretization makes all the different schemes equivalent[2]. Moreover, we see that the sampling error quickly stabilizes with medium budgets which highlights that the integration error was successfully minimized.

C.5 Theoretical results on the duality of log-concavity

In this section, we detail the results on log-concavity provided in Section 4.4.2 and Section 4.4.3. For sake of readability, those are stated with a general $\sigma > 0$. Our first result provides uniform upper bounds on the Hessian of the log-density of the observation process and its corresponding log-posterior density.

Lemma 63. *Assume **A**3. Let $t \in (0, T_{gen})$. We recall that p^{α}_t stands for the marginal distribution at time t of the observation process defined by (4.6), while q^{α}_t stands for the corresponding posterior density. Under*

[2]We recall that we choose t_0 and T so that the starting and ending levels of SNR are the same across different schemes.

regularity assumptions on π detailed in the proof of the lemma, we have for any $(x, y) \in \mathbb{R}^d \times \mathbb{R}^d$ that

$$\nabla_y^2 \log p_t^\alpha(y) \preccurlyeq \zeta_p(t)\, \mathrm{I}_d \;, \quad \text{where } \zeta_p(t) = \tfrac{\alpha(t)^2 dR^2}{(\alpha(t)^2\tau^2+\sigma^2 t)^2} - \tfrac{1}{\alpha(t)^2\tau^2+\sigma^2 t} \;, \tag{C.49}$$

$$\nabla_x^2 \log q_t^\alpha(x|y) \preccurlyeq \zeta_q(t)\, \mathrm{I}_d \;, \quad \text{where } \zeta_q(t) = \tfrac{dR^2}{\tau^4} - \tfrac{1}{\tau^2} - \tfrac{g(t)^2}{\sigma^2} \;. \tag{C.50}$$

In particular, ζ_q is a strictly decreasing function on $(0, T_{gen})$.

Proof. We begin by proving (C.49). Consider the stochastic process $(Y_t^\alpha)_{t\in[0,T_{\mathrm{gen}})}$, given in (4.6), and defined by $Y_t^\alpha = \alpha(t)X + \sigma W_t$ where $X \sim \pi$ and $(W_t)_{t\geq 0}$ is a standard Brownian motion. Due to **A**3, X can be written as $X = U + G$ where $U \sim \mu$, with $\mathbb{E}[\|U - \mathbf{m}_\pi\|^2] \leq dR^2$, and $G \sim \mathrm{N}(0, \tau^2\, \mathrm{I}_d)$. We thus have the following decomposition

$$p_t^\alpha(y) = \textstyle\int_{\mathbb{R}^d} p_t^\alpha(y|u)\mathrm{d}\mu(u) \;, \tag{C.51}$$

where $p_t^\alpha(\cdot|u)$ is the conditional density of Y_t^α given $U = u \in \mathbb{R}^d$ defined as

$$p_t^\alpha(y|u) = \mathrm{N}(y; \alpha(t)u, \{\alpha(t)^2\tau^2 + \sigma^2 t\}\, \mathrm{I}_d) \;. \tag{C.52}$$

Then, it comes that

$$\nabla_y \log p_t^\alpha(y|u) = \frac{-y + \alpha(t)u}{\alpha(t)^2\tau^2 + \sigma^2 t} \;, \; \nabla_y^2 \log p_t^\alpha(y|u) = -\frac{1}{\alpha(t)^2\tau^2 + \sigma^2 t}\mathrm{I}_d \;. \tag{C.53}$$

We now make the technical assumption that for any $k \in \{1, 2\}$, there exists $\varphi_{p,k} : \mathbb{R}^d \to \mathbb{R}_+$ such that $\int_{\mathbb{R}^d} \varphi_{p,k}(u)\mathrm{d}\mu(u) < \infty$ and for any $(u, y) \in \mathbb{R}^d \times \mathbb{R}^d$, we have $\left\|\nabla_y^k p_t^\alpha(y|u)\right\| \leq \varphi_{p,k}(u)$. By combining this assumption with the result of Lemma 52-(C.6), we obtain that

$$\nabla_y^2 \log p_t^\alpha(Y_t^\alpha) = -\frac{1}{\alpha(t)^2\tau^2 + \sigma^2 t}\mathrm{I}_d + \mathrm{Cov}\left[\frac{-Y_t^\alpha + \alpha(t)U}{\alpha(t)^2\tau^2 + \sigma^2 t}\middle| Y_t^\alpha\right] \tag{C.54}$$

$$= -\frac{1}{\alpha(t)^2\tau^2 + \sigma^2 t}\mathrm{I}_d + \frac{\alpha(t)^2}{(\alpha(t)^2\tau^2 + \sigma^2 t)^2}\mathrm{Cov}[U|Y_t^\alpha] \;. \tag{C.55}$$

Since $\mathbb{E}[\|U - \mathbf{m}_\pi\|^2] \leq dR^2$, we have by Cauchy-Schwartz inequality that $\mathrm{Cov}[U|Y_t^\alpha] \preccurlyeq dR^2\, \mathrm{I}_d$[3]. Then, we obtain the bound (C.49) as $\nabla^2 \log p_t^\alpha$ is continuous and p_t^α is positive on $\mathbb{R}^d$.

We now prove (C.50). We recall from (4.10) that $q_t^\alpha(x|y) \propto \pi(x)\mathrm{N}(x; y/\alpha(t), \sigma^2/g(t)^2\, \mathrm{I}_d)$. Therefore

$$\nabla_x^2 \log q_t^\alpha(X|Y_t^\alpha) = \nabla_x^2 \log \pi(X) - \frac{g(t)^2}{\sigma^2}\mathrm{I}_d \;. \tag{C.56}$$

In particular, we have $\pi(x) = \int_{\mathbb{R}^d} \mathrm{N}(x; u, \tau^2\, \mathrm{I}_d)\mathrm{d}\mu(u)$.

We now make the technical assumption that for any $k \in \{1, 2\}$, there exists $\varphi_{q,k} : \mathbb{R}^d \to \mathbb{R}_+$ such that $\int_{\mathbb{R}^d} \varphi_{q,k}(u)\mathrm{d}\mu(u) < \infty$ and for any $(u, x) \in \mathbb{R}^d \times \mathbb{R}^d$, we have $\left\|\nabla_x^k \mathrm{N}(x; u, \tau^2\, \mathrm{I}_d)\right\| \leq \varphi_{p,k}(u)$. Then, by using again the result of Lemma 52-(C.6), we obtain that

$$\nabla_x^2 \log \pi(X) = -\frac{1}{\tau^2}\mathrm{I}_d + \frac{1}{\tau^4}\mathrm{Cov}[U|X] \;, \tag{C.57}$$

Since $\mathbb{E}[\|U - \mathbf{m}_\pi\|^2] \leq dR^2$, we also have by Cauchy-Schwartz inequality that $\mathrm{Cov}[U|X] \preccurlyeq dR^2\, \mathrm{I}_d$[4]. We obtain (C.50) with similar reasoning as before. □

Note that the upper bounds obtained in Lemma 63 can be made tighter in the case where the target distribution is a Gaussian mixture, as shown in Lemma 64.

[3]Note that this upper bound may be loose, since it does not depend on Y_t^α.
[4]Note that this upper bound may be loose, since it does not depend on X.

Lemma 64. *Let $a > 0$, $\gamma > 0$ and $w \in (0,1)$. Consider the target distribution π defined as the mixture of the Gaussian distributions $\mathrm{N}(-a\,\mathbf{1}_d, \gamma^2\,\mathrm{I}_d)$ and $\mathrm{N}(+a\,\mathbf{1}_d, \gamma^2\,\mathrm{I}_d)$, with respective weights w and $1-w$. Then for any $(x,y) \in \mathbb{R}^d \times \mathbb{R}^d$, we have*

$$\nabla_y^2 \log p_t^\alpha(y) = \tfrac{2\alpha(t)^2 a^2}{(\alpha(t)^2\gamma^2+\sigma^2 t)^2(1+\cosh g(t,y))}\,\mathbf{1}_d\mathbf{1}_d^\top - \tfrac{1}{\alpha(t)^2\gamma^2+\sigma^2 t}\,\mathrm{I}_d\,, \tag{C.58}$$

$$\text{with } g(t,y) = \frac{2\alpha(t) a y^\top \mathbf{1}_d}{\alpha(t)^2\gamma^2+\sigma^2 t} + \log(\frac{1}{w}-1)$$

$$\nabla_x^2 \log q_t^\alpha(x|y) = \tfrac{2a^2}{\gamma^4(1+\cosh f(t,x))}\,\mathbf{1}_d\mathbf{1}_d^\top - \tfrac{1}{\gamma^2}\,\mathrm{I}_d - \tfrac{g(t)^2}{\sigma^2}\,\mathrm{I}_d\,, \tag{C.59}$$

$$\text{with } f(t,x) = \frac{2a x^\top \mathbf{1}_d}{\gamma^2} + \log(\frac{1}{w}-1)$$

The proof of this result is inspired from the derivation of [Saremi et al., 2024, Eq. (4.6)].

Proof. As explained in Appendix C.3, π verifies **A**3, where $\tau = \gamma$ and μ is a mixture of two Dirac masses at $-a\,\mathbf{1}_d$ and $a\,\mathbf{1}_d$ with respective weights w and $1-w$, whose density is given by $p(u) = w\mathbb{1}_{-a\,\mathbf{1}_d}(u) + (1-w)\mathbb{1}_{a\,\mathbf{1}_d}(u)$.

We recall that the stochastic process $(Y_t^\alpha)_{t\in[0,T_{\mathrm{gen}})}$, given in (4.6), is defined by $Y_t^\alpha = \alpha(t)X + \sigma W_t$ where $X \sim \pi$ and $(W_t)_{t\geq 0}$ is a standard Brownian motion. Due to **A**3, X can be written as $X = U + G$ where $U \sim \mu$ and $G \sim \mathrm{N}(0, \gamma^2\,\mathrm{I}_d)$.

We begin by proving (C.58). Following the first part of the proof of Lemma 63, we have

$$\nabla_y^2 \log p_t^\alpha(y) = -\frac{1}{\alpha(t)^2\gamma^2+\sigma^2 t}\mathrm{I}_d + \frac{\alpha(t)^2}{(\alpha(t)^2\gamma^2+\sigma^2 t)^2}\mathrm{Cov}[U_y]\,, \tag{C.60}$$

where U_y is a random vector distributed according to $p_t^\alpha(\cdot|y)$, the conditional distribution of U given $Y_t^\alpha = y \in \mathbb{R}^d$, whose density is given by

$$p_t^\alpha(u|y) \propto p(u) p_t^\alpha(y|u)\,, \tag{C.61}$$

where $p_t^\alpha(y|u)$ is defined in (C.52). Therefore, we obtain that

$$p_t^\alpha(u|y) \propto \underbrace{w \exp\left(-\frac{\|y+\alpha(t)a\mathbf{1}_d\|^2}{2\{\alpha(t)^2\gamma^2+\sigma^2 t\}}\right)}_{w_1(t,y)} \mathbf{1}_{-a\,\mathbf{1}_d}(u) + \underbrace{(1-w)\exp\left(-\frac{\|y-\alpha(t)a\mathbf{1}_d\|^2}{2\{\alpha(t)^2\gamma^2+\sigma^2 t\}}\right)}_{w_2(t,y)} \mathbf{1}_{a\,\mathbf{1}_d}(u)\,. \tag{C.62}$$

Hence, $p_t^\alpha(\cdot|y)$ is a mixture of two Dirac masses at $-a\,\mathbf{1}_d$ and $a\,\mathbf{1}_d$ with respective weights $\tilde{w}_1(t,y) = w_1(t,y)/\{w_1(t,y)+w_2(t,y)\}$ and $\tilde{w}_2(t,y) = w_2(t,y)/\{w_1(t,y)+w_2(t,y)\}$. Therefore, we get that

$$\mathrm{Cov}[U_y] = \mathbb{E}[U_y U_y^\top] - \mathbb{E}[U_y]\mathbb{E}[U_y]^\top \tag{C.63}$$

$$= \tilde{w}_1(t,y)a^2\mathbf{1}_d\mathbf{1}_d^\top + \tilde{w}_2(t,y)a^2\mathbf{1}_d\mathbf{1}_d^\top - (\tilde{w}_2(t,y)-\tilde{w}_1(t,y))^2 a^2\mathbf{1}_d\mathbf{1}_d^\top \tag{C.64}$$

$$= \{1-(\tilde{w}_2(t,y)-\tilde{w}_1(t,y))^2\}a^2\mathbf{1}_d\mathbf{1}_d^\top\,. \tag{C.65}$$

In particular, we have

$$\tilde{w}_2(t,y) - \tilde{w}_1(t,y) = \frac{w_2(t,y)-w_1(t,y)}{w_2(t,y)+w_1(t,y)} = \frac{\exp(\log w_2(t,y)) - \exp(\log w_1(t,y))}{\exp(\log w_2(t,y)) - \exp(\log w_1(t,y))}\,, \tag{C.66}$$

where $\log w_1(t,y) = -\frac{\|y+\alpha(t)a\mathbf{1}_d\|^2}{2\{\alpha(t)^2\gamma^2+\sigma^2 t\}} + \log(w)$ and $\log w_2(t,y) = -\frac{\|y-\alpha(t)a\mathbf{1}_d\|^2}{2\{\alpha(t)^2\gamma^2+\sigma^2 t\}} + \log(1-w)$. Then, it comes that

$$1-(\tilde{w}_2(t,y)-\tilde{w}_1(t,y))^2 = 1 - \tanh\left(\frac{\log w_2(t,y) - \log w_1(t,y)}{2}\right) \tag{C.67}$$

$$= \frac{2}{1+\cosh(\log w_2(t,y) - \log w_1(t,y))}\,. \tag{C.68}$$

Moreover, we get that

$$\log w_2(t,y) - \log w_1(t,y) = \frac{2\alpha(t)ay^\top\mathbf{1}_d}{\alpha(t)^2\gamma^2+\sigma^2 t} + \log\left(\frac{1}{w}-1\right)\,. \tag{C.69}$$

Denote $g(t,y) = \log w_2(t,y) - \log w_1(t,y)$. We finally obtain (C.58) by combining previous computations.

We now prove (C.59). Following the second part of the proof of Lemma 63, we have

$$\nabla_x^2 \log q_t^\alpha(x|y) = -\frac{1}{\gamma^2}\mathrm{I}_d - \frac{g(t)^2}{\gamma^2}\mathrm{I}_d + \frac{1}{\gamma^4}\mathrm{Cov}[U_x]\;, \tag{C.70}$$

where U_x is a random vector distributed according to $\mu(\cdot|x)$, the conditional distribution of U given $X = x \in \mathbb{R}^d$, whose density is given by

$$\mu(u|x) \propto \mathrm{N}(x; u, \gamma^2\,\mathrm{I}_d)\mu(u)\;. \tag{C.71}$$

Therefore, we obtain that

$$\mu(u|x) \propto \underbrace{w \exp\left(-\frac{\|x + a\mathbf{1}_d\|^2}{2\gamma^2}\right)}_{w_1(t,x)} \mathbf{1}_{-a\,\mathbf{1}_d}(u) + \underbrace{(1-w)\exp\left(-\frac{\|x - a\mathbf{1}_d\|^2}{2\gamma^2}\right)}_{w_2(t,x)} \mathbf{1}_{a\,\mathbf{1}_d}(u)\;. \tag{C.72}$$

Hence, $\mu(\cdot|x)$ is a mixture of two Dirac masses at $-a\,\mathbf{1}_d$ and $a\,\mathbf{1}_d$ with respective weights $\tilde{w}_1(t,x) = w_1(t,x)/\{w_1(t,x)+w_2(t,x)\}$ and $\tilde{w}_2(t,x) = w_2(t,x)/\{w_1(t,x)+w_2(t,x)\}$. Therefore, we get that

$$\begin{aligned}
\mathrm{Cov}[U_x] &= \mathbb{E}[U_x U_x^\top] - \mathbb{E}[U_x]\mathbb{E}[U_x]^\top && \text{(C.73)}\\
&= \tilde{w}_1(t,x)a^2\mathbf{1}_d\mathbf{1}_d^\top + \tilde{w}_2(t,x)a^2\mathbf{1}_d\mathbf{1}_d^\top - (\tilde{w}_2(t,x) - \tilde{w}_1(t,x))^2 a^2\mathbf{1}_d\mathbf{1}_d^\top && \text{(C.74)}\\
&= \{1 - (\tilde{w}_2(t,x) - \tilde{w}_1(t,x))^2\}a^2\mathbf{1}_d\mathbf{1}_d^\top\;. && \text{(C.75)}
\end{aligned}$$

Similarly to the computations derived above, we obtain that

$$1 - (\tilde{w}_2(t,x) - \tilde{w}_1(t,x))^2 = \frac{2}{1+\cosh f(t,x)}\;, \tag{C.76}$$

where

$$f(t,x) = \log w_2(t,x) - \log w_1(t,x) = \frac{2ax^\top\mathbf{1}_d}{\gamma^2} + \log\left(\frac{1}{w} - 1\right)\;, \tag{C.77}$$

which finally leads to (C.59).

□

We provide below a result, that combines formal versions of Theorem 4 and Theorem 5.

Proposition 65. *Assume **A**3. Denote by g^{-1} the inverse function of the denoising schedule g.*

If $dR^2 > \tau^2$, define

$$t_{\mathrm{p}} = g^{-1}\left(\tfrac{\sigma}{\sqrt{dR^2-\tau^2}}\right) \quad \text{and} \quad t_{\mathrm{q}} = g^{-1}\left(\tfrac{\sigma\sqrt{dR^2-\tau^2}}{\tau^2}\right)\;, \tag{C.78}$$

otherwise, define $t_{\mathrm{p}} = T_{gen}$ and $t_{\mathrm{q}} = 0$. Then,

(a) for any $y \in \mathbb{R}^d$, $q_t^\alpha(\cdot|y)$ is strongly log-concave for $t \in (t_{\mathrm{q}}, T_{gen})$ and gets more log-concave as t increases,

(b) p_t^α is strongly log-concave for $t \in (0, t_{\mathrm{p}})$.

Proof. Consider the expressions of t_{p} and t_{q} given above. Note that they are well defined in the case where $dR^2 > \tau^2$ since g is a bijection from $[0, T_{\mathrm{gen}})$ to $\mathbb{R}_+$. We begin with the proof of the result (a). Let $t \in (0, T_{\mathrm{gen}})$. Following Lemma 63-(C.50), for any $y \in \mathbb{R}^d$, $q_t^\alpha(\cdot|y)$ is strongly log concave if

$$\zeta_q(t) < 0 \iff g(t)^2 > \frac{\sigma^2(dR^2-\tau^2)}{\tau^4} \iff t > t_{\mathrm{q}}\;. \tag{C.79}$$

Moreover, q_t^α gets more log-concave as t increases as ζ_q is a strictly decreasing function, see Lemma 63. We now give the proof of the result (b). Let $t \in (0, T_{\mathrm{gen}})$. Following Lemma 63-(C.49), p_t^α is strongly log concave if

$$\zeta_p(t) < 0 \iff \frac{\alpha(t)^2 dR^2}{\alpha(t)^2\tau^2 + \sigma^2 t} < 1 \iff g(t)^2(dR^2 - \tau^2) < \sigma^2 \iff t < t_{\mathrm{p}}\;. \tag{C.80}$$

□

Hence, this result shows that the condition $dR^2 > \tau^2$ is restrictive on the log-concavity of the marginal distribution and the posterior of the localization model. In terms of Gaussian mixtures, this condition can be interpreted as having the distance between the modes that is larger than the variance of the modes. We now restate Theorem 6 and give its proof.

Proposition 66. *Assume* ***A****3, where* $dR^2 < 2\tau^2$*. Then,* $t_{\mathrm{q}} < t_{\mathrm{p}}$*, where* t_{q} *and* t_{p} *are defined in Proposition 65.*

Proof. If $dR^2 \leq \tau^2$, this result is directly obtained since $t_{\mathrm{q}} = 0$ and $t_{\mathrm{p}} = T_{\mathrm{gen}}$. Assume that $dR^2 > \tau^2$. We have

$$t_{\mathrm{q}} < t_{\mathrm{p}} \iff \frac{dR^2 - \tau^2}{\tau^4} < \frac{1}{dR^2 - \tau^2} \iff (dR^2 - \tau^2)^2 < \tau^4 \iff dR^2 < 2\tau^2 \,, \tag{C.81}$$

which gives the result. □

Hence, Proposition 66 shows that a sweet spot for the hyper-parameter t_0 in `SLIPS` exists under a restrictive condition on R and τ in Assumption **A**3, enabling the duality of log-concavity.

C.6 Details on `SLIPS` algorithm

C.6.1 Details on the implementation of `SLIPS`

Langevin-within-Langevin initialization. Consider a timestep t_0 well chosen such that $p^\alpha_{t_0}$ and $q^\alpha_{t_0}$ are both approximately log-concave. The goal of our initialization is to sample from $p^\alpha_{t_0}$ at lowest cost. Thanks to log-concavity of $p^\alpha_{t_0}$, we may consider to apply ULA to sample from this distribution. Given $N \geq 1$ and a step-size $\lambda > 0$, this amounts to consider a sequence of random variables $\{Y^{(n)}\}_{n=0}^N$, defined by the following recursion

$$Y^{(n+1)} = Y^{(n)} + \lambda \nabla \log p^\alpha_{t_0}(Y^{(n)}) + \sqrt{2\lambda} Z^{(n+1)} \,, \tag{C.82}$$

where $\{Z^{(n)}\}_{n=1}^N$ are independently distributed according to the standard centered Gaussian distribution. We recall that the score $\nabla \log p^\alpha_{t_0}$ is however not tractable but can be re-expressed with (4.12) such that for any $y \in \mathbb{R}^d$, we have

$$\nabla \log p^\alpha_{t_0}(y) = \{\alpha(t_0) u^\alpha_{t_0}(y) - y\}/\sigma^2 t_0 \,, \tag{C.83}$$

where $u^\alpha_{t_0}(y)$ is the expectation of the posterior $q^\alpha_{t_0}(\cdot|y)$ given in (4.10). Although this term is intractable too, it can be estimated by approximately sampling from $q^\alpha_{t_0}$ via MALA, since $q^\alpha_{t_0}$ is also ensured to be approximately log-concave.

Building upon this relation, at each step $n \in \{0, \dots, N-1\}$, we propose to approximate $\nabla \log p^\alpha_{t_0}(Y^{(n)})$ in the recursion (C.82) with a MCMC-based estimator of $u^\alpha_{t_0}(Y^{(n)})$ obtained by sampling from $q^\alpha_{t_0}(\cdot|Y^{(n)})$. This results in the Langevin-within-Langevin procedure summarized in Algorithm 2. We finally highlight three main choices in the design of this algorithm: (i) we choose the Gaussian approximation $Y^{(0)} \sim \mathrm{N}(0, \sigma^2 t_0 \, \mathrm{I}_d)$ for ULA initialization, relying on the study conducted in Appendix C.4.3; (ii) consequently, we set the step-size λ to be slightly smaller than the variance of this distribution, *i.e.*, $\lambda = \sigma^2 t_0/2$; (iii) the MALA-based posterior sampling is initialized with the best estimation of $u^\alpha_{t_0}(Y^{(n)})$ that we have, *i.e.*, $Y^{(n)}/\alpha(t_0)$, recalling that $Y^\alpha_t/\alpha(t)$ and $u^\alpha_t(Y^\alpha_t)$ have the same localization behaviour, see Appendix C.2.2.

Algorithmic technicalities. We now present three algorithmic subtleties featured in `SLIPS`.

(a) The step-size of MALA used to sample from the posterior is adapted. Using the acceptance ratio of the Metropolis-Hasting filter, we geometrically decrease (respectively increase) the step-size when the acceptance ratio is below (respectively above) a target ratio of 75%;

(b) The last MCMC samples obtained when sampling the posterior at step k will be the first samples when sampling at step $k+1$. This persistent trick is motivated by the fact that the posterior is expected to change little from one iteration to the next. This behaviour is notably illustrated in Figure 4.3 *(bottom row)*.

(c) Lastly, as mentioned and justified in Appendix C.2.2, we use the estimated denoiser $\tilde{U}^\alpha_T$ at the final integration timestep T as an approximate sample from π rather than $Y^\alpha_T/\alpha(T)$ to align with related work.

C.6.2 Ablation study

In this section, we investigate the impact of the different hyper-parameters of SLIPS. Similarly to Appendix C.4, we run our study by applying our algorithm on the unbalanced bimodal Gaussian mixture defined in (C.39) with $d = 10$. This section will clarify three points:

(a) The behaviour of SLIPS within the assumptions of Theorem 6;

(b) The existence of a "sweet spot" for t_0 outside the assumptions of Theorem 6 and the impact of the estimation of R_π;

(c) The tuning of hyper-parameters in the Langevin-within-Langevin initialization (see Algorithm 2).

Behaviour of SLIPS within the restrictive assumptions. In this part only, we consider the case of a mixture of two Gaussian distributions in dimension 5 given by $\mathrm{N}(-(2/3)a\,\mathbf{1}_5, \tau^2 \mathrm{I}_5)$ and $\mathrm{N}(+(4/3)a\,\mathbf{1}_5, \tau^2 \mathrm{I}_5)$, where $\tau^2 = 0.1$ and $a > 0$ is *not fixed*, with respective weights $2/3$ and $1/3$. By varying a, we vary R, recalling that $R = 2a\sqrt{2}/3$, see Appendix C.3. In Figure C.4, we run SLIPS with different schemes while setting t_0 as the mean value between t_p and t_q derived in Proposition 65, in accordance with Theorem 6. When p_t^α and q_t^α are both log-concave independently of t, *i.e.*, when $dR^2/\tau^2 < 1$ (see Proposition 65), we always set $t_0 = 10^{-2}$. This choice ensures that the Gaussian approximation which corresponds to the initialization of the Langevin-within-Langevin algorithm is accurate (see Figure C.2 for details). Figure C.4 shows that choosing t_0 according to Proposition 65 leads to accurate sampling below the threshold and degraded (or equal) performance above this threshold.

Existence of a sweet spot for duality outside of the restrictive assumptions. Given the target distribution described in (C.39) with $d = 10$, we have that $R^2 = 8/9$ and $\tau^2 = 0.05$, which means that $R^2 d/\tau^2 \approx 178$. Hence, the distribution at stake does not fit the additional assumption of Theorem 6. However, Figure C.5 shows that there still exists a sweet spot for t_0 in this case. Moreover, we observe in Figure C.6 that a poor estimation of R_π shifts the sweet spot. This makes sense as the sweet spot likely corresponds to a SNR level. Recalling that $\log \mathrm{SNR}(t) = 2\log(g(t)) + \log(R_\pi^2/(\sigma^2 d))$, if $\sigma > R_\pi/\sqrt{d}$ (*i.e.*, we overestimate R_π) the log-SNR is shifted downwards and the resulting t_0 should be larger (as we see in the right column of Figure C.6); if $\sigma < R_\pi/\sqrt{d}$ (*i.e.*, we underestimate R_π), the resulting t_0 should be smaller (as we see in the left column of Figure C.6).

Choice of hyper-parameters in the Langevin-within-Langevin initialization. Figure C.7 shows that only a few steps in the Langevin-within-Langevin algorithm (see Algorithm 2) are needed to reach a stationary sampling error.

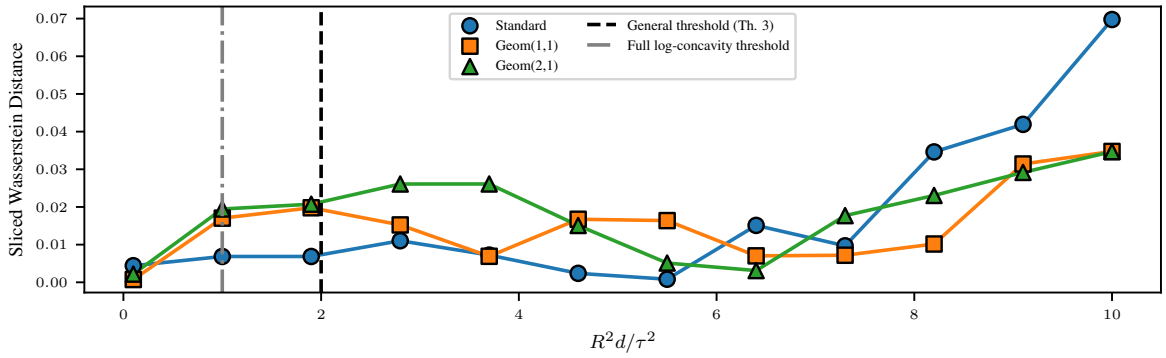


Figure C.4: **Sliced Wasserstein distance depending on the ratio $R^2 d/\tau^2$. Left**: $R^2 d/\tau^2 = 2$ is the threshold of duality of log-concavity in Theorem 6. In both cases, the grey line corresponds to the threshold where the log-concavity conditions start to be restrictive.

C.7 Details on related work

In this appendix, we provide further details on related works, highlighting similarities and identifying the main limitations.

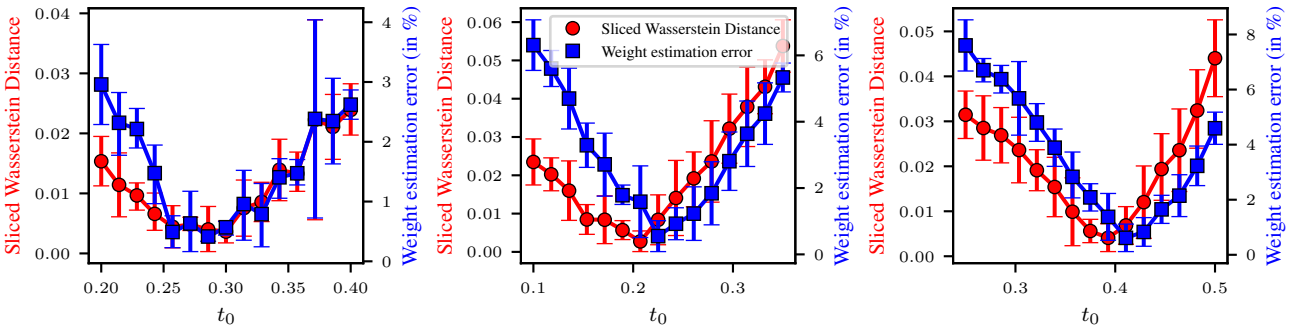


Figure C.5: **Sliced Wasserstein distance and estimation error on the relative weight depending on t_0 with different schemes. Left**: Standard. **Middle**: Geom(1,1). **Right**: Geom(2,1).

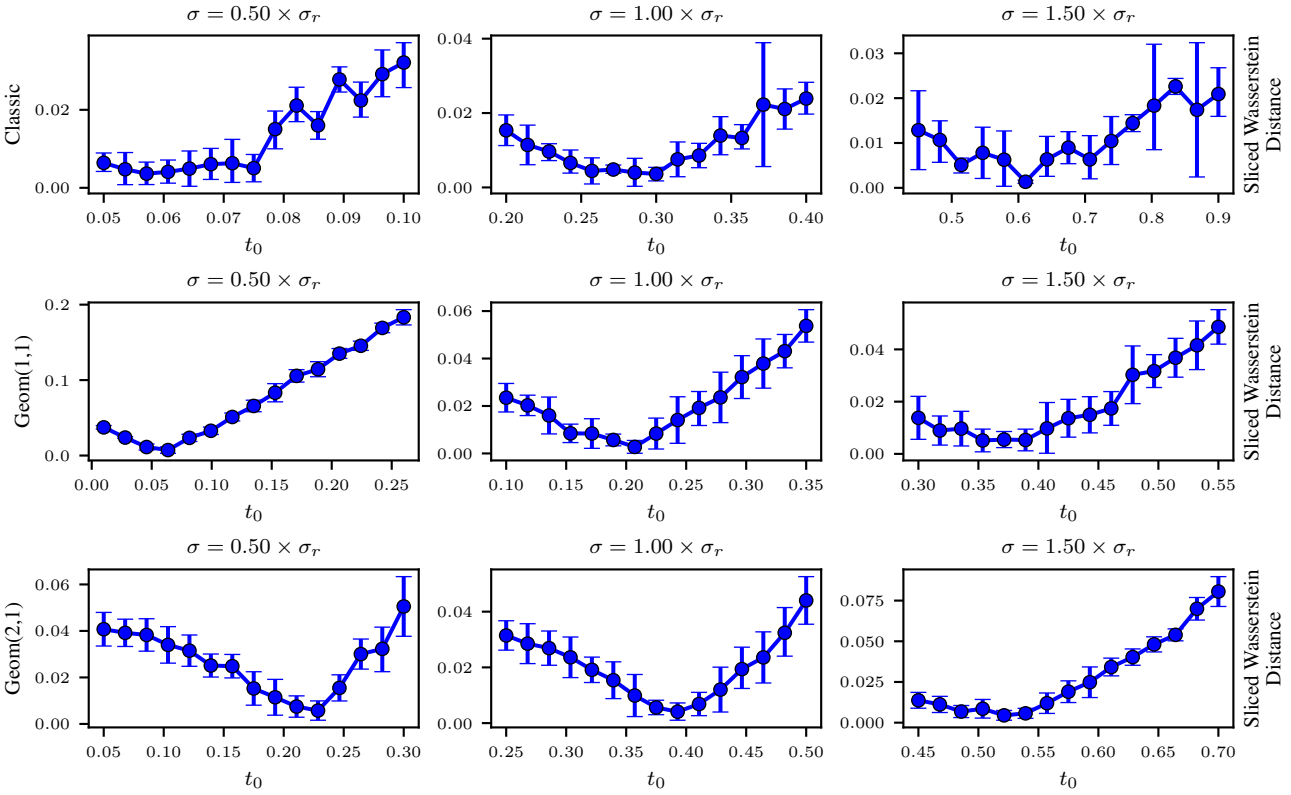


Figure C.6: **Sliced Wasserstein distance depending on t_0 for different values of σ.** We denote $\sigma_r = R_\pi/\sqrt{d}$.

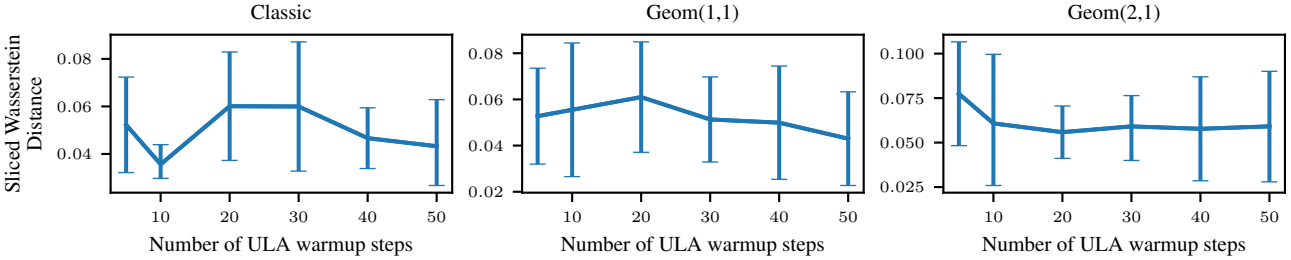


Figure C.7: **Sliced Wasserstein distance depending on the number of Langevin-within-Langevin steps.**

C.7.1 Reverse Diffusion Monte Carlo Huang et al. [2024a]

Given $T > 0$, the authors consider the denoising process $(Y_t)_{t\in[0,T]}$ that is solution to the following SDE

$$\mathrm{d}Y_t = \{Y_t + 2\nabla \log p_{T-t}(Y_t)\} + \sqrt{2}\mathrm{d}B_t \ , \ Y_0 \sim p_T \ , \tag{C.84}$$

where $(B_t)_{t\geq 0}$ is a Brownian motion in $\mathbb{R}^d$ and p_s is the *intractable* marginal distribution defined for any $s > 0$ and any $y \in \mathbb{R}^d$ by $p_s(y) = \int_{\mathbb{R}^d} \mathrm{N}(y; \mathrm{e}^{-s}x, (1-\mathrm{e}^{-2s})\,\mathrm{I}_d)\mathrm{d}\pi(x)$. Under mild conditions on π Cattiaux et al. [2023], $(Y_t)_{t\in[0,T]}$ is the time-reversed process of the standard Ornstein-Uhlenbeck process defined in (C.11) and corresponds to a Variance-Preserving diffusion model Song et al. [2021b]. For any $t \in [0,T]$, given $X \sim \pi$, we therefore have

$$Y_t = \mathrm{e}^{-(T-t)}X + \sqrt{1-\mathrm{e}^{-2(T-t)}}Z \ , \tag{C.85}$$

where Z is distributed according to the standard centered Gaussian distribution. In this case, in a similar fashion to the SL posterior given in (4.10), the conditional density of X given $Y_t = y \in \mathbb{R}^d$ can be defined as

$$q_t(x|y) \propto \pi(x)\mathrm{N}(x; \mathrm{e}^{T-t}y, (\mathrm{e}^{2(T-t)}-1)\,\mathrm{I}_d), \ t \in [0,T) \ , \tag{C.86}$$

and the denoiser function as $u_t(y) = \int_{\mathbb{R}^d} x q_t(x|y)\mathrm{d}x$. Similarly to our framework, the random vector $u_t(Y_t)$ can be seen as the denoiser of Y_t. Following Tweedie's formula given in Lemma 52, the denoiser function is related to the score of p_{T-t} for any $t \in [0,T)$ by

$$\nabla \log p_{T-t}(y) = -\frac{y}{1-\mathrm{e}^{-2(T-t)}} + \frac{\mathrm{e}^{-(T-t)}}{1-\mathrm{e}^{-2(T-t)}}u_t(y) \ . \tag{C.87}$$

Therefore, the SDE (C.84) describing the denoising process is strictly equivalent to the following SDE

$$\mathrm{d}Y_t = \left\{\frac{\mathrm{e}^{-2(T-t)}+1}{\mathrm{e}^{-2(T-t)}-1}Y_t + \frac{2\mathrm{e}^{-(T-t)}}{1-\mathrm{e}^{-2(T-t)}}u_t(Y_t)\right\}\mathrm{d}t + \sqrt{2}\mathrm{d}B_t, \ Y_0 \sim p_T \ . \tag{C.88}$$

This last SDE has to be compared with the SDE describing our observation process in (4.8): in both cases, the drift involves the denoiser function, which is not tractable in practice. Here, the target distribution is approximated by the distribution of Y_T, while it is approximated by the distribution of $Y_T^\alpha/\alpha(T)$ in our framework.

The approach of Huang et al. [2024a] to sample from the SDE (C.88), while handling the estimation of the denoiser u_t, is close to ours. Indeed, after discretizing this SDE with a certain time discretization $(t_k)_{k=1}^K$ of $[0,T]$, they propose to estimate the denoiser at time t_k with a Markov Chain Monte Carlo estimation. To do so, at each step t_k, they approximately sample from the random posterior $q_{t_k}(\cdot|\tilde{Y}_{t_k})$ with ULA (while we use MALA), where $\tilde{Y}_{t_k}$ is the k-th realization of their discretized process. Besides this, they also propose a Langevin-within-Langevin initialization to approximately sample from p_T, that involves the corresponding posterior q_0. Although they briefly discuss the difficulty of this initialization by considering the log-Sobolev constants of p_T and q_0, their analysis lacks readability and does not emphasize any crucial trade-off on T (which corresponds to our t_0).

Their scheme can nonetheless be analyzed as a localization scheme under the scope of what we call "duality of log-concavity", see Section 4.4.3. Assume that π verifies **A**3 and is not log-concave. Since we have $q_0(x|y) \propto \pi(x)\mathrm{N}(x; \mathrm{e}^T y, (\mathrm{e}^{2T}-1)\,\mathrm{I}_d)$, one can show that

(a) if T is large: q_0 will be close to π (not log-concave), while p_T will be close to $\mathrm{N}(0, \mathrm{I}_d)$ (log-concave),

(b) if T is small: q_0 will localize as a Dirac mass (log-concave), while p_T will be close to π (not log-concave).

Therefore, the setting of T is crucial in RDMC to ensure a proper initialization via Langevin-within-Langevin algorithm. However, the authors claim that setting T large would only incur computational waste and that RDMC shows insensitivity to T, aside from the case where T is too close to 0[5]. As explained above, these claims are not realistic, since sampling from the posterior q_0 becomes as hard as sampling from π when T is large. We illustrate this fundamental limitation in Figure C.8.

In contrast, we pay a particular attention to point out the importance of our hyper-parameter t_0 throughout our analysis, with practical and theoretical perspectives. Finally, we highlight that RDMC relies on a uniform time discretization (while we propose a SNR-adapted discretization), and do not use persistent initialization of their Markov chains in posterior sampling.

[5] See [Huang et al., 2024a, Appendix F.4].

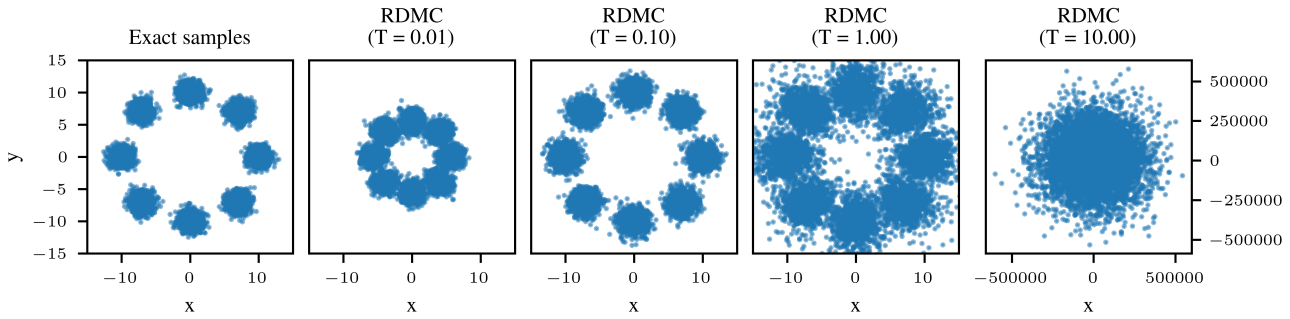


Figure C.8: **Sampling the 8-Gaussians distribution via RDMC with different values of T.**

C.7.2 Multi-Noise Measurements sampling Saremi and Srivastava [2022], Saremi et al. [2023, 2024]

Given $M \geq 1$, the Multi-Noise Measurements (MNM) model introduced by Saremi and Srivastava [2022] defines a sequence of random variables $(Y^m)_{m=1}^M$ as

$$Y^m = X + \sigma Z^m \,, \tag{C.89}$$

where $X \sim \pi$, $\sigma > 0$ and $(Z^m)_{m=1}^M$ are independently distributed according to the standard centered Gaussian distribution. This *non-Markovian* stochastic process can be seen as a denoising process. The rationale behind this formulation is that simulating an increasing number of *equally* noised measurements of a sample from π helps to obtain more information on this sample, and finally approximate it. We now explain how it can be interpreted as a non-Markovian analog to the standard localization scheme given in (4.1).

For any $m \in \{1, \dots, M\}$, the conditional density of X given the m-tuple $Y^{1:m} = y_{1:m} \in (\mathbb{R}^d)^m$ is defined as

$$q_m(x|y_{1:m}) \propto \pi(x)\mathrm{N}(x; \bar{y}_{1:m}, \sigma^2/m\, \mathrm{I}_d) \,, \tag{C.90}$$

where $\bar{y}_{1:m}$ denotes the empirical mean of the noised measurements, *i.e.*, $\bar{y}_{1:m} = (1/m)\sum_{i=1}^m y_i$. By defining $u_m(y_{1:m}) = \int_{\mathbb{R}^d} x q_m(x|y_{1:m})\mathrm{d}x$, one can derive the Bayes estimator of X given $Y^{1:m}$ as the random vector $u_m(Y^{1:m})$ Saremi and Srivastava [2022]. Interestingly, this denoiser shares the same rate of convergence as the denoiser of standard stochastic localization given in Proposition 56 when $t = m$. Indeed, [Saremi et al., 2024, Proposition 2] states that for any target distribution π, we have $W_2(\pi, \tilde{\pi}_m) \leq \sigma\sqrt{d/m}$, where $\tilde{\pi}_m$ denotes the distribution of $u_m(Y^{1:m})$. In other words, the MNM denoiser localizes to X with the same localization rate as the standard SL denoiser, defined in Section 4.2, by taking $T = M$.

To sample from π, the MNM approach consists in first sampling $Y^{1:M}$, and then computing the denoiser $u_M(Y^{1:M})$, in the same fashion as in the SL framework. Recently, Saremi et al. [2024] proposed to tackle the sampling of the M-tuple $Y^{1:M}$ by first simulating Y^1 and then sequentially sampling Y^m given $Y^{1:m-1}$ for $m \in \{2, \dots, M\}$ using a Monte Carlo Markov Chain method - in this case, Underdamped Langevin Algorithm Sachs et al. [2017]. At step m, the Langevin procedure then involves pointwise evaluations of the conditional score of the distribution of Y^m given $Y^{1:m-1}$ (or simply the score of the distribution of Y_1 when $m = 1$). This introduces the *Once-At-a-Time* (OAT) algorithm. The interest of such method lies in the fact that, under Assumption **A**3 that may include multi-modal distributions, the distributions to sample from are increasingly log-concave with m, see [Saremi et al., 2024, Theorem 1]. Hence, the most challenging step of this approach lies in the sampling of Y^1, in the same spirit as in `SLIPS`.

In their work, Saremi et al. [2024] mainly consider the case where the scores are analytically available. To handle realistic settings, they implement an IS estimator, see [Saremi et al., 2024, Section 4.2.1], but their numerical results show that its performance significantly degrades compared to setting of perfect knowledge of the score, see [Saremi et al., 2024, Appendix H]. We include this approach in our numerical benchmark. Alternatively, for high-dimensional settings, they propose an estimator based on posterior sampling, see [Saremi et al., 2024, Appendix F.2.], in the same fashion as in `SLIPS`. However, they do not pay attention to the limitation of this approach, in particular at the initialization of their algorithm. Indeed, while [Saremi et al., 2024, Theorem 1] suggests to take σ very large to ensure that the distribution of Y^1 is approximately log-concave (*i.e.*, taking t_0 small in `SLIPS`), the corresponding posterior becomes unfortunately closer to π, and then hard to sample with standard MCMC methods when π is not log-concave. This fundamental constraint is well illustrated in [Saremi et al., 2024, Figure 7, left], where the posterior sampling approach is shown to systematically fail for an

arbitrary choice of σ, independently of the computational budget allocated to the MCMC sampling. Therefore, the combination of OAT with posterior sampling requires a trade-off on the hyper-parameter σ, similarly to t_0 in SLIPS, that reflects once again the "duality of log-concavity".

C.8 Details on numerical experiments

C.8.1 The failure of MCMC methods when targeting multi-modal distributions

As we show in Figure C.9, *local* MCMC samplers such as MALA, HMC, NUTS or ESS tend to produce Markov chains that get trapped in modes while our methodology SLIPS generates samples reaching both modes.

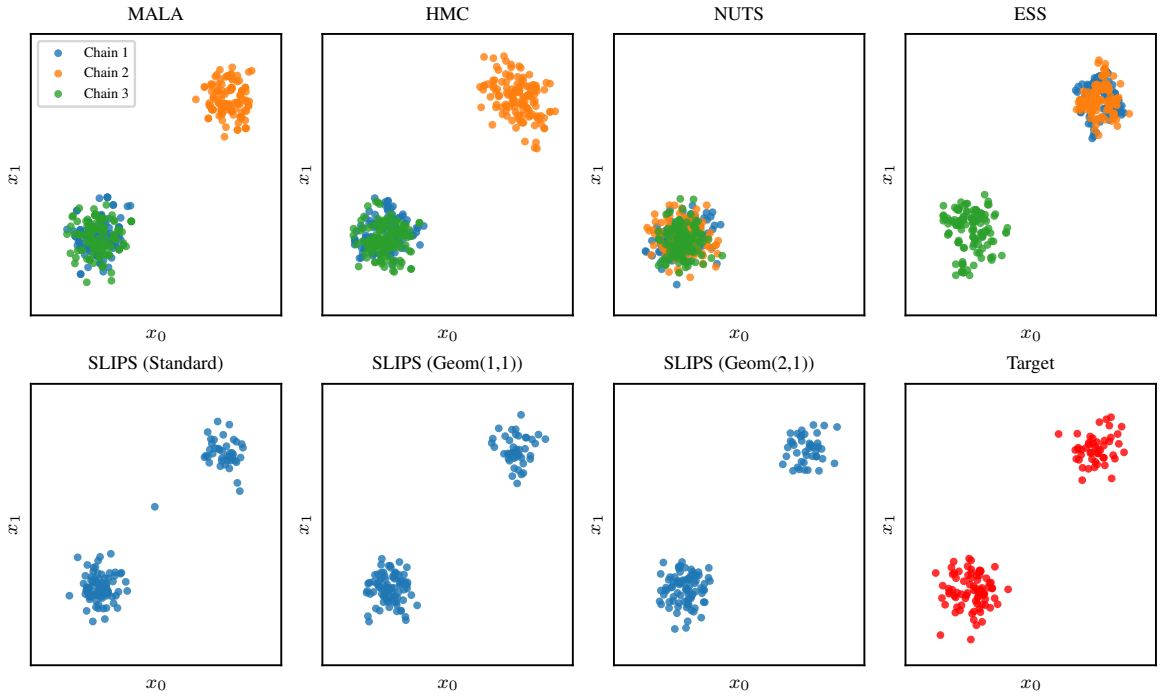


Figure C.9: **Samples from the Gaussian mixture defined in Section 6 ($d = 8$) obtained using different algorithms.** Here, we display the first two coordinates of $n = 128$ samples. For SLIPS, we used the values of t_0 and η given in Table 5, $K = 10$ discretization steps and $n_{\mathrm{MCMC}} = 16$ MCMC steps. On the other hand, the standard MCMC algorithms produce 3 chains, each one being of size $K \times n_{\mathrm{MCMC}} \times n$ for fairness of comparison on the number of target density evaluations; we only display the n last MCMC samples of each chain. The chains were initialized in $\mathrm{N}(0, \sigma^2 \mathrm{I}_d)$. For HMC, we grid-searched an optimal trajectory length in $[4, 8, 12, 16]$ which ended up being 8 (the step-size was automatically tuned). For ESS, we used $\mathrm{N}(0, \sigma^2\, \mathrm{I}_d)$ as a prior.

C.8.2 Algorithms and hyper-parameters

Using the information from R_π and adapting the algorithms. As our definition of σ in SLIPS requires to have an estimate of R_π (see Section 4.3.2), we informed other algorithms with such knowledge when possible. More precisely, when it is possible, we determine R_π analytically; otherwise, we use ground truth samples to obtain a Monte Carlo estimation of R_π. Besides this, we assume that R and τ are assumed to be known for each target distribution verifying **A**3.

- For SMC and AIS, we use a starting distribution ρ_0 as the centered Gaussian with variance $R^2 d\, \mathrm{I}_d$;
- For OAT, we use σ as per [Saremi et al., 2024, Equation 4.3] by setting $\sigma^2 = R^2 d - \tau^2$ (ensuring log-concavity);
- For RDMC, we do not include this information as the authors do not give any heuristic on T with respect to the variance of the target distribution.

Moreover, the implementations of the algorithms were slightly adjusted from their general definition. SMC and AIS use a MALA kernel (which leaves the target distribution invariant) as transition kernel. Additionally,

Table C.1: **Hyper-parameter grids used for SLIPS on different targets.**

		MoG and Bayesian	ϕ^4 model	Others
Standard	η	$\{5.0\}$	$\{5.7, 6.1\}$	$\{5.0, 5.7\}$
	t_0	$\{0.03, 0.05, 0.1, 0.2, 0.4\}$	$\{0.8, 1.0, 1.2, 1.4, 1.8\}$	$\{0.1, 0.2, 0.4, 1.0, 1.2\}$
Geom(1,1)	η	$\{5.0\}$	$\{5.7, 6.1\}$	$\{4.6, 5.0\}$
	t_0	$\{0.03, 0.05, 0.1, 0.15, 0.25\}$	$\{0.30, 0.35, 0.40, 0.45\}$	$\{0.1, 0.15, 0.20\}$
Geom(2,1)	η	$\{5.0\}$	$\{5.7, 6.1\}$	$\{4.6, 5.0\}$
	t_0	$\{0.15, 0.20, 0.25, 0.35, 0.45\}$	$\{0.40, 0.45, 0.50, 0.55\}$	$\{0.30, 0.35, 0.45\}$

RDMC also uses a MALA kernel for posterior sampling instead of ULA. This reduces the estimation bias and also enables automatic tuning of the Langevin step-size by leveraging the acceptance ratio (here we adapt the step-size to maintain the acceptance ratio at 75%). Still on RDMC, we drop the first 50% of the MCMC samples to ignore the warm-up period in the estimation. We also ignore the warm-up period with the same proportion in SLIPS. In SLIPS, we reuse the step-size from the MCMC sampling of q_{t_k} when sampling $q_{t_{k+1}}$ and we also initialize the chains of the later with the last samples of the chain from the former. This is an intuitive initialization when looking at the bottom row of Figure 4.3 as the modes of the posterior seem to be stable and q_{t_k} is expected to be close to $q_{t_{k+1}}$.

Estimating the scores in OAT. In OAT, denoting $p(y)$ the likelihood of the measurement process $Y = X + \sigma Z$ with $X \sim \pi$ and $Z \sim \mathrm{N}(0, \mathrm{I}_d)$, the score can be written using the following identity from [Saremi et al., 2024, Appendix F] : $\nabla \log p(y) = \sigma^{-1} \mathbb{E}_{Z \sim q(\cdot|y;\sigma)}[Z]$ where the posterior $q(\cdot|y)$ is defined by $q(z|y) \propto \pi(y + \sigma z)\mathrm{N}(z; 0, \mathrm{I}_d)$. This means that the score $\nabla \log p(y)$ can be estimated with the following IS estimator $\nabla \log p(y) \approx \sigma^{-1} \sum_{i=1}^N w_i Z_i$ where $Z_i \overset{i.i.d.}{\sim} \mathrm{N}(0, \mathrm{I}_d)$ and $w_i = \pi(y + \sigma Z_i) / \sum_{j=1}^N \pi(y + \sigma Z_j)$. We reduce the variance of this estimator by applying the antithetic trick.

Selecting the hyper-parameters of the algorithms. For each algorithm, we search its hyper-parameters within a predetermined grid. The selection is based on the metrics which will be later detailed. The metrics were computed by comparing 4096 samples against true samples. We globally fixed the computational budget by setting the SDE discretization of SLIPS as $K = 1024$. Moreover, we define the number of MCMC steps of SLIPS, denoted by L, to be equal to 32 except with the mixture of Gaussian in high dimensions where it is equal to 48, 64 and 96 in dimensions 32, 64 and 128 respectively and in the ϕ^4 experiments were it is set to 64. The selected hyper-parameters for each algorithm are summarized in Table C.2. Below, we detail how the grids were built for each one.

- The SMC and AIS algorithms define a sequence of annealed distributions ρ_k for $k \in \{0, \dots, K\}$ from ρ_0 (defined above) to $\rho_K = \pi$ as $\rho_k \propto \exp((1 - \beta_k) \log \rho_0 + \beta_k \log \pi))$ where $(\beta_k)_{k=0}^K$ is a linear schedule of size K between 0 and 1. Both algorithms used $N = 4096$ particles;
- The RDMC algorithm has a single hyper-parameter T which we search within the grid given in [Huang et al., 2024a, Appendix F.1]

$$T \in \{-\log 0.99, -\log 0.95, -\log 0.9, -\log 0.8, -\log 0.7\} .$$

  However, we also find that large T may not work systematically (see Figure C.8). We use 16 steps of [Huang et al., 2024a, Algorithm 3] with 4 MCMC chains. The chains are initialized with an IS approximation of the posterior powered by 128 particles. The SDE is discretized over K steps and the expectation leading to the drift is estimated with L Langevin steps. The initial sample is distributed according to $\mathrm{N}(0, (1 - \exp(-2T))\, \mathrm{I}_d)$, the best estimation of p_T that we have, and the initial step-size is taken according to its variance ;
- The OAT algorithm has its parameter M set as $\lfloor K/2 \rfloor$ and uses as many MCMC steps per noise level as SLIPS or RDMC. This choice of K ensures that the computational complexity of OAT is on par with the other algorithms. The Langevin steps are done using the underdamped Langevin algorithm as suggested by the authors. Its step-size is searched in $\{0.03, 1.0\}$ and the efficient friction is searched in $\{0.0625, 0.05\}$ as recommended by the authors. These prescriptions were extracted from [Saremi et al., 2024, Appendix G]. The OAT algorithm has slightly shorter grid sizes because of its prohibitive computational cost;

- The SLIPS algorithm is declined in three flavors depending on the choice of the schedule α. Since the choice of t_0 is sensitive to the accuracy of the estimation of R_π (different values of R_π will shift the log-SNR upwards or downwards), we decided to search the hyper-parameters in different areas depending on the target distribution. Those grids can be found in Table C.1. The values for t_0 were chosen by approximate equal log-SNR spacing in $[-3.5, -1.0]$ for Gaussian mixtures and Bayesian logistic regression, in $[-1.0, 0.2]$ for ϕ^4 and $[-2.0, 0.0]$ for the others. The values for η were chosen to be around 5.0.

Table C.2: **Hyper-parameters selected for the experiments for each algorithm and target density.**

Target	RDMC	OAT		SLIPS Standard		SLIPS Geom(1,1)		SLIPS Geom(2,1)	
	Time T	step-size δ	Fricт. $\delta\gamma$	η	t_0	η	t_0	η	t_0
8-Gaussians	$-\log(0.80)$	0.05	0.03	5.7	0.60	5.7	0.35	5.0	0.35
Rings	$-\log(0.80)$	0.0625	1.0	4.6	1.20	4.6	0.10	4.6	0.30
Funnel	$-\log(0.90)$	0.05	0.03	5.0	1.00	4.6	0.30	4.6	0.40
Mixture ($d = 8$)	$-\log(0.70)$	0.0625	0.03	5.0	0.40	5.0	0.25	5.0	0.45
Mixture ($d = 16$)	$-\log(0.70)$	0.0625	0.03	5.0	0.20	5.0	0.15	5.0	0.35
Mixture ($d = 32$)	$-\log(0.70)$	0.05	0.03	5.0	0.10	5.0	0.10	5.0	0.25
Mixture ($d = 64$)	$-\log(0.70)$	0.0625	0.03	5.0	0.05	5.0	0.05	5.0	0.20
Ionosphere	$-\log(0.95)$	0.0625	0.03	5.0	0.03	5.0	0.03	5.0	0.15
Sonar	$-\log(0.95)$	0.0625	0.03	5.0	0.03	5.0	0.03	5.0	0.15
ϕ^4 ($b = 0$)	$-\log(0.95)$	0.0625	0.03	5.7	0.80	5.7	0.30	5.7	0.40
ϕ^4 ($b = 0.025$)	$-\log(0.95)$	0.0625	1.0	5.7	1.80	5.7	0.35	6.1	0.45
ϕ^4 ($b = 0.05$)	$-\log(0.70)$	0.05	1.0	6.1	1.00	5.7	0.30	5.7	0.40
ϕ^4 ($b = 0.075$)	$-\log(0.70)$	0.0625	0.03	5.7	1.80	5.7	0.35	5.7	0.40
ϕ^4 ($b = 0.1$)	$-\log(0.90)$	0.05	0.03	5.7	1.40	5.7	0.45	5.7	0.40

C.8.3 Target distributions and metrics

8 Gaussians, Rings and Funnel distributions and their respective metrics.

(a) The 8 Gaussians distribution consists of 8 equally weighted Gaussian distributions with mean $\mathbf{m}_i = 10 \times (\cos(2\pi i/8), \sin(2\pi i/8))$ for $i \in \{0, \dots, 7\}$ and covariance $0.7\,\mathrm{I}_2$. This distribution satisfies **A**3 with $R = 10/\sqrt{2}$ and $\tau^2 = 0.7$.

(b) The Rings distribution is the inverse polar reparameterization of a distribution p_z which has itself a decomposition into two univariate marginals p_r and p_θ: p_r is a mixture of 4 Gaussian distributions $\mathrm{N}(i+1, 0.15^2)$ with $i \in \{0, \dots, 3\}$ describing the radial position and p_θ is a uniform distribution over $[0, 2\pi]$, which describes the angular position of the samples. This distribution satisfies **A**3 with $R = 4/2$ and $\tau = 0.15$.

(c) The Funnel distribution Neal [2003] has its density given by

$$x_{1:10} \mapsto \mathrm{N}(x_1; 0, \sigma^2 \mathrm{I}_{10}) \mathrm{N}(x_{2:10}; 0, \exp(x_1) \mathrm{I}_{10}) \,,$$

where $\sigma^2 = 9$. We approximate R by the maximum scalar standard deviation of the distribution, *i.e.*, $R = 2.1$, and set $\tau = 0$.

For 8 Gaussians and Rings, we evaluate the quality of sampling by approximating the entropy-regularized 2-Wasserstein distance defined in Appendix C.1, with regularization $\varepsilon = 0.05$ via POT library Flamary et al. [2021]. Regarding the Funnel, we evaluate the quality of sampling using the Kolmogorov-Smirnov distance, see Appendix C.1, and adopt the sliced version from [Grenioux et al., 2023b, Appendix D.1].

Bayesian Logistic Regression models. Consider a training dataset $\mathcal{D} = \{(x_j, y_j)\}_{j=1}^M$ where $x_j \in \mathbb{R}^d$ and $y_j \in \{0, 1\}$ for all $j \in \{1, \dots, M\}$. We evaluate the likelihood of a pair (x, y) as given by $p(y|x; w, b) = \mathrm{Bernoulli}(y; \sigma(x^T x + b))$ where $w \in \mathbb{R}^d$ is a weight vector, $b \in \mathbb{R}$ is an intercept and σ is the sigmoid function. Given a prior distribution $p(w, b)$, we sample from the posterior distribution $p(w, b|\mathcal{D}) \propto p(\mathcal{D}|w, b)p(w, b) = \prod_{j=1}^M p(y_j|x_j; w, b)p(w, b)$. The prior is built as $p(w, b) = \mathrm{N}(w; 0, \mathrm{I}_d)\mathrm{N}(b; 0, (2.5)^2)$. In our experiments, we approximate R by the maximum scalar standard deviation of the prior distribution, *i.e.*, $R = 1.1$ and set $\tau = 0$. The quality of the samples are obtained by computing the mean predictive log-likelihood (*i.e.*, computing $p(w, b|\mathcal{D}_{\text{test}})$ with $\mathcal{D}_{\text{test}}$ a separate test dataset).

High dimensional Gaussian mixture. We consider the mixture of two Gaussian distributions defined in (C.39). Following the computations of Appendix C.3, we set $R = 2\sqrt{2}/3$ and $\tau^2 = 0.05$. Complementary to the estimation error on the relative weight given in the main paper, we also report the Sliced Wasserstein distance Bonneel et al. [2015], Nadjahi et al. [2019] on Figure C.10 (*left*). This figure shows that SLIPS also recovers the local structure of the target distribution quite well. Moreover, Figure C.10 (*right*) shows that SMC and AIS are unable to scale with dimension and collapse into a single mode.

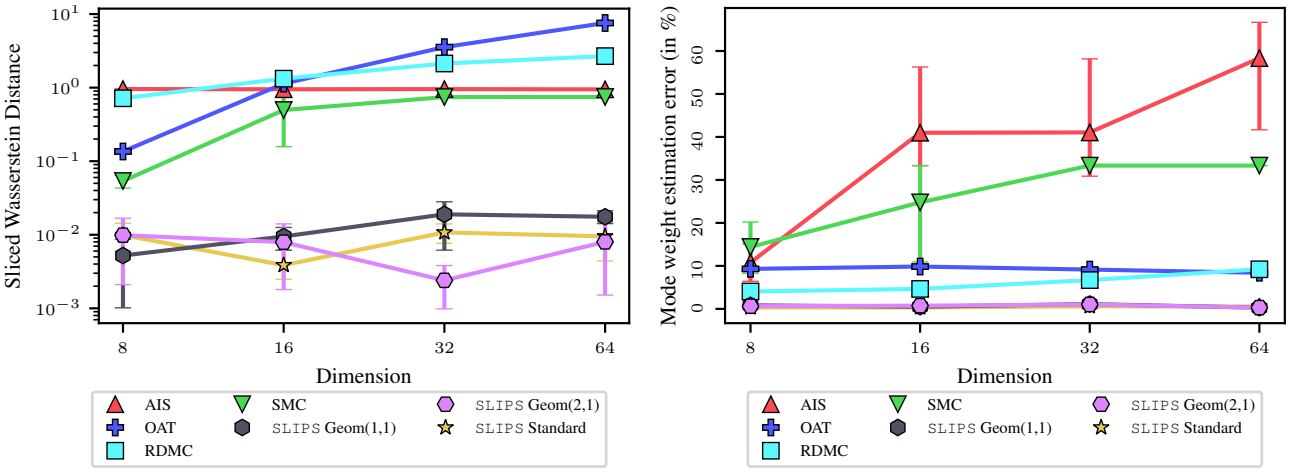


Figure C.10: **(Left): Sliced Wasserstein distance computed on bimodal Gaussian mixtures with increasing dimension. (Right): Estimation error when computing the relative weight of the corresponding two modes.**

The ϕ^4 distribution. The ϕ^4 model is a toy model defined as a continuous relaxation of the Ising model that serves the study of phase transitions in statistical mechanics. Following Gabrié et al. [2022], we consider a version of the model discretized on a 1-dimensional grid of size $d = 100$. One configuration is therefore a d-dimensional vector $(\phi_i)_{i=1}^d$. We additionally clip the field to 0 at both extremities by defining the extra $\phi_0 = \phi_{d+1} = 0$. The negative log-density of the distribution writes

$$\ln \pi_h(\phi) = -\beta \left(\frac{ad}{2} \sum_{i=1}^{d+1} (\phi_i - \phi_{i-1})^2 + \frac{1}{4ad} \sum_{i=1}^{d} (1 - \phi_i^2)^2 + h\phi_i \right) . \tag{C.91}$$

We chose parameter values for which the system is bimodal, $a = 0.1$ and inverse temperature $\beta = 20$, and vary the value of h. We denote by w_+ the statistical occurrence of configurations such that $\phi_{d/2} > 0$ and w_- the statistical occurrence of configurations such that $\phi_{d/2} < 0$. At $h = 0$, the measure is invariant under the symmetry $\phi \to -\phi$, such that we expect $w_+ = w_-$. For $h > 0$, the negative mode dominates. We plot the two modes on Figure C.11.

When d is large, the relative probability of the modes can be estimated by a Laplace approximations at 0-th and 2-nd order. Denoting by ϕ_+^h and ϕ_-^h the local maxima of (C.91), these approximations yield respectively,

$$\frac{w_-}{w_+} \approx \frac{\pi_h(\phi_-^h)}{\pi_h(\phi_+^h)} , \quad \frac{w_-}{w_+} \approx \frac{\pi_h(\phi_-^h) \times |\det H_h(\phi_-^h)|^{-1/2}}{\pi_h(\phi_+^h) \times |\det H_h(\phi_+^h)|^{-1/2}} , \tag{C.92}$$

where H_h is the Hessian of the function $\phi \to \ln \pi_h(\phi)$. In our experiments, we considered $R = 0.85$ and $\tau = 0.15$ by running MALA chains started in each mode : τ^2 is set as the variance within the modes while R corresponds to the distance between the modes and 0_{100}. In this setting, we observed that SMC and AIS suffered from mode collapse, while OAT and RDMC produced degenerate samples; in contrast, our methodology `SLIPS` recovers accurate samples from the target distribution, with correct relative weight, as shown in Figure 4.5.

C.8.4 Empirical complexity of `SLIPS`

The results of Figure C.12 ($d = 32$) show that under a much lower computational budget ($K = 20$ here) than in Section 4.6 ($K = 1024$), `SLIPS` maintains the same performance, while still outperforming its competitors. Moreover, these results demonstrate that `SLIPS` has very reasonable execution times (below 1 minute to obtain 8192 samples) making it completely practical. Lastly, Figure C.12 emphasizes that, even using high computational budgets, competitors cannot solve such multi-modal tasks in high-dimension. Note that those observations stay valid in problems with lower dimension (see Figure C.13 where $d = 4$).

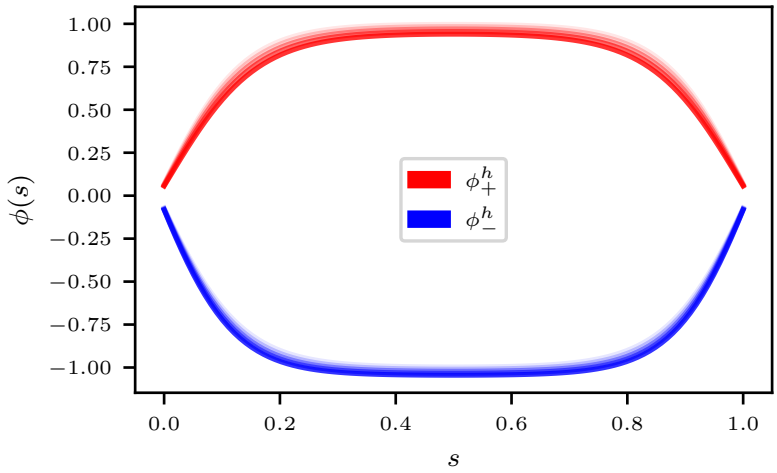


Figure C.11: **Modes of the ϕ^4 distribution (ϕ_+^h in shades of red and ϕ_-^h in shades of blue) for different values of h (corresponding to the transparency levels).** The time interval $[0, 1]$ on the x-axis is discretized into a grid of size $d = 100$, which corresponds to the dimensionality of the samples.

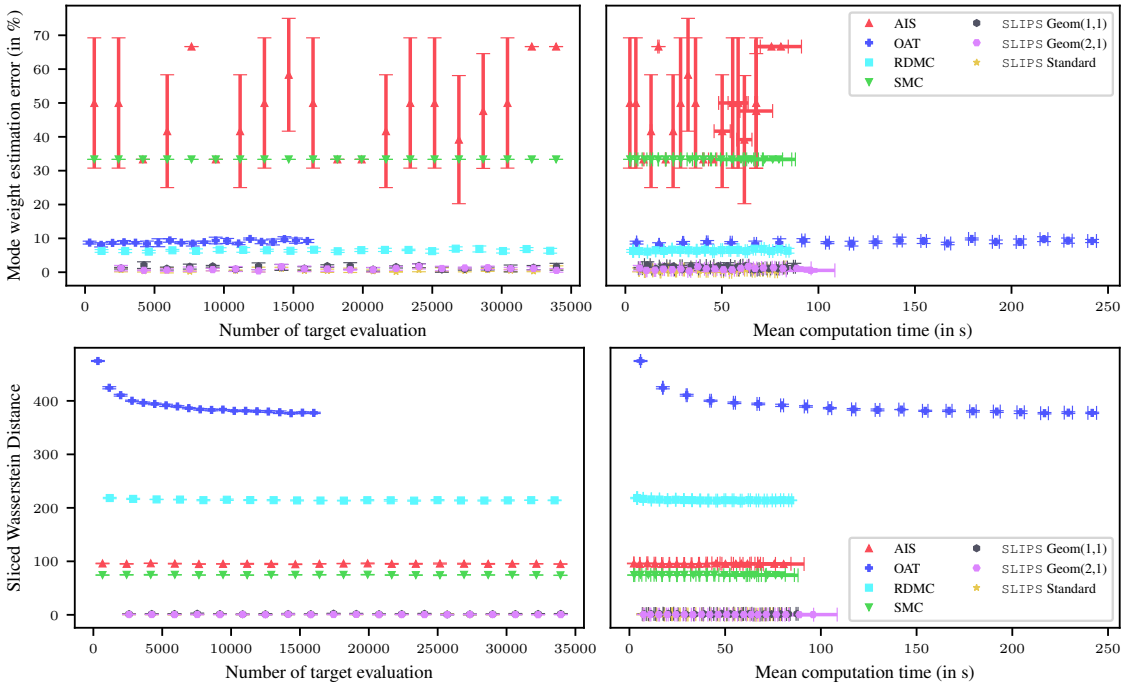


Figure C.12: **Metrics when sampling from the Gaussian mixture defined in Section 4.6 ($d = 32$) using different algorithms. Top**: Weight estimation error. **Bottom**: Sliced Wasserstein Distance. **Left**: Metric depending on the number of target density evaluations. **Right**: Metric depending on wall time. The computational budgets are computed by evolving K linearly from 20 to 90. The number of MCMC steps is fixed at 32. The computations were run on the same Nvidia V100 GPU.

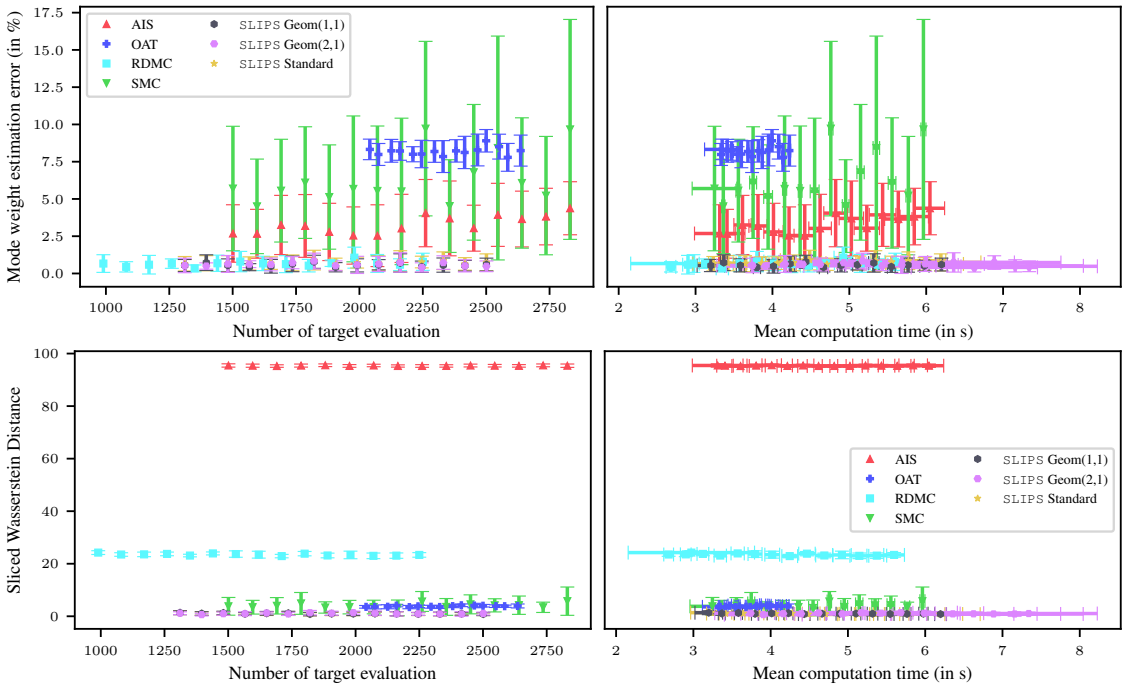


Figure C.13: **Metrics when sampling from the Gaussian mixture defined in Section 4.6 ($d = 4$) using different algorithms. Top**: Weight estimation error. **Bottom**: Sliced Wasserstein Distance. **Left**: Metric depending on the number of target density evaluations. **Right**: Metric depending on wall time. The computational budgets are computed by evolving K linearly from 20 to 90. The number of MCMC steps is fixed at 16. The computations were run on the same Nvidia V100 GPU. The computational budgets were aligned to match SLIPS's.

Appendix D

Appendix of Chapter 5

Organization of the supplementary

The appendix is organized as follows. In Appendix D.1, we provide the pseudo-codes of every RDS instance presented in the main part of this paper (standard RDS, GMM-LRDS & EBM-LRDS). Appendix D.2 summarizes general facts that will be useful for proofs and corresponding computations. In Appendix D.3, we describe the noising diffusion schemes considered in this work, namely the Pinned Brownian Motion and the Variance-Preserving setting, and propose novel schemes. For RDS and related variational diffusion-based methods, we dispense details on their continuous-time and corresponding discrete-time variational objectives in Appendix D.4. We respectively provide an overview of existing annealed MCMC methods and multi-level Energy-Based models in Appendix D.5 and Appendix D.6. In Appendix D.7, we provide expressions of additional variational metrics proposed by Blessing et al. [2024] in the RDS setting. Details on our experimental settings, our implementation of RDS and other competing methods are given in Appendix D.8. Finally, we present additional theoretical and experimental results in Appendix D.9.

Our codebase is available at https://github.com/h2o64/sde_sampler_lrds.

Notation. Let $\mathbb{P} \in \mathscr{P}(\mathbf{C}_T)$ and $t \in [0,T]$. For any $x_t \in \mathbb{R}^d$, we denote by $\mathbb{P}^{x_t}_{|t} \in \mathscr{P}(\mathbf{C}_T)$ the path measure associated to the stochastic process $(X_t)_{t\in[0,T]} \sim \mathbb{P}$ conditioned on $X_t = x_t$. Furthermore, for any $\pi \in \mathscr{P}(\mathbb{R}^d)$, $\pi \otimes \mathbb{P}^{x_t}_{|t} \in \mathbf{C}_T$ stands for the path measure $\int_{\mathbb{R}^d} \mathbb{P}^{x_t}_{|t} \mathrm{d}\pi(x_t)$. For any $K \geq 1$, we denote the set of joint distributions $\mathscr{P}((\mathbb{R}^d)^{K+1})$ by $\mathscr{P}^{(K+1)}$. Finally, we adopt the following notation to design sample batches: for any $\bar{L} \geq \underline{L} \geq 0$ and $\bar{K} \geq \underline{K} \geq 0$, we denote by $X^{\underline{L}:\bar{L}}_{\underline{K}:\bar{K}}$ the sequence of samples $\{X^\ell_k\}^{\bar{K},\bar{L}}_{k=\underline{K},\ell=\underline{L}}$.

D.1 Pseudo-codes of RDS-based algorithms

We respectively give sampling procedures and training procedures of a general version of RDS in Algorithm 4 and Algorithm 5. Relying on this, we derive the complete training schemes for GMM-LRDS (Algorithm 6) and EBM-LRDS (Algorithm 7).

Algorithm 4 Reference-based Diffusion Sampling (RDS) : sampling stage

Input: Time discretization $\{t_k\}_{k=0}^K$ of $[0,T]$, reference scores $\{s^{\mathrm{ref}}_{T-t_k}\}_{k=0}^{K-1}$, neural network $\phi^\theta : [0,T] \times \mathbb{R}^d \to \mathbb{R}^d$, variational coefficients $\{a_k, b_k, c_k\}_{k=0}^{K-1}$ defined in Appendix D.4.2

▷ Initialization

$Y_0 \sim \pi^{\mathrm{base}}$

$(Z_k)_{k=0}^{K-1} \overset{\text{i.i.d.}}{\sim} \mathrm{N}(0, \mathrm{I}_d)$

for $k \leftarrow 0$ **to** $K-1$ **do**

  ▷ Compute the k-th diffusion step

  $Y_{k+1} \leftarrow a_k Y_k + b_k\{s^{\mathrm{ref}}_{T-t_k}(Y_k) + \phi^{\hat{\theta}}_{T-t_k}(Y_k)\} + \sqrt{c_k} Z_k$

end

Output: Discrete time process $Y_{0:K}$ approximating $(Y^{\hat{\theta}}_{t_k})_{k=0}^K$

Algorithm 5 Reference-based Diffusion Sampling (RDS) : training stage

Input: Time discretization $\{t_k\}_{k=0}^K$ of $[0,T]$, target density γ, reference density γ^{ref} and scores $\{s^{\mathrm{ref}}_{T-t_k}\}_{k=0}^{K-1}$, neural network $\phi^\theta : [0,T]\times\mathbb{R}^d \to \mathbb{R}^d$ initialized such that $\phi^{\theta_0}=0$, number of training iterations N, batch size B, variational coefficients $\{w_k, a_k, b_k, c_k\}_{k=0}^{K-1}$ defined in Appendix D.4.2

for $n \leftarrow 0$ **to** $N-1$ **do**

▷ Simulate B trajectories of the process $(Y_t^{\hat\theta})_{t\in[0,T]}$

$Y_{0:K}^{1:B} \overset{\text{i.i.d.}}{\sim} \mathrm{SamplingRDS}(\phi^{\theta_n})$, see Algorithm 4

▷ Apply a stochastic gradient descent step on θ_n

Compute a MC estimator $\hat{\mathcal{L}}_{\mathrm{RDS}}(\theta_n)$ of the loss $\mathcal{L}_{\mathrm{RDS}}(\theta_n)$ defined in (5.7) with samples $Y_{0:K}^{1:B}$

Compute the gradient $\nabla_\theta \hat{\mathcal{L}}_{\mathrm{RDS}}(\theta_n)$ and update θ_n to θ_{n+1} with Adam optimizer

end

Output: Learned guidance ϕ^{θ_N}

Algorithm 6 Gaussian Mixture Model LRDS (GMM-LRDS) : training stage

Input: Time discretization $\{t_k\}_{k=0}^K$ of $[0,T]$, target density γ, location of the target modes $\{\mathbf{x}_i\}_{i=1}^I$, number of Markov chains per target mode $M \geq 1$, size of Markov chains $N_{\mathrm{tot}} \geq 1$, effective size of Markov chains $N_{\mathrm{eff}} \in [\![1, N_{\mathrm{tot}}]\!]$, number of reference modes $J \geq I$

▷ (a) Obtain *reference samples*

For each $i \in \{1,\dots,I\}$, build via MALA M Markov chains $\{X^{i,m}_{0:N_{\mathrm{tot}}}\}_{m=1}^M$ of size N_{tot} starting at $\mathbf{x}_i$

Keep the last N_{eff} samples of each Markov chain to define $\hat\pi^{\mathrm{ref}} \cong \{X^{i,m}_{N_{\mathrm{tot}}-N_{\mathrm{eff}}+1:N_{\mathrm{tot}}}\}_{i=1,m=1}^{I,M}$

▷ (b) Define the reference process

if $J=1$ **then**

Fit on $\hat\pi^{\mathrm{ref}}$ a Gaussian model with parameterized density γ^ψ (Maximum Likelihood estimation)

else

Fit on $\hat\pi^{\mathrm{ref}}$ a J-component Gaussian Mixture Model with parameterized density γ^ψ(EM algorithm)

end

Set $\gamma^{\mathrm{ref}} \leftarrow \gamma^\psi$ and compute $\{s^{\mathrm{ref}}_{T-t_k}\}_{k=0}^{K-1}$ analytically, see Appendix D.2

▷ (c) Run the RDS variational optimization

Run Algorithm 5

Output: Trained GMM-LRDS sampler

Algorithm 7 Energy-Based Model LRDS (EBM-LRDS) : training stage

Input: Time discretization $\{t_k\}_{k=0}^K$ of $[0,T]$, target density γ, location of the target modes $\{\mathbf{x}_i\}_{i=1}^I$, number of Markov chains per target mode $M \geq 1$, size of Markov chains $N_{\mathrm{tot}} \geq 1$, effective size of Markov chains $N_{\mathrm{eff}} \in [\![1, N_{\mathrm{tot}}]\!]$

▷ (a) Obtain *reference samples*

For each $i \in \{1,\dots,I\}$, build via MALA M Markov chains $\{X^{i,m}_{0:N_{\mathrm{tot}}}\}_{m=1}^M$ of size N_{tot} starting at $\mathbf{x}_i$

Keep the last N_{eff} samples of each Markov chain to define $\hat\pi^{\mathrm{ref}} \cong \{X^{i,m}_{N_{\mathrm{tot}}-N_{\mathrm{eff}}+1:N_{\mathrm{tot}}}\}_{i=1,m=1}^{I,M}$

▷ (b) Define the reference process

Based on $\hat\pi^{\mathrm{ref}}$, fit a multi-level EBM E^ψ using Algorithm 14

Set $\gamma^{\mathrm{ref}}(x) \leftarrow \exp(-E^\psi(0,x))$ and $s^{\mathrm{ref}}_{T-t_k}(x) = -\nabla_x E^\psi(T-t_k, x)$ for any $k \in \{0,\dots,K-1\}$

▷ (c) Run the RDS variational optimization

Run Algorithm 5

Output: Trained EBM-RDS sampler

D.2 Preliminaries

Linear SDE integration. We first dispense a useful lemma to compute exact integration in SDEs with linear drift.

A11 (Integrability conditions on f and g)**.** *Coefficients $f : [0,T] \to \mathbb{R}$ and $g : [0,T] \to (0,\infty)$ are such that (a) f is integrable on $(0,T)$ and (b) g is integrable on $(0,T)$.*

Lemma 67. *Let $T > 0$ and $b \in \mathbb{R}^d$. Consider the SDE defined on $[0,T]$ by $\mathrm{d}Y_t = f(t)(Y_t + b)\mathrm{d}t + g(t)\mathrm{d}B_t$, where coefficients f and g verify **A**11. Then, for any pair of time-steps (s,t) such that $T \geq t > s \geq 0$, the conditional distribution of Y_t given $Y_s = y_s \in \mathbb{R}^d$, denoted by $p_{t|s}(\cdot|y_s)$, verifies*

$$p_{t|s}(\cdot|y_s) = \mathrm{N}\left(\exp(\textstyle\int_s^t f(u)\mathrm{d}u)y_s + \left(\exp(\int_s^t f(u)\mathrm{d}u) - 1\right)b, \int_s^t g^2(u)\exp(2\int_u^t f(r)\mathrm{d}r)\mathrm{d}u\,\mathrm{I}_d\right).$$

Proof. Assume **A**11. Define the function $\zeta : t \in [0,T] \to \exp(-\int_0^t f(u)\mathrm{d}u)$ and consider the stochastic process

$(Z_t)_{t\in[0,T]}$ defined by $Z_t = \zeta(t)Y_t$ for any $t \in [0,T]$. By Îto's formula, we have $\mathrm{d}Z_t = f(t)\zeta(t)b\mathrm{d}t + \zeta(t)g(t)\mathrm{d}B_t = -\dot{\zeta}(t)b\mathrm{d}t + \zeta(t)g(t)\mathrm{d}B_t$. Therefore, for any time-steps (s,t) such that $T \geq t > s \geq 0$, we have

$$\zeta(t)Y_t - \zeta(s)Y_s = \{\zeta(s) - \zeta(t)\}b + \int_s^t \zeta(u)g(u)\mathrm{d}B_u \;, \tag{D.1}$$

and then

$$Y_t = \exp(\textstyle\int_s^t f(u)\mathrm{d}u)Y_s + \left(\exp(\int_s^t f(u)\mathrm{d}u) - 1\right) b + \int_s^t g(u)\exp(\int_u^t f(r)\mathrm{d}r)\mathrm{d}B_u \;, \tag{D.2}$$

which gives the result using Îto's isometry and that Y_s is independent from $(B_t - B_s)_{t\in[s,T]}$. □

General SDE integration. Consider a general SDE of the form $\mathrm{d}Y_t = f(t)(Y_t + g_t(Y_t))\mathrm{d}t + g(t)\mathrm{d}B_t$ defined on $[0,T]$, where f, g satisfy **A**11 and $g : [0,T]\times\mathbb{R}^d \to \mathbb{R}^d$ is a black-box function. Let (s,t) be a pair of time-steps such that $T \geq t > s \geq 0$. We aim to compute the conditional distribution of Y_t given $Y_s = y_s \in \mathbb{R}^d$. While this distribution is intractable for general function g, it can be approximated by a tractable Gaussian kernel if t is close enough to s. Below, we restate two common first-order methods that produce such approximation following the result of Lemma 67.

(a) *Euler-Maruyama (EM) scheme*, which consists of exactly integrating on time interval $[s,t]$ the SDE defined by $\mathrm{d}Y_u = f(s)(y_s + h_s(y_s))\mathrm{d}u + g(s)\mathrm{d}B_u$ with constant drift and constant volatility.

(b) *Exponential Integration (EI) scheme* [Durmus and Moulines, 2015], more precise than the EM scheme, which consists of exactly integrating on time interval $[s,t]$ the SDE defined by $\mathrm{d}Y_u = f(u)(Y_u + h_s(y_s))\mathrm{d}u + g(u)\mathrm{d}B_u$ with linear drift, where the intractable term is frozen at $h_s(y_s)$.

Computation of scores for Gaussian mixtures. Below, we provide a useful lemma to compute the marginal scores of a noising diffusion process applied to a general Gaussian mixture.

Lemma 68. *Let $\pi \in \mathscr{P}(\mathbb{R}^d)$. Consider the noising process $(X_t)_{t\in[0,T]}$ defined by SDE (5.2) where $X_0 \sim \pi$. Then, for any $t \in (0,T)$, the conditional distribution of X_t given $X_0 = x_0 \in \mathbb{R}^d$ is defined by the Gaussian kernel*

$$q_{t|0}(\cdot|x_0) = \mathrm{N}(S(t)x_0, S(t)^2\sigma^2(t)\,\mathrm{I}_d) \;, \tag{D.3}$$

where $S(t) = \exp(\int_0^t f(u)\mathrm{d}u)$ and $\sigma^2(t) = \int_0^t g^2(u)/S(u)^2\mathrm{d}u$.

In particular, if π is a Gaussian distribution i.e., $\pi = \mathrm{N}(\mathbf{m},\Sigma)$ with mean $\mathbf{m} \in \mathbb{R}^d$ and covariance $\Sigma \in \mathbb{R}^{d\times d}$, then for any $t \in [0,T]$, the marginal distribution of X_t is Gaussian, defined by

$$p_t = \mathrm{N}(S(t)\mathbf{m}, S(t)^2\Sigma + S(t)^2\sigma^2(t)\,\mathrm{I}_d) \;. \tag{D.4}$$

On the other hand, if π is a Gaussian mixture i.e., $\pi = \sum_{j=1}^J w_j\mathrm{N}(\mathbf{m}_j,\Sigma_j)$ where $J \geq 1$ is the number of components, $\{w_j\}_{j=1}^J \in \Delta_J$ are the component weights, $\mathbf{m}_j \in \mathbb{R}^d$ and $\Sigma_j \in \mathbb{R}^{d\times d}$ are respectively the mean and the covariance of the j-th component, for any $t \in [0,T]$, the marginal distribution of X_t is a Gaussian mixture, defined by

$$p_t = \sum_{j=1}^J w_j\mathrm{N}(S(t)\mathbf{m}_j, S(t)^2\Sigma_j + S(t)^2\sigma^2(t)\,\mathrm{I}_d) \;. \tag{D.5}$$

Note that differentiating (D.5) provides closed-form solutions of the scores.

Proof. The proof of (D.3) directly follows from Lemma 67. Then, (D.4) and (D.5) are obtained by applying the affine mapping defined by the transition kernel (D.3) to the distribution π. □

In particular, (D.5) demonstrates that any noising diffusion scheme applied to a Gaussian mixture does not show any mode switching.

Integral probability Metrics. Finally, we dispense the definition of standard probability metrics, that will be used to assess the performance of the considered samplers, *when having access to ground truth samples*. Consider two distributions $\mu \in \mathscr{P}(\mathbb{R}^d)$ and $\nu \in \mathscr{P}(\mathbb{R}^d)$ that we aim to compare.

We recall that the 2-Wasserstein distance between μ and ν is given by

$$W_2(\mu,\nu) = \inf\{\textstyle\int_{\mathbb{R}^d\times\mathbb{R}^d} \|x_1 - x_0\|^2 \mathrm{d}\pi(x_0,x_1) : \pi \in \mathscr{P}(\mathbb{R}^d \times \mathbb{R}^d), \pi_0 = \mu, \pi_1 = \nu\}^{1/2} ,$$

where π_i denotes the i-th marginal of π for $i \in \{0,1\}$. In practice, we rather turn to its entropy regularized version [Peyré et al., 2019]

$$W_{2,\varepsilon}(\mu,\nu) = \inf\{\textstyle\int_{\mathbb{R}^d\times\mathbb{R}^d} \|x_1 - x_0\|^2 \mathrm{d}\pi(x_0,x_1) - \mathscr{H}(\pi) : \pi \in \mathscr{P}(\mathbb{R}^d \times \mathbb{R}^d), \pi_0 = \mu, \pi_1 = \nu\}^{1/2} ,$$

where $\varepsilon > 0$ is a regularization hyper-parameter and $\mathscr{H}(\pi) = -\int_{\mathbb{R}^d\times\mathbb{R}^d} \log \pi(x_0,x_1)\mathrm{d}\pi(x_0,x_1)$ refers to as the entropy of π. When having access to samples from μ and ν, we compute a statistical estimation of $W_{2,\varepsilon}(\mu,\nu)$ via Sinkhorn algorithm [Cuturi, 2013] with $\varepsilon = 10^{-3}$, based on the GPU-oriented implementation from Francis Williams.

We also consider the Maximum Mean Discrepancy (MMD) [Gretton et al., 2012], which quantifies the dissimilarity between μ and ν by comparing their mean embeddings in a reproducing kernel Hilbert space $\mathcal{H}_k$ with kernel k, as followed

$$\mathrm{MMD}(\mu,\nu) = \sup_{f\in\mathcal{H}_K : \|f\|_{\mathcal{H}_K}\leq 1} \mathbb{E}[f(X) - f(Y)],\ X \sim \mu, Y \sim \nu . \tag{D.6}$$

In particular, $\mathrm{MMD}(\mu,\nu) = 0$ if and only if $\mu = \nu$ almost surely. When having access to samples $\{x_i\}_{i=1}^n$ from μ and $\{y_i\}_{i=1}^m$ from ν, we compute an unbiased statistical estimation of $\mathrm{MMD}(\mu,\nu)$ given by

$$\sqrt{\tfrac{2}{n(n-1)}\textstyle\sum_{1\leq i<j\leq n} k(x_i,x_j) + \tfrac{2}{m(m-1)}\sum_{1\leq i<j\leq m} k(y_i,y_j) - \tfrac{2}{mn}\sum_{i=1}^n\sum_{j=1}^m k(x_i,y_j)} ,$$

where $k(x,y) = \exp\left(-\|x-y\|^2/(2\alpha)\right)$ is the squared exponential kernel whose positive bandwidth is fixed as the median of the squared pairwise distances computed on the joint set $\{x_i\}_{i=1}^n \cup \{y_j\}_{j=1}^m$ [Gretton et al., 2012].

Finally, we recall that the Kolmogorov-Smirnov (KS) distance between μ and ν is defined by

$$\mathrm{KS}(\mu,\nu) = \sup_{x\in\mathbb{R}^d} |F_\mu(x) - F_\nu(x)| , \tag{D.7}$$

where F_μ (respectively F_ν) denotes the cumulative distribution function of μ (respectively ν). When having access to samples from μ and ν, we implement a statistical estimation of the sliced KS distance [Grenioux et al., 2023b] using 128 random projections.

D.3 Details on noising diffusion processes

In this section, we consider a noising diffusion process $(X_t)_{t\in[0,T]}$ induced by SDE (5.1) for an arbitrary target distribution $\pi \in \mathscr{P}(\mathbb{R}^d)$ and $T > 0$, and denote by $\mathbb{P}^\star \in \mathscr{P}(\mathbf{C}_T)$ the corresponding path measure. Below, we provide detailed results on this diffusion process, depending on the choice of f and g: we first provide details on the general setting in Appendix D.3.1; then, we fully describe the Pinned Brownian Motion and the Variance-Preserving scheme in Appendix D.3.2 and Appendix D.3.3, that are used in our experiments; finally, we present novel noising schemes in Appendix D.3.4.

D.3.1 General noising scheme

Here, we consider the most general form of SDE (5.1).

Lemma 69. *Let $\pi \in \mathscr{P}(\mathbb{R}^d)$. Assume that f and g both verify **A**11. Then, for any pair of time-steps (s,t) such that $T \geq t > s \geq 0$, the conditional distribution of X_t given $X_s = x_s \in \mathbb{R}^d$ is defined by the Gaussian kernel*

$$q_{t|s}(\cdot|x_s) = \mathrm{N}\left(\exp(\log S(t) - \log S(s))x_s, S(t)^2\{\sigma^2(t) - \sigma^2(s)\}\,\mathrm{I}_d\right) , \tag{D.8}$$

where $S(t) = \exp(\int_0^t f(u)\mathrm{d}u)$ and $\sigma^2(t) = \int_0^t g^2(u)/S(u)^2\mathrm{d}u$.

Proof. The proof of (D.8) directly follows from Lemma 67. □

D.3.2 Pinned Brownian motion

For any $t \in [0, T]$, define $\alpha(t) = \int_0^t g^2(t)\mathrm{d}t$ and consider the case where

$$f(t) = -\frac{g^2(t)}{\alpha(T) - \alpha(t)} \, .$$

Then, SDE (5.1) can be rewritten as

$$\mathrm{d}X_t = -\frac{g^2(t)X_t}{\alpha(T) - \alpha(t)}\mathrm{d}t + g(t)\mathrm{d}W_t \, , \; X_0 \sim \pi \, . \tag{D.9}$$

This noising scheme, known as the *Pinned Brownian Motion* (PBM) and previously considered by Tzen and Raginsky [2019], Zhang and Chen [2022], Vargas et al. [2023b], can be obtained by applying a Doob's h-transform on the uncontrolled SDE $\mathrm{d}X_t = g(t)\mathrm{d}W_t$, associated to path measure $\mathbb{Q}$, to hit 0 at time T. Interestingly, it can be shown that the resulting path measure $(\mathbb{P}^\star)^R$ is solution to the following stochastic optimal control problem

$$\arg\min\{\mathrm{KL}(\mathbb{P} \mid \mathbb{Q}) \, : \, \mathbb{P} \in \mathscr{P}(\mathbf{C}_T), \mathbb{P}_T = \pi\} \, , \tag{D.10}$$

which is often referred to as a half Schrödinger Bridge problem.

Lemma 70. *Let $\pi \in \mathscr{P}(\mathbb{R}^d)$. Assume that $f(t) = -\frac{g^2(t)}{\alpha(T)-\alpha(t)}$ and g verifies **A**11. Then, for any pair of time-steps (s,t) such that $T \geq t > s \geq 0$, the conditional distribution of X_t given $X_s = x_s \in \mathbb{R}^d$ is defined by the Gaussian kernel*

$$q_{t|s}(\cdot|x_s) = \mathrm{N}\left(\frac{\alpha(T) - \alpha(t)}{\alpha(T) - \alpha(s)}x_s, \frac{(\alpha(T) - \alpha(t))(\alpha(t) - \alpha(s))}{\alpha(T) - \alpha(s)}\mathrm{I}_d\right) \, . \tag{D.11}$$

Since $p_T^\star(x) = \int_{\mathbb{R}^d} q_{T|0}(x|x_0)\mathrm{d}\pi(x_0)$, it results that $\mathbb{P}_T^\star = \delta_0$.

Proof. The proof of (D.11) directly follows from Lemma 67. □

Following Lemma 70, the PBM is an 'exact' noising scheme and we have $\pi^{\mathrm{base}} = \delta_0$ in this setting. Moreover, under mild assumptions on π, the time-reversed SDE (5.2) writes as

$$\mathrm{d}Y_t = g^2(T-t)\left\{\frac{Y_t}{\alpha(T) - \alpha(T-t)} + \nabla \log p_{T-t}^\star(Y_t)\right\}\mathrm{d}t + g(T-t)\mathrm{d}B_t, \; Y_0 = 0 \, . \tag{D.12}$$

On the choice of the g-schedule. Previous works have considered constant schedule $g^2(t) = \sigma^2$ (and therefore, $\alpha(t) = \sigma^2 t$), where $\sigma > 0$ can be arbitrarily chosen, see e.g., Zhang and Chen [2022], Richter et al. [2023]. We also follow this setting in practice.

D.3.3 Variance-Preserving diffusion

For any $t \in [0, T]$, denote $\alpha(t) = \int_0^t g^2(t)\mathrm{d}t$ and consider the case where $f(t) = -g^2(t)/2\sigma^2$ with $\sigma > 0$ and g being such that $\int_0^T g^2(s)\mathrm{d}sg1$. Then, SDE (5.1) can be rewritten as

$$\mathrm{d}X_t = -\frac{g^2(t)}{2\sigma^2}X_t\mathrm{d}t + g(t)\mathrm{d}W_t \, , \; X_0 \sim \pi \, . \tag{D.13}$$

This noising scheme, known as the *Variance-Preserving* (VP) scheme [Song et al., 2021b] and previously considered by Vargas et al. [2023a], is an Ornstein-Uhlenbeck diffusion process and is largely used in score-based generative models.

Lemma 71. *Let $\pi \in \mathscr{P}(\mathbb{R}^d)$ and $\sigma > 0$. Assume that $f(t) = -g^2(t)/2\sigma^2$ and g verifies **A**11 such that $\int_0^T g^2(s)\mathrm{d}sg1$. Then, for any pair of time-steps (s,t) such that $T \geq t > s \geq 0$, the conditional distribution of X_t given $X_s = x_s \in \mathbb{R}^d$ is defined by the Gaussian kernel*

$$q_{t|s}(\cdot|x_s) = \mathrm{N}\left(\sqrt{1 - \lambda_{s,t}}x_s, \sigma^2\lambda_{s,t}\mathrm{I}_d\right) \, , \tag{D.14}$$

where $\lambda_{s,t} = 1 - \exp(\alpha(s) - \alpha(t))$. Using that $p_T^\star(x) = \int_{\mathbb{R}^d} q_{T|0}(x|x_0)\mathrm{d}\pi(x_0)$, it results that $\mathbb{P}_T^\star \approx \mathrm{N}(0, \sigma^2\,\mathrm{I}_d)$.

Proof. The proof of (D.14) directly follows from Lemma 67, along with fact that $\lambda_{0,T} \approx 1$. □

Following Lemma 71, the VP scheme is an 'ergodic' noising scheme, converging exponentially fast to the Gaussian distribution $\mathrm{N}(0, \sigma^2 \mathrm{I}_d)$; therefore, we have $\pi^{\mathrm{base}} = \mathrm{N}(0, \sigma^2 \mathrm{I}_d)$ in this setting. Moreover, under mild assumptions on π, the time-reversed SDE (5.2) writes as

$$\mathrm{d}Y_t = \frac{g^2(T-t)}{2}\{Y_t + 2\sigma^2 \nabla \log p^\star_{T-t}(Y_t)\}\mathrm{d}t + \sigma g(T-t)\mathrm{d}B_t,\ Y_0 \sim \mathbb{P}^\star_T\ , \tag{D.15}$$

after simple linear time reparameterization.

On the choice of the g-schedule. Previous works have considered a linear schedule $g^2(t) = \beta_{\min}(1-t/T) + \beta_{\max}(t/T)$ where $\beta_{\min} = 0.1$, $\beta_{\max} \in \{10, 20\}$ and $T = 1$, see e.g., Song et al. [2021b], Richter et al. [2023], Reu et al. [2024] or cosine parameterization [Nichol and Dhariwal, 2021, Vargas et al., 2023a], which has been proved to perform better in generative modeling. In our sampling experiments, we did not observe any significant difference between these two settings. Hence, we fix the linear g-schedule to be the default setting for our numerics (except DDS, originally implemented with the cosine schedule), and let σ be arbitrarily chosen.

D.3.4 Additional examples of noising diffusion processes

Below, we present original noising schemes, but did not consider them in our experiments.

Pinned Ornstein-Uhlenbeck Motion. We first propose to consider the case where $f(t) = -\frac{g^2(t)}{2}\coth(\int_t^T \frac{g^2(u)}{2}\mathrm{d}u)$. Then, SDE (5.1) can be rewritten as

$$\mathrm{d}X_t = -\frac{g^2(t)}{2}\coth(\textstyle\int_t^T \frac{g^2(u)}{2}\mathrm{d}u)X_t\mathrm{d}t + g(t)\mathrm{d}W_t,\ X_0 \sim \pi\ . \tag{D.16}$$

Analogously to the PBM, one can show that this noising scheme can be obtained by applying a Doob's h-transform on the Ornstein-Uhlenbeck process induced by the SDE $\mathrm{d}\tilde{X}_t = -\frac{g^2(t)}{2}\tilde{X}_t\mathrm{d}t + g(t)\mathrm{d}W_t$ to hit 0 at time T. We refer to this noising diffusion scheme as the *Pinned Ornstein-Uhlenbeck Motion* (POUM).

Lemma 72. *Let $\pi \in \mathscr{P}(\mathbb{R}^d)$. Assume that $f(t) = -\frac{g^2(t)}{2}\coth(\int_t^T \frac{g^2(u)}{2}\mathrm{d}u)$ and g verifies **A**11. Then, for any pair of time-steps (s,t) such that $T \geq t > s \geq 0$, the conditional distribution of X_t given $X_s = x_s \in \mathbb{R}^d$ is defined by the Gaussian kernel $q_{t|s}(\cdot|x_s)$ given by*

$$\mathrm{N}\left(\frac{\left|\sinh(\int_t^T \frac{g^2(u)}{2}\mathrm{d}u)\right|}{\left|\sinh(\int_s^T \frac{g^2(u)}{2}\mathrm{d}u)\right|}x_s, 2\sinh(\textstyle\int_t^T \frac{g^2(u)}{2}\mathrm{d}u)^2\left\{\coth(\int_s^T \frac{g^2(u)}{2}\mathrm{d}u) - \coth(\int_t^T \frac{g^2(u)}{2}\mathrm{d}u)\right\}\mathrm{I}_d\right)\ . \tag{D.17}$$

Since $p^\star_T(x) = \int_{\mathbb{R}^d} q_{T|0}(x|x_0)\mathrm{d}\pi(x_0)$, it results that $\mathbb{P}^\star_T = \delta_0$.

Proof. The proof of (D.17) directly follows from Lemma 67. □

Following Lemma 72, the POUM scheme is an 'exact' noising scheme and we have $\pi^{\mathrm{base}} = \delta_0$ in this setting. Moreover, under mild assumptions on π, the time-reversed SDE (5.2) writes as

$$\mathrm{d}Y_t = \frac{g^2(T-t)}{2}\{\coth(\textstyle\int_{T-t}^T \frac{g^2(u)}{2}\mathrm{d}u)Y_t + 2\nabla \log p^\star_{T-t}(Y_t)\}\mathrm{d}t + g(T-t)\mathrm{d}B_t,\ Y_0 = 0\ . \tag{D.18}$$

Gaussianized Pinned Brownian Motion. Let $\zeta : [0,T] \to (0,\infty)$ be a square integrable function on $(0,T)$. Denote $\alpha(t) = \int_0^t \zeta(u)\mathrm{d}u$. Inspired by Dai et al. [2023], we propose to consider $f(t) = -\frac{\zeta(t)}{\alpha(T)-\alpha(t)}$ and $g^2(t) = \frac{2\zeta(t)}{\alpha(T)-\alpha(t)}$. Then, SDE (5.1) can be rewritten as

$$\mathrm{d}X_t = -\frac{\zeta(t)X_t}{\alpha(T)-\alpha(t)}\mathrm{d}t + \sqrt{\frac{2\zeta(t)}{\alpha(T)-\alpha(t)}}\mathrm{d}W_t,\ X_0 \sim \pi\ . \tag{D.19}$$

Lemma 73. *Let $\pi \in \mathscr{P}(\mathbb{R}^d)$. Assume that $f(t) = -\frac{\zeta(t)}{\alpha(T)-\alpha(t)}$ and $g^2(t) = \frac{2\zeta(t)}{\alpha(T)-\alpha(t)}$, where $\zeta : [0,T] \to (0,\infty)$ is a square integrable function on $(0,T)$. Then, for any pair of time-steps (s,t) such that $T \geq t > s \geq 0$, the conditional distribution of X_t given $X_s = x_s \in \mathbb{R}^d$ is defined by the Gaussian kernel*

$$q_{t|s}(\cdot|x_s) = \mathrm{N}\left(\frac{\alpha(T)-\alpha(t)}{\alpha(T)-\alpha(s)}x_s, \left(1 - \frac{(\alpha(T)-\alpha(t))^2}{(\alpha(T)-\alpha(s))^2}\right)\mathrm{I}_d\right) . \tag{D.20}$$

Since $p^\star_T(x) = \int_{\mathbb{R}^d} q_{T|0}(x|x_0)\mathrm{d}\pi(x_0)$, it results that $\mathbb{P}^\star_T = \mathrm{N}(0,\mathrm{I}_d)$.

Proof. The proof of (D.20) directly follows from Lemma 67. □

Following Lemma 72, the noising scheme given above is 'exact' with base distribution given by $\pi^{\mathrm{base}} = \mathrm{N}(0,\mathrm{I}_d)$, this setting being quite unusual in diffusion model community. Based on this, we refer to this noising diffusion process as the *Gaussianized Pinned Brownian Motion* (GPBM). Moreover, under mild assumptions on π, the time-reverse SDE (5.2) writes as

$$\mathrm{d}Y_t = \frac{\zeta(T-t)}{\alpha(T)-\alpha(T-t)}\left\{Y_t + 2\nabla \log p^\star_{T-t}(Y_t)\right\}\mathrm{d}t + \sqrt{\frac{2\zeta(T-t)}{\alpha(T)-\alpha(T-t)}}\mathrm{d}B_t,\ Y_0 \sim \mathrm{N}(0,\mathrm{I}_d) . \tag{D.21}$$

D.4 Variational objectives for diffusion-based methods

In this section, we first provide details on the continuous-time framework of variational diffusion-based methods (including RDS) in Appendix D.4.1. Then, we describe in Appendix D.4.2 the RDS discrete-time formulation, related to the variational loss $\mathcal{L}_{\mathrm{RDS}}$ given in (5.7), and compare this framework to discrete-time formulations of previous variational methods.

In the rest of the section, we consider a target distribution $\pi \in \mathscr{P}(\mathbb{R}^d)$ and recall that $\mathbb{P}^\star$ denotes the path measure associated to the noising SDE (5.1) initialized at $X_0 \sim \pi$.

D.4.1 Continuous time formulation

RDS setting

Let $\pi^{\mathrm{ref}} \in \mathscr{P}(\mathbb{R}^d)$. We recall that $\mathbb{P}^{\mathrm{ref}}$ denotes the path measure associated to the noising SDE (5.1) starting at $X_0 \sim \pi^{\mathrm{ref}}$. First note that $\mathbb{P}^\star = \pi \otimes \mathbb{P}^{\mathrm{ref}}_{|0}$. Therefore we have the following relation $\mathbb{P}^\star = \frac{\mathrm{d}\pi}{\mathrm{d}\pi^{\mathrm{ref}}} \cdot \mathbb{P}^{\mathrm{ref}}$, *i.e.*, $\mathbb{P}^\star$ is absolutely continuous with respect to $\mathbb{P}^{\mathrm{ref}}$, with Radon-Nikodym derivative given by $(\mathrm{d}\mathbb{P}^\star/\mathrm{d}\mathbb{P}^{\mathrm{ref}})(X_{[0,T]}) = (\mathrm{d}\pi/\mathrm{d}\pi^{\mathrm{ref}})(X_0)$, for any diffusion process $(X_t)_{t\in[0,T]}$ with *forward* time direction.

Below, we first provide the proof of Proposition 7.

Proof of Proposition 7. Assume that $\mathbb{P}^\star_T = \mathbb{P}^{\mathrm{ref}}_T = \pi^{\mathrm{base}}$ and there exists $\theta^\star \in \Theta$ such that $\phi^{\theta^\star}_t = \phi_t$. Under those two assumptions, it is clear that $(Y^{\theta^\star}_t)_{t\in[0,T]} \sim (\mathbb{P}^\star)^R$; therefore, we have $\mathcal{L}_{\mathrm{LV}}(\theta^\star) = 0$, *i.e.*, $\theta^\star$ achieves optimal solution. We now detail how to obtain (5.6).

Consider the detached diffusion process $(Y^{\hat\theta}_t)_{t\in[0,T]} \sim \mathbb{P}^{\hat\theta}$. Using the fact that $\mathbb{P}^\star = \frac{\mathrm{d}\pi}{\mathrm{d}\pi^{\mathrm{ref}}} \cdot \mathbb{P}^{\mathrm{ref}}$, the log-ratio in the objective (5.5) can be computed as

$$\log \frac{\mathrm{d}\mathbb{P}^\theta}{\mathrm{d}(\mathbb{P}^\star)^R}(Y^{\hat\theta}_{[0,T]}) = \log \frac{\mathrm{d}\mathbb{P}^\theta}{\mathrm{d}\,(\mathbb{P}^{\mathrm{ref}})^R}\left(Y^{\hat\theta}_{[0,T]}\right) + \log \frac{\gamma^{\mathrm{ref}}}{\gamma}\left(Y^{\hat\theta}_T\right) - \log \mathcal{Z}^{\mathrm{ref}} + \log \mathcal{Z} . \tag{D.22}$$

Moreover, by applying Girsanov's theorem to the path measures $(\mathbb{P}^{\mathrm{ref}})^R$ and $\mathbb{P}^\theta$ with the assumption $\mathbb{P}^{\mathrm{ref}}_T = \pi^{\mathrm{base}}$, we have

$$\log \frac{\mathrm{d}\mathbb{P}^\theta}{\mathrm{d}(\mathbb{P}^{\mathrm{ref}})^R}(Y^{\hat\theta}_{[0,T]}) = \int_0^T \frac{g^2(T-t)}{2}\left\|\phi^\theta_{T-t}(Y^{\hat\theta}_t)\right\|^2 \mathrm{d}t + \int_0^T g(T-t)\phi^\theta_{T-t}(Y^{\hat\theta}_t)^\top \mathrm{d}B_t . \tag{D.23}$$

Since the Radon-Nikodym derivative between path measures and their respective time-reversals is the same, we obtain the explicit LV objective combine the previous results to obtain the LV objective (5.6) by combining (D.22) and (D.23). □

In contrast, previous reference-based works from Zhang and Chen [2022], Vargas et al. [2023a] relied on the reverse KL continuous-time objective

$$\mathcal{L}^c_{\mathrm{KL}}(\theta) = \mathrm{KL}(\mathbb{P}^\theta \mid (\mathbb{P}^\star)^R) = \mathbb{E}\left[\log \frac{\mathrm{d}\mathbb{P}^\theta}{\mathrm{d}(\mathbb{P}^\star)^R}\left(Y^\theta_{[0,T]}\right)\right] \ , \ Y^\theta_{[0,T]} \sim \mathbb{P}^\theta \tag{D.24}$$

where the expectation is taken w.r.t. to the parameterized process, not the detached one. In this case too, the objective may be simplified as presented next.

Proposition 74. *Assume there exists $\theta^\star \in \Theta$ such that $\phi^\theta_t = \phi_t$. Then, the loss defined in (D.24) achieves optimal solution at $\theta^\star$ and, setting $\varrho = \log(\gamma^{ref}/\gamma)$, it simplifies as*

$$\mathcal{L}^c_{KL}(\theta) = \mathbb{E}\left[\int_0^T \frac{g^2(T-t)}{2}\left\|\phi^\theta_{T-t}(Y^\theta_t)\right\|^2 \mathrm{d}t + \varrho(Y^\theta_T)\right] \ , \tag{D.25}$$

where the equality holds, up to constants independent of θ.

Proof. For any $\theta \in \Theta$, we have by the KL chain rule

$$\mathcal{L}^c_{\mathrm{KL}}(\theta) = \mathrm{KL}(\pi^{\mathrm{base}} \mid \mathbb{P}^\star_T) + \mathbb{E}_{\pi^{\mathrm{base}}}\left[\mathrm{KL}(\mathbb{P}^\theta_{|0} \mid (\mathbb{P}^\star)^R_{|0})\right] \ , \tag{D.26}$$

where the first term does not depend on θ. Then, it is clear that $\mathcal{L}^c_{\mathrm{KL}}$ is minimized when $\theta = \theta^\star$. The result of (D.25) directly follows from (D.22) and (D.23); in particular, the additional constants are $-\log \mathcal{Z}^{\mathrm{ref}}$ and $\log \mathcal{Z}$. □

Based on the simplification (D.25), one may recognize a standard formulation of stochastic control-affine problem with terminal cost ϱ, where the uncontrolled process is given by $\mathbb{P}^{\mathrm{ref}}$.

DIS setting

In their paper, Berner et al. [2023] seek to approximate the time-reversal of $\mathbb{P}^\star$ for the VP noising scheme, see Appendix D.3.3. To do so, they consider the class of variational path measures $(\mathbb{P}^\theta)_{\theta\in\Theta} \subset \mathscr{P}(\mathbf{C}_T)$ where for any $\theta \in \Theta$, $\mathbb{P}^\theta$ is associated to the SDE

$$\mathrm{d}Y^\theta_t = \frac{g^2(T-t)}{2} Y^\theta_t \mathrm{d}t + \sigma^2 g^2(T-t) s^\theta_{T-t}(Y^\theta_t)\mathrm{d}t + \sigma g(T-t)\mathrm{d}B_t \ , Y^\theta_0 \sim \mathrm{N}(0, \sigma^2 \mathrm{I}_d), \tag{D.27}$$

where the g-schedule is detailed in Appendix D.3.3, and $s^\theta : [0,T] \times \mathbb{R}^d$ is a neural network that aims at approximating the marginal scores of $\mathbb{P}^\star$, *i.e.*, $s^\theta(t,\cdot) \approx \nabla \log p^\star_t$. The resulting variational method, DIS, then consists in minimizing the KL-based objective $\theta \mapsto \mathrm{KL}(\mathbb{P}^\theta \mid (\mathbb{P}^\star)^R)$, later extended by Richter et al. [2023] to the LV-based setting. We refer to [Richter et al., 2023, Section 3.2.] for detailed expressions of these continuous-time losses.

CMCD setting

Alternatively, Vargas et al. [2024] propose to approximate a diffusion process with marginals prescribed by a curve of distributions $(\pi_t)_{t\in[0,T]}$, such that $\pi_0 = \pi^{\mathrm{base}}$ and $\pi_T = \pi$, with unnnormalized densities that are assumed to be tractable. To do so, they consider the class of pairs of variational path measures $\left((\mathbb{P}^\theta, \mathbb{Q}^\theta)\right)_{\theta\in\Theta} \subset \mathscr{P}(\mathbf{C}_T)^2$ where, for any $\theta \in \Theta$, $\mathbb{Q}^\theta$ is associated to the SDE

$$\mathrm{d}X^\theta_t = \frac{\sigma^2}{2}(\nabla \log \pi_{T-t}(X^\theta_t) - h^\theta_{T-t}(X^\theta_t))\mathrm{d}t + \sigma \mathrm{d}W_t, \quad X^\theta_0 \sim \pi \ . \tag{D.28}$$

and $\mathbb{P}^\theta$ is associated to the SDE

$$\mathrm{d}Y^\theta_t = \frac{\sigma^2}{2}(\nabla \log \pi_t(Y^\theta_t) + h^\theta_t(Y^\theta_t))\mathrm{d}t + \sigma \mathrm{d}B_t, \quad Y^\theta_0 \sim \pi^{\mathrm{base}} \ , \tag{D.29}$$

with $h^\theta : [0,T] \times \mathbb{R}^d \to \mathbb{R}^d$ being a neural network. The resulting variational method, CMCD, consists in minimizing the KL-based objective $\theta \mapsto \mathrm{KL}(\mathbb{P}^\theta \mid (\mathbb{Q}^\theta)^R)$ or the LV-based extension. We refer to the original paper for detailed expressions of these continuous-time losses.

D.4.2 Discrete time formulation

Following the setting of Section 5.2.2, we consider a time discretization of the interval $[0, T]$ given by an increasing sequence of timesteps $\{t_k\}_{k=0}^K$ such that $t_0 = 0$[1], $t_K = T$ and $K \geq 1$. In the rest of the section, we denote $\delta_k = t_{k+1} - t_k$.

This section is organized as follows. We first explain in Appendix D.4.3 the discretization setting that we adopt for RDS, before deriving the RDS variational objective obtained by EM integration (Appendix D.4.3), EI integration in the PBM case (Appendix D.4.3) and EI integration in the VP case (Appendix D.4.3). Finally, for comparison purpose, we provide a discrete time approach to DIS in Appendix D.4.4 and CMCD in Appendix D.4.5. All of the presented settings are implemented in our code.

D.4.3 General RDS setting

Let $\theta \in \Theta$ and let $\pi^{\mathrm{ref}} \in \mathscr{P}(\mathbb{R}^d)$ be an arbitrary reference distribution. Consider the variational diffusion process $(Y_t^\theta)_{t\in[0,T]}$ defined by SDE (5.4). We recall that under mild assumptions on π^{ref}, see e.g., Cattiaux et al. [2023], the time-reversal of $\mathbb{P}^{\mathrm{ref}}$ is associated to the SDE

$$\mathrm{d}Y_t^{\mathrm{ref}} = -f(T-t)Y_t^{\mathrm{ref}}\mathrm{d}t + g^2(T-t)s_{T-t}^{\mathrm{ref}}(Y_t^{\mathrm{ref}})\mathrm{d}t + g(T-t)\mathrm{d}B_t \ , \ Y_0^{\mathrm{ref}} \sim \mathbb{P}_T^{\mathrm{ref}} \ . \tag{D.30}$$

Assume that we are given transition kernels $(\mathcal{A}, y_k) \mapsto p_{k+1|k}^\theta(\mathcal{A}|y_k)$ and $(\mathcal{A}, y_k) \mapsto p_{k+1|k}^{\mathrm{ref}}(\mathcal{A}|y_k)$ respectively approximating $\mathbb{P}(Y_{t_{k+1}}^\theta \in \mathcal{A} \mid Y_{t_k}^\theta = y_k)$ and $\mathbb{P}(Y_{t_{k+1}}^{\mathrm{ref}} \in \mathcal{A} \mid Y_{t_k}^{\mathrm{ref}} = y_k)$. Then, for K sufficiently large, the path measures $\mathbb{P}^\theta$ and $(\mathbb{P}^{\mathrm{ref}})^R$ may respectively be approximated by the joint distributions $p_{0:K}^\theta \in \mathscr{P}^{(K+1)}$ and $p_{0:K}^{\mathrm{ref}} \in \mathscr{P}^{(K+1)}$, defined by $p_{0:K}^\theta = \pi^{\mathrm{base}} \prod_{k=0}^{K-1} p_{k+1|k}^\theta$ and $p_{0:K}^{\mathrm{ref}} = \pi^{\mathrm{base}} \prod_{k=0}^{K-1} p_{k+1|k}^{\mathrm{ref}}$. Using the relation $\mathbb{P}^\star = \frac{\mathrm{d}\pi}{\mathrm{d}\pi^{\mathrm{ref}}} \cdot \mathbb{P}^{\mathrm{ref}}$, $(\mathbb{P}^\star)^R$ may also be approximated by the joint distribution $p_{0:K}^\star = \frac{\mathrm{d}\pi}{\mathrm{d}\pi^{\mathrm{ref}}} \cdot p_{0:K}^{\mathrm{ref}} \in \mathscr{P}^{(K+1)}$.

Using this correspondence between path measures and joint distributions, we propose to approximate $\mathcal{L}_{\mathrm{LV}}$ defined in (5.5) by the discrete time LV-based objective

$$\mathcal{L}_{\mathrm{RDS}}(\theta) = \mathrm{Var}\left[\log \frac{\mathrm{d}p_{0:K}^\theta}{\mathrm{d}p_{0:K}^\star}(Y_{0:K}^{\hat\theta})\right] \ , \ Y_{0:K}^{\hat\theta} \sim p_{0:K}^{\hat\theta} \ . \tag{D.31}$$

Similarly, $\mathcal{L}_{\mathrm{KL}}^{\mathrm{c}}$ defined in (D.24) may be approximated by the discrete time KL-based objective

$$\mathcal{L}_{\mathrm{KL}}(\theta) = \mathbb{E}\left[\log \frac{\mathrm{d}p_{0:K}^\theta}{\mathrm{d}p_{0:K}^\star}(Y_{0:K}^{\theta})\right] \ , \ Y_{0:K}^{\theta} \sim p_{0:K}^{\theta} \ . \tag{D.32}$$

By definition of the joint distributions, and similarly to the continuous-time relation , the log-ratio in (D.31) and (D.32) can be simplified as

$$\log \frac{\mathrm{d}p_{0:K}^\theta}{\mathrm{d}p_{0:K}^\star}(Y_{0:K}^{\bar\theta}) = \log \frac{\mathrm{d}p_{0:K}^\theta}{\mathrm{d}p_{0:K}^{\mathrm{ref}}}(Y_{0:K}^{\bar\theta}) + \log \frac{\gamma^{\mathrm{ref}}}{\gamma}(Y_K^{\bar\theta}) + \log \mathcal{Z} - \log \mathcal{Z}^{\mathrm{ref}} \ . \tag{D.33}$$

where $\bar\theta \in \{\theta, \hat\theta\}$. In the following sections, building upon (D.33), we derive the exact expression of the variational loss (D.31) for each combination of noising and discretization schemes. We emphasize that these derivations are in particular applicable to PIS and DDS settings by taking π^{ref} as given in Table 5.1.

EM-based RDS

In this section, we consider an arbitrary noising scheme that fits within the framework detailed in Appendix D.3.1. For any $k \in \{0, \dots, K-1\}$, define the coefficients

$$w_k^{\mathrm{EM}} = g^2(t)T - t_k)\delta_k \ , \tag{D.34}$$
$$a_k^{\mathrm{EM}} = 1 - f(T-t_k)\delta_k \ , \tag{D.35}$$
$$b_k^{\mathrm{EM}} = g^2(t)T - t_k)\delta_k \ , \tag{D.36}$$
$$c_k^{\mathrm{EM}} = g^2(t)T - t_k)\delta_k \ . \tag{D.37}$$

[1] In the only case of PBM, we consider $t_0 > 0$ to avoid numerical integration error (typically $t_0 = 10^{-4}$).

To approximate SDEs (D.30) and (5.4) on time interval $[t_k, t_{k+1}]$, we may define the following transition kernels

$$p^{\mathrm{ref}}_{k+1|k}(\cdot|y_k) = \mathrm{N}(a^{\mathrm{EM}}_k y_k + b^{\mathrm{EM}}_k s^{\mathrm{ref}}_{T-t_k}(y_k), c^{\mathrm{EM}}_k \mathrm{I}_d) \;, \tag{D.38}$$

$$p^{\theta}_{k+1|k}(\cdot|y_k) = \mathrm{N}(a^{\mathrm{EM}}_k y_k + b^{\mathrm{EM}}_k \{s^{\mathrm{ref}}_{T-t_k}(y_k) + \phi^{\theta}_{T-t_k}(y_k)\}, c^{\mathrm{EM}}_k \mathrm{I}_d) \;, \tag{D.39}$$

that can be obtained by applying the EM integration scheme, see Appendix D.2. In this case, the log-ratio $\log \mathrm{d}p^{\theta}/\mathrm{d}p^{\mathrm{ref}}$ can be computed as follows.

Lemma 75. *Assume that the joint distributions p^{θ} and p^{ref} are respectively induced by the transition kernels defined in (D.38) and (D.39). Then, for any $y_{0:K} \in (\mathbb{R}^d)^{K+1}$, we have*

$$\log \frac{\mathrm{d}p^{\theta}}{\mathrm{d}p^{ref}}(y_{0:K}) = \sum_{k=0}^{K-1} \frac{w^{EM}_k}{2} \|\phi^{\theta}_{T-t_k}(y_k)\|^2 + \sum_{k=0}^{K-1} \sqrt{w^{EM}_k} \phi^{\theta}_{T-t_k}(y_k)^{\top} \epsilon_k \;, \tag{D.40}$$

$$\textit{where } \epsilon_k = \frac{1}{\sqrt{c^{EM}_k}} \left(y_{k+1} - a^{EM}_k y_k - b^{EM}_k \{s^{ref}_{T-t_k}(y_k) + \phi^{\theta}_{T-t_k}(y_k)\} \right) \;. \tag{D.41}$$

Proof. In this proof, for ease of reading, we will respectively denote a^{EM}_k, b^{EM}_k, c^{EM}_k and w^{EM}_k by a_k, b_k, c_k and w_k. Let $y_{0:K} \in (\mathbb{R}^d)^{K+1}$. We have

$$\log \frac{\mathrm{d}p^{\theta}}{\mathrm{d}p^{\mathrm{ref}}}(y_{0:K}) = \sum_{k=0}^{K-1} \log \frac{\mathrm{d}p^{\theta}_{k+1|k}(y_{k+1}|y_k)}{\mathrm{d}p^{\mathrm{ref}}_{k+1|k}(y_{k+1}|y_k)} \tag{D.42}$$

$$= \sum_{k=0}^{K-1} \left\{ -\frac{1}{2} \|\epsilon_k\|^2 + \frac{1}{2} \left\| \epsilon_k + (b_k/\sqrt{c_k}) \phi^{\theta}_{T-t_k}(y_k) \right\|^2 \right\} \tag{D.43}$$

$$= \sum_{k=0}^{K-1} \frac{b_k^2}{2c_k} \left\| \phi^{\theta}_{T-t_k}(y_k) \right\|^2 + \sum_{k=0}^{K-1} \frac{b_k}{\sqrt{c_k}} \phi^{\theta}_{T-t_k}(y_k)^{\top} \epsilon_k \;, \tag{D.44}$$

where ϵ_k is defined by (D.41). Lemma 75 directly follows by noting that $w_k = b_k^2/c_k$. □

By combining Lemma 75 with (D.33), we deduce the expression of the RDS loss in this setting.

Corollary 76 (Expression of RDS loss – EM setting)**.** *When using the EM scheme to integrate the SDE (5.4), the RDS loss simplifies as*

$$\mathcal{L}_{RDS}(\theta) = \mathrm{Var} \left[\sum_{k=0}^{K-1} w^{EM}_k \phi^{\theta}_{T-t_k}(Y_k)^{\top} \{ \phi^{\hat{\theta}}_{T-t_k}(Y_k) - \frac{1}{2} \phi^{\theta}_{T-t_k}(Y_k) \} \right. \tag{D.45}$$

$$\left. + \sum_{k=0}^{K-1} \sqrt{w^{EM}_k} \phi^{\theta}_{T-t_k}(Y_k)^{\top} Z_k + \log \frac{\gamma^{ref}}{\gamma}(Y_K) \right] \;, \tag{D.46}$$

where $\{Z_k\}_{k=0}^{K-1}$ are independently distributed according to $\mathrm{N}(0, \mathrm{I}_d)$, and $\{Y_k\}_{k=0}^{K}$ is recursively defined by $Y_0 \sim \pi^{base}$, and for any $k \in \{0, \dots, K-1\}$,

$$Y_{k+1} = a^{EM}_k Y_k + b^{EM}_k \{ s^{ref}_{T-t_k}(Y_k) + \phi^{\hat{\theta}}_{T-t_k}(Y_k) \} + \sqrt{c^{EM}_k} Z_k \;, \tag{D.47}$$

with $\hat{\theta}$ being a detached version of θ.

Proof. Consider the discrete time process $\{Y_k\}_{k=0}^{K}$ defined by recursion (D.47). With the same notation as in Lemma 75, we have for any $k \in \{0, \dots, K-1\}$

$$\epsilon_k = Z_k + \sqrt{w^{\mathrm{EM}}_k} \{ \phi^{\hat{\theta}}_{T-t_k}(Y_k) - \phi^{\theta}_{T-t_k}(Y_k) \} \;, \tag{D.48}$$

recalling that $w^{\mathrm{EM}}_k = (b^{\mathrm{EM}}_k)^2/c^{\mathrm{EM}}_k$. Using (D.33), we obtain the expression of the RDS loss. □

For completeness, we also provide a similar result for the KL-based objective given in (D.32).

Corollary 77 (Expression of reverse KL loss – EM setting)**.** *When using the EM scheme to integrate the SDE (5.4), up to additional constants independent of θ, the discrete time KL loss simplifies as*

$$\mathcal{L}_{KL}(\theta) = \mathbb{E}\left[\sum_{k=0}^{K-1} \frac{w_k^{EM}}{2} \left\|\phi^\theta_{T-t_k}(Y_k^\theta)\right\|^2 + \log \frac{\gamma^{ref}}{\gamma}(Y_K^\theta)\right] , \tag{D.49}$$

where $\{Z_k\}_{k=0}^{K-1}$ are independently distributed according to $\mathrm{N}(0, \mathrm{I}_d)$, and $\{Y_k^\theta\}_{k=0}^K$ is recursively defined by $Y_0^\theta \sim \pi^{base}$, and for any $k \in \{0, \dots, K-1\}$

$$Y_{k+1}^\theta = a_k^{EM} Y_k^\theta + b_k^{EM}\{s^{ref}_{T-t_k}(Y_k^\theta) + \phi^\theta_{T-t_k}(Y_k^\theta)\} + \sqrt{c_k^{EM}} Z_k \, . \tag{D.50}$$

As observed by Vargas et al. [2023a], there may be a significant bias raising from the use of the Euler-Maruyama discretization when using diffusion-based sampling methods. This motivates us to consider EI discretization in the RDS setting.

EI-based RDS (PBM)

We first provide the expression of the RDS loss in the case of the PBM noising scheme, which is detailed in Appendix D.3.2. We recall the notation $\alpha(t) = \int_0^t g^2(u)\mathrm{d}u$. For any $k \in \{0, \dots, K-1\}$, define the coefficients

$$w_k^{\mathrm{PBM}} = \frac{(\alpha(T) - \alpha(T-t_k))(\alpha(T-t_k) - \alpha(T-t_{k+1}))}{\alpha(T) - \alpha(T-t_{k+1})} \, , \tag{D.51}$$

$$a_k^{\mathrm{PBM}} = \frac{\alpha(T) - \alpha(T-t_{k+1})}{\alpha(T) - \alpha(T-t_k)} \, , \tag{D.52}$$

$$b_k^{\mathrm{PBM}} = \alpha(T-t_k) - \alpha(T-t_{k+1}) \, , \tag{D.53}$$

$$c_k^{\mathrm{PBM}} = \frac{(\alpha(T) - \alpha(T-t_{k+1}))(\alpha(T-t_k) - \alpha(T-t_{k+1}))}{\alpha(T) - \alpha(T-t_k)} \, . \tag{D.54}$$

To approximate SDEs (D.30) and (5.4) on time interval $[t_k, t_{k+1}]$, we may define the following transition kernels

$$p^{\mathrm{ref}}_{k+1|k}(\cdot|y_k) = \mathrm{N}(a_k^{\mathrm{PBM}} y_k + b_k^{\mathrm{PBM}} s^{\mathrm{ref}}_{T-t_k}(y_k), c_k^{\mathrm{PBM}} \mathrm{I}_d) \, , \tag{D.55}$$

$$p^{\theta}_{k+1|k}(\cdot|y_k) = \mathrm{N}(a_k^{\mathrm{PBM}} y_k + b_k^{\mathrm{PBM}} \{s^{\mathrm{ref}}_{T-t_k}(y_k) + \phi^\theta_{T-t_k}(y_k)\}, c_k^{\mathrm{PBM}} \mathrm{I}_d) \, , \tag{D.56}$$

that can be obtained by applying the EI scheme, see Appendix D.2. In this case, the log ratio $\log \mathrm{d}p^\theta / \mathrm{d}p^{\mathrm{ref}}$ can be computed as follows.

Lemma 78. *Assume that the joint distributions p^θ and p^{ref} are respectively induced by the transition kernels defined in (D.55) and (D.56). Then for any $y_{0:K} \in (\mathbb{R}^d)^{K+1}$, we have*

$$\log \frac{\mathrm{d}p^\theta}{\mathrm{d}p^{ref}}(y_{0:K}) = \sum_{k=0}^{K-1} \frac{w_k^{PBM}}{2} \|\phi^\theta_{T-t_k}(y_k)\|^2 + \sum_{k=0}^{K-1} \sqrt{w_k^{PBM}} \phi^\theta_{T-t_k}(y_k)^\top \epsilon_k \, , \tag{D.57}$$

$$\text{where } \epsilon_k = \frac{1}{\sqrt{c_k^{PBM}}} \left(y_{k+1} - a_k^{PBM} y_k - b_k^{PBM}\{s^{ref}_{T-t_k}(y_k) + \phi^\theta_{T-t_k}(y_k)\}\right) \, . \tag{D.58}$$

Proof. The proof is exactly the same as Lemma 75, noting that we still have $w_k^{\mathrm{PBM}} = (b_k^{\mathrm{PBM}})^2 / c_k^{\mathrm{PBM}}$. □

By combining Lemma 78 with (D.33), we deduce the expression of the RDS loss in this setting. The proof of the following is the same as in Corollary 76.

Corollary 79 (Expression of RDS loss – EI setting – PBM scheme)**.** *When using the EI scheme to integrate SDE (5.4) implemented with PBM, the RDS loss simplifies as*

$$\mathcal{L}_{RDS}(\theta) = \mathrm{Var}\left[\sum_{k=0}^{K-1} w_k^{PBM} \phi^\theta_{T-t_k}(Y_k)^\top \{\phi^{\hat\theta}_{T-t_k}(Y_k) - \frac{1}{2}\phi^\theta_{T-t_k}(Y_k)\}\right. \tag{D.59}$$

$$\left. + \sum_{k=0}^{K-1} \sqrt{w_k^{PBM}} \phi^\theta_{T-t_k}(Y_k)^\top Z_k + \log \frac{\gamma^{ref}}{\gamma}(Y_K)\right] \, , \tag{D.60}$$

where $\{Z_k\}_{k=0}^{K-1}$ *are independently distributed according to* $\mathrm{N}(0, \mathrm{I}_d)$*, and* $\{Y_k\}_{k=0}^{K}$ *is recursively defined by* $Y_0 \sim \pi^{base}$*, and for any* $k \in \{0, \dots, K-1\}$

$$Y_{k+1} = a_k^{PBM} Y_k + b_k^{PBM}\{s^{ref}_{T-t_k}(Y_k) + \phi^{\hat{\theta}}_{T-t_k}(Y_k)\} + \sqrt{c_k^{PBM}} Z_k \;, \tag{D.61}$$

with $\hat{\theta}$ *being a detached version of* θ*.*

For completeness, we also provide a similar result for the KL-based objective given in (D.32).

Corollary 80 (Expression of reverse KL loss – EI setting – PBM noising)**.** *When using the EI scheme to integrate SDE (5.4) implemented with PBM, up to additional constants independent of* θ*, the discrete time KL loss simplifies as*

$$\mathcal{L}_{KL}(\theta) = \mathbb{E}\left[\sum_{k=0}^{K-1} \frac{w_k^{PBM}}{2} \left\|\phi^{\theta}_{T-t_k}(Y_k^{\theta})\right\|^2 + \log \frac{\gamma^{ref}}{\gamma}(Y_K^{\theta})\right] \;, \tag{D.62}$$

where $\{Z_k\}_{k=0}^{K-1}$ *are independently distributed according to* $\mathrm{N}(0, \mathrm{I}_d)$*, and* $\{Y_k^{\theta}\}_{k=0}^{K}$ *is recursively defined by* $Y_0^{\theta} \sim \pi^{base}$*, and for any* $k \in \{0, \dots, K-1\}$

$$Y_{k+1}^{\theta} = a_k^{PBM} Y_k^{\theta} + b_k^{PBM}\{s^{ref}_{T-t_k}(Y_k^{\theta}) + \phi^{\theta}_{T-t_k}(Y_k^{\theta})\} + \sqrt{c_k^{PBM}} Z_k \;. \tag{D.63}$$

EI-based RDS (VP)

Here, we provide the expression of the RDS loss in the case of the VP noising scheme, which is detailed in Appendix D.3.3. We recall the notation $\alpha(t) = \int_0^t g^2(u)\mathrm{d}u$. For any $k \in \{0, \dots, K-1\}$, define $\lambda_k = \exp\left((\alpha(T-t_k) - \alpha(T-t_{k+1})\right) - 1$ and the coefficients

$$w_k^{\mathrm{VP}} = \frac{4\sigma^2(\sqrt{1+\lambda_k}-1)^2}{\lambda_k} = 4\sigma^2 \tanh\left(\frac{\alpha(T-t_k) - \alpha(T-t_{k+1})}{4}\right) \tag{D.64}$$

$$a_k^{\mathrm{VP}} = \sqrt{1+\lambda_k} \tag{D.65}$$

$$b_k^{\mathrm{VP}} = 2\sigma^2\{\sqrt{1+\lambda_k} - 1\} \tag{D.66}$$

$$c_k^{\mathrm{VP}} = \sigma^2 \lambda_k \tag{D.67}$$

$$\tag{D.68}$$

To approximate SDEs (D.30) and (5.4) on time interval $[t_k, t_{k+1}]$, we may define the following transition kernels

$$p^{\mathrm{ref}}_{k+1|k}(\cdot|y_k) = \mathrm{N}(a_k^{\mathrm{VP}} y_k + b_k^{\mathrm{VP}} s^{\mathrm{ref}}_{T-t_k}(y_k), c_k^{\mathrm{VP}} \mathrm{I}_d) \;, \tag{D.69}$$

$$p^{\theta}_{k+1|k}(\cdot|y_k) = \mathrm{N}(a_k^{\mathrm{VP}} y_k + b_k^{\mathrm{VP}}\{s^{\mathrm{ref}}_{T-t_k}(y_k) + \phi^{\theta}_{T-t_k}(y_k)\}, c_k^{\mathrm{VP}} \mathrm{I}_d) \;, \tag{D.70}$$

that can be obtained by applying the EI scheme, see Appendix D.2. In this case, the log ratio $\log \mathrm{d}p^{\theta}/\mathrm{d}p^{\mathrm{ref}}$ can be computed as follows.

Lemma 81. *Assume that the joint distributions* p^{θ} *and* p^{ref} *are respectively induced by the transition kernels defined in (D.69) and (D.69). Then, for any* $y_{0:K} \in (\mathbb{R}^d)^{K+1}$*, we have*

$$\log \frac{\mathrm{d}p^{\theta}}{\mathrm{d}p^{ref}}(y_{0:K}) = \sum_{k=0}^{K-1} \frac{w_k^{VP}}{2} \|\phi^{\theta}_{T-t_k}(y_k)\|^2 + \sum_{k=0}^{K-1} \sqrt{w_k^{VP}} \phi^{\theta}_{T-t_k}(y_k)^{\top} \epsilon_k \tag{D.71}$$

$$\text{where } \epsilon_k = \frac{1}{\sqrt{c_k^{VP}}}\left(y_{k+1} - a_k^{VP} y_k - b_k^{VP}\{s^{ref}_{T-t_k}(y_k) + \phi^{\theta}_{T-t_k}(y_k)\}\right) \tag{D.72}$$

Proof. The proof is exactly the same as Lemma 75, noting that we still have $w_k^{\mathrm{VP}} = (b_k^{\mathrm{VP}})^2/c_k^{\mathrm{VP}}$. □

By combining Lemma 81 with (D.33), we deduce the expression of the RDS loss in this setting. The proof of the following is the same as in Corollary 76.

Corollary 82 (Expression of RDS loss – EI setting - VP scheme)**.** *When using the EI scheme to integrate SDE* (5.4) *implemented with VP , the RDS loss simplifies as*

$$\mathcal{L}_{RDS}(\theta) = \mathrm{Var}\left[\sum_{k=0}^{K-1} w_k^{VP} \phi^\theta_{T-t_k}(Y_k)^\top \{\phi^{\hat\theta}_{T-t_k}(Y_k) - \frac{1}{2}\phi^\theta_{T-t_k}(Y_k)\}\right. \tag{D.73}$$

$$\left. + \sum_{k=0}^{K-1} \sqrt{w_k^{VP}} \phi^\theta_{T-t_k}(Y_k)^\top Z_k + \log \frac{\gamma^{ref}}{\gamma}(Y_K)\right] , \tag{D.74}$$

where $\{Z_k\}_{k=0}^{K-1}$ *are independently distributed according to* $\mathrm{N}(0, \mathrm{I}_d)$*, and* $\{Y_k\}_{k=0}^{K}$ *is recursively defined by* $Y_0 \sim \pi^{base}$*, and for any* $k \in \{0, \dots, K-1\}$

$$Y_{k+1} = a_k^{VP} Y_k + b_k^{VP}\{s^{ref}_{T-t_k}(Y_k) + \phi^{\hat\theta}_{T-t_k}(Y_k)\} + \sqrt{c_k^{VP}} Z_k \, , \tag{D.75}$$

with $\hat\theta$ *being a detached version of* θ*.*

For completeness, we also provide a similar result for the KL-based objective given in (D.32).

Corollary 83 (Expression of reverse KL loss – EI setting - VP noising)**.** *When using the EI scheme to integrate SDE* (5.4) *implemented with VP, up to additional constants independent of* θ*, the discrete time KL loss simplifies as*

$$\mathcal{L}_{KL}(\theta) = \mathbb{E}\left[\sum_{k=0}^{K-1} \frac{w_k^{VP}}{2} \left\|\phi^\theta_{T-t_k}(Y_k^\theta)\right\|^2 + \log \frac{\gamma^{ref}}{\gamma}(Y_K^\theta)\right] , \tag{D.76}$$

where $\{Z_k\}_{k=0}^{K-1}$ *are independently distributed according to* $\mathrm{N}(0, \mathrm{I}_d)$*, and* $\{Y_k^\theta\}_{k=0}^{K}$ *is recursively defined by* $Y_0^\theta \sim \pi^{base}$*, and for any* $k \in \{0, \dots, K-1\}$*,*

$$Y^\theta_{k+1} = a_k^{VP} Y_k^\theta + b_k^{VP}\{s^{ref}_{T-t_k}(Y_k^\theta) + \phi^\theta_{T-t_k}(Y_k^\theta)\} + \sqrt{c_k^{VP}} Z_k \, . \tag{D.77}$$

D.4.4 DIS setting

Here, we propose a discrete time alternative to the continuous time objective proposed by Berner et al. [2023] and Richter et al. [2023]. Let $\theta \in \Theta$. Consider the variational diffusion process $(Y_t^\theta)_{t\in[0,T]}$ defined by SDE (D.27).

Assume that we are given transition kernels $(\mathcal{A}, y_k) \mapsto p^\theta_{k+1|k}(\mathcal{A}|y_k)$ and $(\mathcal{A}, y_{k+1}) \mapsto p^\star_{k|k+1}(\mathcal{A}|y_{k+1})$ respectively approximating $\mathbb{P}(Y^\theta_{t_{k+1}} \in \mathcal{A} \mid Y^\theta_{t_k} = y_k)$ and $\mathbb{P}(X_{t_k} \in \mathcal{A} \mid X_{t_{k+1}} = y_{k+1})$. Then, for K sufficiently large, the path measures $\mathbb{P}^\theta$ and $(\mathbb{P}^\star)^R$ may be approximated by the joint distributions $p^\theta_{0:K} \in \mathscr{P}^{(K+1)}$ and $p^\star_{0:K} \in \mathscr{P}^{(K+1)}$ defined by $p^\theta_{0:K} = \pi^{\text{base}} \prod_{k=0}^{K-1} p^\theta_{k+1|k}$ and $p^\star_{0:K} = \pi \prod_{k=0}^{K-1} p^\star_{k|k+1}$.

Following the RDS methodology, we propose to approximate the KL-based continuous-time DIS objective by

$$\mathcal{L}^{\text{DIS}}_{\text{KL}}(\theta) = \mathbb{E}\left[\log \frac{\mathrm{d}p^\theta_{0:K}}{\mathrm{d}p^\star_{0:K}}(Y^\theta_{0:K})\right] \, , \; Y^\theta_{0:K} \sim p^\theta_{0:K} \, , \tag{D.78}$$

and the LV-based continuous-time DIS objective by

$$\mathcal{L}^{\text{DIS}}_{\text{LV}}(\theta) = \mathrm{Var}\left[\log \frac{\mathrm{d}p^\theta_{0:K}}{\mathrm{d}p^\star_{0:K}}(Y^{\hat\theta}_{0:K})\right] \, , \; Y^{\hat\theta}_{0:K} \sim p^{\hat\theta}_{0:K}. \tag{D.79}$$

To approximate SDEs (D.27) and (5.1) on time interval $[t_k, t_{k+1}]$ in the case of the VP noising scheme, we may define the following transition kernels

$$p^\star_{k|k+1}(\cdot|y_{k+1}) = \mathrm{N}(\sqrt{1 - \lambda_{T-t_{k+1},T-t_k}} y_{k+1}, \sigma^2 \lambda_{T-t_{k+1},T-t_k} \, \mathrm{I}_d) \, , \tag{D.80}$$

$$p^\theta_{k+1|k}(\cdot|y_k) = \mathrm{N}(a_k^{\text{VP}} y_k + b_k^{\text{VP}} + s^\theta_{T-t_k}(y_k), c_k^{\text{VP}} \mathrm{I}_d) \, , \tag{D.81}$$

where $\lambda_{s,t}$ is defined in Lemma 71 with $s = T - t_{k+1}$ and $t = T - t_k$, and $\{a_k^{\mathrm{VP}}, b_k^{\mathrm{VP}}, c_k^{\mathrm{VP}}\}$ are defined via the VP-EI integration scheme detailed in Appendix D.4.3. In particular, the following equalities hold:

$$\sqrt{1 - \lambda_{T-t_{k+1},T-t_k}} = \exp\left(\frac{1}{2}[\alpha(T-t_{k+1}) - \alpha(T-t_k)]\right) = \frac{1}{a_k^{\mathrm{VP}}}\,, \tag{D.82}$$

$$\sigma^2 \lambda_{T-t_{k+1},T-t_k} = \sigma^2[1 - \exp\left(\alpha(T-t_{k+1}) - \alpha(T-t_k)\right)] = \frac{c_k^{\mathrm{VP}}}{(a_k^{\mathrm{VP}})^2}\,. \tag{D.83}$$

Therefore, we have $p^\star_{k|k+1}(y_k|y_{k+1}) = \mathrm{N}(y_{k+1}; a_k^{\mathrm{VP}} y_k, c_k^{\mathrm{VP}}\,\mathrm{I}_d)$. In this case, the log-ratio $\log \mathrm{d}p^\theta/\mathrm{d}p^\star$ can be computed as follows.

Lemma 84. *Assume that the joint distributions $p^\star$ and p^θ are respectively induced by the transition kernels defined in (D.80) and (D.81). Then, for any $y_{0:K} \in (\mathbb{R}^d)^{K+1}$, we have*

$$\log \frac{\mathrm{d}p^\theta}{\mathrm{d}p^\star}(y_{0:K}) = \log Z + \log \frac{\pi^{base}(y_0)}{\gamma(y_K)} + \sum_{k=0}^{K-1} \log \frac{\mathrm{d}p^\theta_{k+1|k}(y_{k+1}|y_k)}{\mathrm{d}p^\star_{k|k+1}(y_k|y_{k+1})} \tag{D.84}$$

$$= \log Z + \log \frac{\pi^{base}(y_0)}{\gamma(y_K)} + \sum_{k=0}^{K-1} \frac{w_k^{VP}}{2} \|s^\theta_{T-t_k}(y_k)\|^2 + \sum_{k=0}^{K-1} \sqrt{w_k^{VP}} s^\theta_{T-t_k}(y_k)^\top \epsilon_k \tag{D.85}$$

$$\text{where } \epsilon_k = \frac{1}{\sqrt{c_k^{VP}}}\left(y_{k+1} - a_k^{VP} y_k - b_k^{VP} s^\theta_{T-t_k}(y_k)\right) \tag{D.86}$$

Proof. This result lies on the following identity

$$\frac{\mathrm{d}p^\theta_{k+1|k}(y_{k+1}|y_k)}{\mathrm{d}p^\star_{k|k+1}(y_k|y_{k+1})} = -\frac{1}{2}\|\epsilon_k\|^2 + \frac{1}{2}\left\|\epsilon_k + \sqrt{w_k^{\mathrm{VP}}} s^\theta_{T-t_k}(y_k)\right\|^2 \tag{D.87}$$

□

Based on Lemma 84, we simply deduce the expression of the discrete time versions of LV-based and KL-based DIS losses.

Corollary 85 (Expression of LV-DIS loss)**.** *When using the EI scheme to integrate SDE (D.27), the LV-based discrete time DIS loss (D.79) simplifies as*

$$\mathcal{L}_{LV}^{DIS}(\theta) = \mathrm{Var}\left[\sum_{k=0}^{K-1} w_k^{VP} s^\theta_{T-t_k}(Y_k)^\top \{s^{\hat\theta}_{T-t_k}(Y_k) - \frac{1}{2} s^\theta_{T-t_k}(Y_k)\}\right. \tag{D.88}$$

$$\left. + \sum_{k=0}^{K-1} \sqrt{w_k^{VP}} s^\theta_{T-t_k}(Y_k)^\top Z_k + \log \frac{\pi^{base}(Y_0)}{\gamma(Y_K)}\right]\,, \tag{D.89}$$

where $\{Z_k\}_{k=0}^{K-1}$ are independently distributed according to $\mathrm{N}(0, \mathrm{I}_d)$, and $\{Y_k\}_{k=0}^K$ is recursively defined by $Y_0 \sim \pi^{base}$, and for any $k \in \{0, \dots, K-1\}$,

$$Y_{k+1} = a_k^{VP} Y_k + b_k^{VP} s^{\hat\theta}_{T-t_k}(Y_k) + \sqrt{c_k^{VP}} Z_k\,, \tag{D.90}$$

with $\hat\theta$ being a detached version of θ.

Proof. This result is an application from Lemma 84, where we have $\epsilon_k = Z_k + \sqrt{w_k^{\mathrm{VP}}}\{s^{\hat\theta}_{T-t_k}(Y_k) - s^\theta_{T-t_k}(Y_k)\}$ for any $k \in \{0, \dots, K-1\}$. □

Corollary 86 (Expression of KL-DIS loss)**.** *When using the EM scheme to integrate SDE (D.27), up to additional constants independent of θ, the KL-based discrete time DIS loss (D.78) simplifies as*

$$\mathcal{L}_{KL}^{DIS}(\theta) = \mathbb{E}\left[\sum_{k=0}^{K-1} \frac{w_k^{VP}}{2} \left\|s^\theta_{T-t_k}(Y_k^\theta)\right\|^2 + \log \frac{\pi^{base}(Y_0^\theta)}{\gamma(Y_K^\theta)}\right]\,, \tag{D.91}$$

where $\{Z_k\}_{k=0}^{K-1}$ *are independently distributed according to* $\mathrm{N}(0, \mathrm{I}_d)$*, and* $\{Y_k^\theta\}_{k=0}^K$ *is recursively defined by* $Y_0^\theta \sim \pi^{base}$*, and for any* $k \in \{0, \ldots, K-1\}$*,*

$$Y_{k+1}^\theta = a_k^{VP} Y_k^\theta + b_k^{VP} s_{T-t_k}^\theta(Y_k^\theta) + \sqrt{c_k^{VP}} Z_k \ . \tag{D.92}$$

D.4.5 CMCD setting

Here, we propose a discrete time alternative to the continuous time objective proposed by Vargas et al. [2024]. Let $\theta \in \Theta$. Consider the variational diffusion processes $(Y_t^\theta)_{t\in[0,T]}$ defined by SDE (D.29), and $(X_t^\theta)_{t\in[0,T]}$, defined by (D.28).

Assume that we are given transition kernels $(\mathcal{A}, y_k) \mapsto p_{k+1|k}^\theta(\mathcal{A}|y_k)$ and $(\mathcal{A}, y_{k+1}) \mapsto q_{k|k+1}^\theta(\mathcal{A}|y_{k+1})$ respectively approximating $\mathbb{P}(Y_{t_{k+1}}^\theta \in \mathcal{A} \mid Y_{t_k}^\theta = y_k)$ and $\mathbb{Q}^\theta(X_{t_k}^\theta \in \mathcal{A} \mid X_{t_{k+1}}^\theta = y_{k+1})$. Then, for K sufficiently large, the path measures $\mathbb{P}^\theta$ and $(\mathbb{Q}^\theta)^R$ may be approximated by the joint distributions $p_{0:K}^\theta \in \mathscr{P}^{(K+1)}$ and $q_{0:K}^\theta \in \mathscr{P}^{(K+1)}$ defined by $p_{0:K}^\theta = \pi^{\mathrm{base}} \prod_{k=0}^{K-1} p_{k+1|k}^\theta$ and $q_{0:K}^\theta = \pi \prod_{k=0}^{K-1} q_{k|k+1}^\theta$.

Following the RDS methodology, we propose to approximate the KL-based continuous-time CMCD objective by

$$\mathcal{L}_{\mathrm{KL}}^{\mathrm{CMCD}}(\theta) = \mathbb{E}\left[\log \frac{\mathrm{d}p_{0:K}^\theta}{\mathrm{d}q_{0:K}^\theta}(Y_{0:K}^\theta)\right] \ , \ Y_{0:K}^\theta \sim p_{0:K}^\theta \ , \tag{D.93}$$

and the LV-based continuous-time CMCD objective by

$$\mathcal{L}_{\mathrm{LV}}^{\mathrm{CMCD}}(\theta) = \mathrm{Var}\left[\log \frac{\mathrm{d}p_{0:K}^\theta}{\mathrm{d}q_{0:K}^\theta}(Y_{0:K}^{\hat\theta})\right] \ , \ Y_{0:K}^{\hat\theta} \sim p_{0:K}^{\hat\theta} \ . \tag{D.94}$$

For any $k \in \{0, \ldots, K-1\}$, let us define $w_k^{\mathrm{CMCD}} = \sigma^2 \delta_k / 4$.

To approximate SDEs (D.28) and (D.29) on time interval $[t_k, t_{k+1}]$, we may define the following transition kernels

$$q_{k|k+1}^\theta(\cdot|y_{k+1}) = \mathrm{N}(y_{k+1} + \frac{\sigma^2\delta_k}{2}(\nabla \log \pi_{t_{k+1}}(y_{k+1}) - h_{t_{k+1}}^\theta(y_{k+1})), \sigma^2\delta_k \, \mathrm{I}_d) \ , \tag{D.95}$$

$$p_{k+1|k}^\theta(\cdot|y_k) = \mathrm{N}(y_k + \frac{\sigma^2\delta_k}{2}(\nabla \log \pi_{t_k}(y_k) + h_{t_k}^\theta(y_k)), \sigma^2\delta_k \, \mathrm{I}_d) \ , \tag{D.96}$$

that can be obtained by applying the EM integration scheme, see Appendix D.2. In this case, the log-ratio $\log \mathrm{d}p^\theta / \mathrm{d}q^\theta$ can be computed as follows.

Lemma 87. *Assume that the joint distributions* q^θ *and* p^θ *are respectively induced by the transition kernels defined in* (D.95) *and* (D.96)*. Then, for any* $y_{0:K} \in (\mathbb{R}^d)^{K+1}$*, we have*

$$\log \frac{\mathrm{d}p^\theta}{\mathrm{d}q^\theta}(y_{0:K}) = \log Z + \log \frac{\pi^{base}(y_0)}{\gamma(y_K)} + \sum_{k=0}^{K-1} \log \frac{\mathrm{d}p_{k+1|k}^\theta(y_{k+1}|y_k)}{\mathrm{d}q_{k|k+1}^\theta(y_k|y_{k+1})} \tag{D.97}$$

$$= \log Z + \log \frac{\pi^{base}(y_0)}{\gamma(y_K)} + \frac{1}{2}\sum_{k=0}^{K-1} w_k^{CMCD} \left\|u_k\right\|^2 + \sum_{k=0}^{K-1} \sqrt{w_k^{CMCD}} u_k^\top \epsilon_k \ , \tag{D.98}$$

where

$$u_k = \nabla \log \pi_{t_k}(y_k) + \nabla \log \pi_{t_{k+1}}(y_{k+1}) + h_{t_k}^\theta(y_k) - h_{t_{k+1}}^\theta(y_{k+1}) \ , \tag{D.99}$$

$$\epsilon_k = \frac{1}{\sigma\sqrt{\delta_k}}\left(y_{k+1} - y_k - \frac{\sigma^2\delta_k}{2}\{\nabla \log \pi_{t_k}(y_k) + h_{t_k}^\theta(y_k)\}\right) \ . \tag{D.100}$$

Proof. Based on the notation of u_k and ϵ_k, we obtain the result by noting that

$$\log \frac{\mathrm{d}p_{k+1|k}^\theta(y_{k+1}|y_k)}{\mathrm{d}q_{k|k+1}^\theta(y_k|y_{k+1})} = -\frac{1}{2}\left\|\epsilon_k\right\|^2 + \frac{1}{2}\left\|-\epsilon_k - \sqrt{w_k^{\mathrm{CMCD}}} u_k\right\|^2 \tag{D.101}$$

□

Based on Lemma 87, we simply deduce the expression of the discrete time versions of LV-based and KL-based CMCD losses.

Corollary 88 (Expression of LV-CMCD loss)**.** *When using the EM scheme to integrate SDE* (D.29)*, the LV-based discrete time CMCD loss* (D.94) *simplifies as*

$$\mathcal{L}_{LV}^{CMCD}(\theta) = \mathrm{Var}\left[\log \frac{\pi^{base}(Y_0)}{\gamma(Y_K)} + \sum_{k=0}^{K-1} w_k^{CMCD}\left\{\frac{1}{2}\left\|u_k^\theta\right\|^2 + (h_{t_k}^{\hat{\theta}}(Y_k) - h_{t_k}^\theta(Y_k))^\top u_k^\theta\right\}\right. \tag{D.102}$$

$$\left. + \sum_{k=0}^{K-1} \sqrt{w_k^{CMCD}}(u_k^\theta)^\top Z_k\right] , \tag{D.103}$$

$$\text{with } u_k^\theta = \nabla \log \pi_{t_k}(Y_k) + \nabla \log \pi_{t_{k+1}}(Y_{k+1}) + h_{t_k}^\theta(Y_k) - h_{t_{k+1}}^\theta(Y_{k+1}) , \tag{D.104}$$

where $\{Z_k\}_{k=0}^{K-1}$ *are independently distributed according to* $\mathrm{N}(0, \mathrm{I}_d)$*, and* $\{Y_k\}_{k=0}^K$ *is recursively defined by* $Y_0 \sim \pi^{base}$*, and for any* $k \in \{0, \dots, K-1\}$*,*

$$Y_{k+1} = Y_k + \frac{\sigma^2 \delta_k}{2}\{\nabla \log \pi_{t_k}(Y_k) + h_{t_k}^{\hat{\theta}}(Y_k)\} + \sigma\sqrt{\delta_k} Z_k . \tag{D.105}$$

Proof. This result is an application from Lemma 87, where we have $\epsilon_k = Z_k + \sqrt{w_k^{\mathrm{CMCD}}}\{h_{t_k}^{\hat{\theta}}(Y_k) - h_{t_k}^\theta(Y_k)\}$ for any $k \in \{0, \dots, K-1\}$. □

Corollary 89 (Expression of KL-CMCD loss)**.** *When using the EM scheme to integrate SDE* (D.29)*, up to additional constants independent of* θ*, the KL-based discrete time CMCD loss* (D.93) *simplifies as*

$$\mathcal{L}_{KL}^{CMCD}(\theta) = \mathbb{E}\left[\log \frac{\pi^{base}(Y_0^\theta)}{\gamma(Y_K^\theta)} + \frac{1}{2}\sum_{k=0}^{K-1} w_k^{CMCD}\left\|u_k^\theta\right\|^2\right] , \tag{D.106}$$

$$\text{with } u_k^\theta = \nabla \log \pi_{t_k}(Y_k^\theta) + \nabla \log \pi_{t_{k+1}}(Y_{k+1}^\theta) + h_{t_k}^\theta(Y_k^\theta) - h_{t_{k+1}}^\theta(Y_{k+1}^\theta) , \tag{D.107}$$

where $\{Z_k\}_{k=0}^{K-1}$ *are independently distributed according to* $\mathrm{N}(0, \mathrm{I}_d)$*, and* $\{Y_k^\theta\}_{k=0}^K$ *is recursively defined by* $Y_0^\theta \sim \pi^{base}$*, and for any* $k \in \{0, \dots, K-1\}$*,*

$$Y_{k+1}^\theta = Y_k^\theta + \frac{\sigma^2 \delta_k}{2}\{\nabla \log \pi_{t_k}(Y_k^\theta) + h_{t_k}^\theta(Y_k^\theta)\} + \sigma\sqrt{\delta_k} Z_k . \tag{D.108}$$

D.5 Annealed MCMC samplers

In this section, we present a line of Monte Carlo algorithms that aim at sampling from a sequence of distributions $\{p_k\}_{k=0}^K \subset \mathscr{P}(\mathbb{R}^d)$, with $K \geq 1$, defined such that p_0 is an easy-to sample distribution and p_k is harder and harder to sample as k increases. In this section, we assume that each distribution p_k is absolutely continuous with respect to the Lebesgue measure, with tractable *unnormalized* density and tractable score, where both can be evaluated pointwise.

Reminders on the Metropolis-Adjusted Langevin Algorithm (MALA). This sampling algorithm is the only one in this section to be designed so as to sample from a single distribution $\pi \in \mathscr{P}(\mathbb{R}^d)$ with density $\gamma : \mathbb{R}^d \to \mathbb{R}_+$ known up to a normalizing constant. It is an extension of the largely used Unadjusted Langevin Algorithm (ULA) [Roberts and Tweedie, 1996].

Given a number of steps $L \geq 1$, a step-size $\lambda > 0$ and an easy-to-sample distribution $p^{\mathrm{init}} \in \mathscr{P}(\mathbb{R}^d)$, ULA builds a Markov chain $(X^\ell)_{\ell=0}^L$ defined by $X^0 \sim p^{\mathrm{init}}$ and for any $\ell \in \{0, \dots, L-1\}$,

$$X^{\ell+1} = X^\ell + \lambda \nabla \log \gamma(X^\ell) + \sqrt{2\lambda} Z^\ell , \; Z^\ell \sim \mathrm{N}(0, \mathrm{I}_d) , \tag{D.109}$$

which reduces to a time discretization of the Langevin diffusion that admits π as invariant distribution. Building upon this recursion, MALA [Roberts and Tweedie, 1996] addresses the discretization bias by adding a Metropolis-Hastings acceptance/rejection step at each iteration, see Algorithm 8. In practice, one can easily adapt the step-size λ by targeting a specific acceptance ratio. We follow this paradigm in our experiments, see Appendix D.8 for more details.

Algorithm 8 Metropolis-Adjusted Langevin Algorithm (MALA)

Input: Target distribution π with unnormalized density γ, step size $\lambda > 0$, number of steps $L \geq 1$, initial distribution p^{init}

▷ Initialization
$X^0 \sim p^{\text{init}}$
for $\ell \leftarrow 0$ **to** $L-1$ **do**
  ▷ Propose a new sample
  $\tilde{X}^{\ell+1} \leftarrow X^\ell + \lambda \nabla \log \gamma(X^\ell) + \sqrt{2\lambda} Z^\ell, \quad Z^\ell \sim \mathrm{N}(0, \mathrm{I}_d)$
  ▷ Compute the acceptance ratio
  $\alpha = \min\left(1, \frac{\gamma(\tilde{X}^{\ell+1}) q(X^\ell \mid \tilde{X}^{\ell+1})}{\gamma(X^\ell) q(\tilde{X}^{\ell+1} \mid X^\ell)}\right)$ where $q(y \mid x) = \mathrm{N}(y; x + \lambda \nabla \log \gamma(x), 2\lambda \mathrm{I}_d)$
  ▷ Accept or reject depending on α
  $U \sim \mathrm{U}([0,1])$
  if $U \leq \alpha$ **then**
    $X^{\ell+1} \leftarrow \tilde{X}^{\ell+1}$
  else
    $X^{\ell+1} \leftarrow X^\ell$
  end
end

Output: Sequence of samples $X^{0:L}$ such that $X^{0:L} \overset{\text{iid}}{\sim} \pi$ (approximately)

Algorithm 9 Annealed Langevin MCMC (AL-MCMC) algorithm

Input: Sequence of target distributions $\{p_k\}_{k=0}^K$, step-sizes $\{\lambda_k\}_{k=1}^K \in (0,\infty)^K$, number of MALA steps per level $L \geq 1$

▷ Initialization
$X_0^{0:L} \overset{\text{iid}}{\sim} p_0$
for $k \leftarrow 0$ **to** $K-1$ **do**
  ▷ Sample from p_{k+1} by starting at samples obtained from p_k
  $X_{k+1}^{0:L} \leftarrow \mathrm{MALA}(p_{k+1}, \lambda_{k+1}, L, \delta_{X_k^L})$, see Algorithm 8)
end

Output: Sequence of samples $X_{0:K}^{0:L}$ such that $X_k^{0:L} \overset{\text{iid}}{\sim} p_k$ (approximately)

Annealed Langevin MCMC (AL-MCMC). Based on exact samples from p_0, the *Annealed Langevin MCMC* algorithm [Song and Ermon, 2019, 2020] consists in sequentially sampling from p_{k+1} with MALA (Algorithm 8) initialized in the last samples obtained at step k. The core idea behind this procedure is that the warm initialization obtained by the easy-to-sample distributions (*i.e.*, when k is low) can help to overcome sampling issues, such as high energy barriers, in the more complex distributions (*i.e.*, when k is large). We refer to Algorithm 9 for the pseudo-code of this approach.

Sequential Monte Carlo (SMC) and extension. The *Sequential Monte Carlo* [Neal, 2001, Del Moral et al., 2006] algorithm lies on the same paradigm as Algorithm 9 but proceeds in *parallel* over N chains (which are referred as *particles*) while ensuring that the warm initialization is correct. To do so, SMC relies on an additional resampling step that may occur before certain MALA run, based on accumulated importance weights that we next define.

Let $N \geq 1$, $k \in \{0, \dots, K-1\}$, $(\tilde{w}_k^n)_{n=1}^N$ be unnormalized importance weights accumulated on each chain up to step k ($\tilde{w}_k^n = 1/N$ if $k = 0$), and $(x_k^n)_{n=1}^N$ be samples from p_k obtained independently as the last samples of N parallel MALA runs (or exactly obtained if $k = 0$). Then, the SMC importance weights designed to sample from p_{k+1}, denoted by $(\tilde{w}_{k+1}^n)_{n=1}^N$, and their renormalized versions, denoted by $(w_{k+1}^n)_{n=1}^N$, are defined as

$$\tilde{w}_{k+1}^n = \frac{p_{k+1}(x_k^n)}{p_k(x_k^n)} \tilde{w}_k^n, \quad w_{k+1}^n = \frac{\tilde{w}_{k+1}^n}{\sum_{j=1}^N \tilde{w}_{k+1}^j}. \tag{D.110}$$

The SMC resampling step then consists in obtaining N new samples $(y_{k+1}^n)_{n=1}^N$ by random selection among the original samples $(x_k^n)_{n=1}^N$ via a multinomial distribution (with replacement) defined by the normalized weights $(w_{k+1}^n)_{n=1}^N$. These novel samples will then be used as starting points for running MALA on the target p_{k+1}. In practice, this resampling step only occurs adaptively based on the value of the Effective Sample Size (ESS)

Algorithm 10 Sequential Monte Carlo (SMC) algorithm with adaptive resampling

Input: Sequence of target distributions $\{p_k\}_{k=0}^K$, step-sizes $\{\lambda_k\}_{k=1}^K \in (0,\infty)^K$, number of MCMC steps per level $L \geq 1$, number of particles $N \geq 1$, ESS resampling threshold $\alpha \in (0,1]$

▷ Initialization
$X_0^{1:N} \overset{\text{iid}}{\sim} p_0$, $\tilde{W}_0^{1:N} = 1/N$
for $k \leftarrow 0$ **to** $K-1$ **do**
  ▷ 1. Compute the SMC importance weights (in parallel, for $n = 1, \dots, N$)
  $\tilde{W}_{k+1}^n \leftarrow (p_{k+1}(X_k^n)/p_k(X_k^n))\, \tilde{W}_{k+1}^n$
  ▷ 2. Compute the accumulated ESS (in parallel, for $n = 1, \dots, N$)
  $\text{ESS}_{k+1} \leftarrow \left(\sum_{n=1}^N \tilde{W}_k^n\right)^2 / \left(\sum_{n=1}^N (\tilde{W}_k^n)^2\right)$
  ▷ 3. Resample $X_k^{1:N}$ if the ESS too low
  if $\text{ESS}_{k+1} < \alpha N$ **then**
    ▷ Normalize the importance weights (in parallel, for $n = 1, \dots, N$)
    $W_{k+1}^n \leftarrow \tilde{W}_{k+1}^n / \sum_{j=1}^N \tilde{W}_{k+1}^j$
    ▷ Resample the particles
    $I^{1:N} \sim \mathcal{M}(W_{k+1}^1, \dots, W_{k+1}^N)$
    $Y_{k+1}^{1:N} \leftarrow X_k^{I^{1:N}}$, $\tilde{W}_k^{1:N} = 1/N$
  end
  else
    $Y_{k+1}^n \leftarrow X_k^n$
  end
  ▷ 4. Sample from p_{k+1} by starting from sample Y_{k+1}^n (in parallel, for $n = 1, \dots, N$)
  $\tilde{X}_{k+1}^{1:L} \leftarrow \text{MALA}(p_{k+1}, \lambda_{k+1}, L, \delta_{Y_{k+1}^n})$, see Algorithm 8
  $X_{k+1}^n \leftarrow \tilde{X}_{k+1}^L$
end

Output: Sequence of samples $X_{0:K}^{1:N}$ such that $X_k^{1:N} \overset{\text{iid}}{\sim} p_k$ (approximately)

defined at step k by

$$\text{ESS}_k = \frac{\left(\sum_{n=1}^N \tilde{w}_k^n\right)^2}{\sum_{n=1}^N (\tilde{w}_k^n)^2} \,. \tag{D.111}$$

We detail the whole SMC procedure in Algorithm 10.

Recently, Phillips et al. [2024] proposed an extension of the SMC algorithm, *Particle Denoising Diffusion Sampler* (PDDS), where they consider the specific case where the intermediate distributions $\{p_k\}_{k=0}^K$ are defined as the approximate marginals of a denoising diffusion process. Below, we explain the fundamentals of PDDS.

Let $\pi \in \mathscr{P}(\mathbb{R}^d)$ be a hard-to-sample distribution. With the same notations as in Section 5.2.1, consider the noising process $(X^\star)_{t\in[0,T]}$ induced by the following SDE

$$\mathrm{d}X_t^\star = f(t)X_t^\star \mathrm{d}t + g(t)\mathrm{d}W_t \;,\; X_0^\star \sim \pi \;. \tag{D.112}$$

Then, assuming that $X_T^\star \sim \pi^{\text{base}}$, the time-reversal of SDE (D.112) is given by

$$\mathrm{d}Y_t^\star = -f(T-t)Y_t^\star \mathrm{d}t + g^2(T-t)\nabla \log p_{T-t}^\star(Y_t^*)\mathrm{d}t + g(t)\mathrm{d}B_t, \quad Y_0^\star \sim \pi^{\text{base}} \;, \tag{D.113}$$

where $p_t^\star$ is the density of the marginal distribution associated to $X_t^\star$, which is intractable in general. Assume that we are given a sequence of intermediate unnormalized densities $(p_t)_{t\in[0,T]}$ that approximate the target densities $(p_t^\star)_{t\in[0,T]}$.

In particular, the time-reversed SDE (D.113) can be approximated by

$$\mathrm{d}Y_t = -f(T-t)Y_t\mathrm{d}t + g^2(T-t)\nabla \log p_{T-t}(Y_t)\mathrm{d}t + g(T-t)\mathrm{d}B_t \;, Y_0 \sim \pi^{\text{base}} \;. \tag{D.114}$$

Consider a time discretization $\{t_k\}_{k=0}^K$ such that $t_0 = 0$ and $t_K = T$, and the sequence of target distributions $\{p_k\}_{k=0}^K$ defined by $p_k = p_{T-t_k}$, such that p_k is expected to be harder and harder to sample from as k increases.

Let $k \in \{0, \dots, K-1\}$, define $\delta_k = t_{k+1} - t_k$. Since SDE (D.112) is linear, it can be exactly integrated on time interval $[T - t_{k+1}, T - t_k]$, see Lemma 67. We denote by $p_{k|k+1}^\star$ the corresponding transition density.

Algorithm 11 Swapping step of Replica Exchange

Input: Sequence of target distributions $\{p_k\}_{k=0}^K$ with even K, current samples $X_{1:K}$, swapping state $s \in \{0, 1\}$
if $s = 0$ **then**
  Indexes $\leftarrow \{0, 2, \dots, K-2\}$
else
  Indexes $\leftarrow \{1, 3, \dots, K-1\}$
end
▷ Compute the swapping probability between k and $k+1$ (in parallel, for $k \in$ Indexes)
$p_{k,k+1} = \min\left(1, \frac{p_k(X_{k+1})p_{k+1}(X_k)}{p_k(X_k)p_{k+1}(X_{k+1})}\right)$
▷ Swap depending on $p_{k,k+1}$ (in parallel, for $k \in$ Indexes)
$U \sim \mathrm{U}([0, 1])$
if $U < p_{k,k+1}$ **then**
  $(X_k, X_{k+1}) \leftarrow (X_{k+1}, X_k)$
end
Output: Randomly swapped sequence $X_{0:K}$

On the other hand, SDE (D.114) can only be *approximately* integrated on time interval $[t_k, t_{k+1}]$ (due to the score term in the drift). For instance, this can be done using the Euler-Maruyama transition kernel[2]

$$p_{k+1|k}(\cdot|y_k) = \mathrm{N}(y_k - f(T - t_k)\delta_k y_k + g^2(t)T - t_k)\delta_k \nabla \log p_{T-t_k}(y_k), g^2(t)T - t_k)\delta_k \,\mathrm{I}_d) \,. \tag{D.115}$$

To sample sequentially from $\{p_k\}_{k=0}^K$, Phillips et al. [2024] suggest to apply the SMC procedure detailed in Algorithm 10, where the k-th resampling step (1-3) is now defined as follows.

0. Sample $\hat{X}_{k+1}^n \sim p_{k+1|k}(\cdot \mid X_k^n)$ (in parallel, for $n = 1, \dots, N$);

1. Compute the importance weights (in parallel, for $n = 1, \dots, N$)

$$\tilde{W}_{k+1}^n = \frac{p_{k+1}(\hat{X}_{k+1}^n)p_{k|k+1}^\star(X_k^n \mid \hat{X}_{k+1}^n)}{p_k(X_k^n)p_{k+1|k}(\hat{X}_{k+1}^n \mid X_k^n)}\tilde{W}_k^n;$$

2. Apply the resampling procedure (based on the ESS value) to the particles $\hat{X}_{k+1}^{1:N}$ with importance weights given by $\{W_{k+1}^n\}_{n=1}^N$.

Hence, this novel resampling step amounts to reweight the samples obtained from the *approximate* time-reversed SDE (D.114) by using the *exact* noising scheme as proposal.

Replica Exchange (RE). The *Replica Exchange* algorithm [Geyer et al., 1991] from the annealed distributions $\{p_k\}_{k=0}^K$ with $K+1$ parallel Markov chains, *i.e.*, the k-th chain targets p_k. For the sake of pedagogy, we assume here that K is even.

At each level, RE sampling is done via a local MCMC sampler, such as MALA, which is expected to have poor performance for large k. To alleviate this issue, every L MALA steps, RE randomly performs a *swapping* between Markov chains. More specifically, each consecutive pair $(k, k+1)$ of Markov chains (*i.e.*, either $(k, k+1) \in \{(0, 1), \dots, (K-2, K-1)\}$ or $(k, k+1) \in \{(1, 2), \dots, (K-1, K)\}$), with respective current states X_k and X_{k+1}, is swapped with probability

$$p_{k,k+1} = \min\left(1, \frac{p_k(X_{k+1})p_{k+1}(X_k)}{p_k(X_k)p_{k+1}(X_{k+1})}\right) \,. \tag{D.116}$$

We provide the pseudo-code of this algorithm in Algorithm 12.

Known limitations of annealed MCMC samplers. In general, annealed MCMC samplers usually target a sequence of distributions $\{p_k\}_{k=0}^K$, which forms a geometric bridge between an easy-to-sample distribution π^{base} and a hard-to-sample distribution π, *i.e.*, $p_k \propto (\pi^{\text{base}})^{(K-k)/K}\pi^{k/K}$. However, this scheme suffer from mode-switching for multi-modal distribution π [Phillips et al., 2024], *i.e.*, the mode weights of the intermediate distributions may vary a lot as k increases. Therefore, transitions between annealing levels is hard in practice,

[2] In the original paper, the authors consider the VP noising scheme and compute $p_{k+1|k}$ as the transition kernel obtained by Exponential Integration of SDE (D.114) on $[t_k, t_{k+1}]$.

Algorithm 12 Replica Exchange (RE) algorithm

Input: Sequence of target distributions $\{p_k\}_{k=0}^K$ with even K, step-sizes $\{\lambda_k\}_{k=1}^K \in (0,\infty)^K$, total number of MCMC steps $M \geq 1$, swap frequency $S \geq 1$

▷ Initialization
$X_{0:K}^0 \overset{\text{iid}}{\sim} p_0$
$s \leftarrow 0$
for $m \leftarrow 0$ **to** $M-1$ **do**
  if $(m+1) \bmod S == 0$ **then**
    ▷ Swap Markov chains for indexes determined by s
    $X_{0:K}^{m+1} \leftarrow \text{swapRE}(X_{0:K}^m, s)$, see Algorithm 11
    $s \leftarrow (s+1) \bmod 2$
  else
    ▷ Sample locally (in parallel, for $k = 0, \ldots, K$)
    $X_k^{m+1} \leftarrow \text{MALA}(p_k, \lambda_k, 1, \delta_{X_k^m})$, see Algorithm 8
  end
end

Output: Sequence of samples $X_{0:K}^{0:M}$ such that $X_k^{0:M} \overset{\text{iid}}{\sim} p_k$ (approximately)

as samples can easily get stuck in a high-probability modes. Moreover, due to the use of reweighting schemes, SMC and RE are highly sensitive to the overlap between consecutive distributions, which requires to take K large in complex multi-modal settings, hence increasing the complexity of the methods.

D.6 Energy-based Models

This section is dedicated to describing Energy-Based Models.

D.6.1 Single-Level EBMs

Let $p \in \mathscr{P}(\mathbb{R}^d)$. We assume having access to samples from p but not to its density. Energy-based Models (EBM) perform a density estimation task by modeling the density of p with $p^\psi(x) = \exp(-E^\psi(x))/\mathcal{Z}^\psi$, where $E^\psi : \mathbb{R}^d \to \mathbb{R}$ is a neural network and $\mathcal{Z}^\psi = \int_{\mathbb{R}^d} \exp(-E^\psi(x))\mathrm{d}x$ is an unknown normalizing constant. This model is trained by maximizing the log-likelihood of the model $\mathcal{L}(\psi) = \mathbb{E}[\log p^\psi(X)]$ where $X \sim p$, whose gradients can be computed as

$$\nabla_\psi \mathcal{L}(\psi) = \mathbb{E}[\nabla_\psi E^\psi(X^-)] - \mathbb{E}[\nabla_\psi E^\psi(X^+)], \quad X^- \sim p^\psi, \quad X^+ \sim p\,. \tag{D.117}$$

Given samples from p (referred to as positive samples) and samples from p^ψ (referred to as negative samples), one can compute a Monte-Carlo estimator of $\nabla_\psi \mathcal{L}(\psi)$ using (D.117). In practice, the negative samples are usually obtained by running a MCMC sampler on p^ψ (which is possible because it is known up to a normalizing constant). However, obtaining truthful negative samples is the main challenge of the EBM training procedure as p^ψ is expected to be as multi-modal as p.

D.6.2 Multi-level EBMs

Let $p \in \mathscr{P}^{(K+1)}$, with $K \geq 1$. For $k = 1, \dots, K$, we assume having access to samples from p_k but not to its density. Regarding p_0, we assume it to be known entirely. Our goal is to approximate the density of p_k at every level $k \in \{1, \dots, K\}$. A naive approach would consist in defining single-level EBMs for each k. Assuming that the distributions p_k are not completely unrelated, doing this would be however very inefficient. This section presents alternatives to this approach, which are all implemented in our code.

Prior works

Diffusion Recovery Likelihood (DRL). Recently, Gao et al. [2021], Zhu et al. [2024] proposed the DRL framework, based on the extra assumption that for any $k \in \{0, \dots, K-1\}$, the conditional distribution $p_{k|k+1}$ is known and given by $p_{k|k+1}(\cdot \mid x_{k+1}) = \mathrm{N}(\alpha_{k+1}x_{k+1}, \sigma_{k+1}^2 \mathrm{I}_d)$ for some $\alpha_{k+1} \in \mathbb{R}$ and $\sigma_{k+1} > 0$. This is typically the case when the joint distribution p is defined via a discrete time approximation of a linear SDE, see Appendix D.3. In the following, we denote $Y_k = \alpha_k X_k$ where $X_k \sim p_k$. In DRL, the authors suggest to use a multi-level EBM $p_k^\psi(\cdot) = \exp(-E^\psi(k, \cdot))/\mathcal{Z}_k^\psi$, where $E^\psi : \{1, \dots, K\} \times \mathbb{R}^d \to \mathbb{R}$ is a neural network and the normalizing constant $\mathcal{Z}_k^\psi = \int_{\mathbb{R}^d} \exp(-E^\psi(k, x))\mathrm{d}x$ is intractable. In this case, the density of p_k is approximated, up to a multiplicative constant, by $\exp(-E^\psi(k, \cdot))$. Using Bayes rule, they define for any $k \in \{0, \dots, K-1\}$ the following conditional EBM

$$p_{k+1|k}^\psi(y_{k+1} \mid x_k) \propto p_{k|k+1}(x_k \mid y_{k+1}/\alpha_{k+1}) p_{k+1}^\psi(y_{k+1}) \tag{D.118}$$

$$= \left(\mathcal{Z}^\psi(k+1, x_k)\right)^{-1} \exp\left(-\frac{1}{2\sigma_{k+1}^2}\|x_k - y_{k+1}\|^2 - E^\psi(k+1, y_{k+1})\right)\,, \tag{D.119}$$

where $\mathcal{Z}^\psi(k+1, x_k) = \int_{\mathbb{R}^d} \exp\left(-\left(2\sigma_{k+1}^2\right)^{-1}\|x_k - y\|^2 - E^\psi(k+1, y)\right)\mathrm{d}y$. Note that if σ_{k+1}^2 is small enough, for instance when K is large, then the quadratic term dominates in the expression of $\log p_{k+1|k}^\psi(\cdot \mid x_k)$, which means that this conditional distribution localizes on x_k. Therefore, by applying a first order Taylor expansion of the energy term at x_k, the following Gaussian approximation holds

$$p_{k+1|k}^\psi(y_{k+1} \mid x_k) \approx \mathrm{N}\left(x_k - \sigma_{k+1}^2 \nabla E^\psi(k+1, x_k), \sigma_{k+1}^2 \mathrm{I}_d\right)\,. \tag{D.120}$$

Hence, if σ_{k+1}^2 is chosen small enough, this conditional distribution will be very easy to sample from. Based on this observation, the authors suggest to use the following maximum likelihood objective

$$\mathcal{L}^{\mathrm{DRL}}(\psi) = \sum_{k=0}^{K-1} \mathcal{L}_k^{\mathrm{DRL}}(\psi)\,, \text{ with } \mathcal{L}_k^{\mathrm{DRL}}(\psi) = \mathbb{E}\left[\log p_{k+1|k}^\psi(\alpha_{k+1}X_{k+1} \mid X_k)\right], \tag{D.121}$$

Algorithm 13 Gibbs-within-Langevin MCMC transition kernel

Input: Previous state $(k, x) \in \{0, \dots, K\} \times \mathbb{R}^d$, step-size $\lambda > 0$
▷ Sample $\tilde{x}$ given k with standard MCMC
$\tilde{x} \leftarrow \mathrm{MALA}(p^\psi(\cdot \mid k), \lambda)$ where $p^\psi(\cdot \mid k) \propto p_k^\psi(\cdot)$
▷ Sample $\tilde{k}$ given $\tilde{x}$ with a multinomial distribution
$\tilde{k} \sim \mathcal{M}\left(\exp(-E^\psi(0, \tilde{x})), \dots, \exp(-E^\psi(K, \tilde{x}))\right)$
Output: Next state $(\tilde{k}, \tilde{x})$

where $X_{k+1} \sim p_{k+1}$ and $X_k \sim p_{k|k+1}(\cdot|X_{k+1})$. In particular, the gradient of the single-level DRL loss defined in (D.121) can be expressed as

$$\nabla_\psi \mathcal{L}_k^{\mathrm{DRL}}(\psi) = \mathbb{E}[\nabla_\psi E^\psi(k, Y_k^-)] - \mathbb{E}[\nabla_\psi E^\psi(k, \alpha_k X_k^+)] \,, \tag{D.122}$$

where $X_k \sim p_k$, $Y_{k+1}^- \sim p_{k+1|k}^\psi(\cdot \mid X_k)$ and $X_{k+1}^+ \sim p_{k+1}$. Up to a multiplicative constant, the density of p_k can then be approximated by $x_k \mapsto p_k^\psi(\alpha_k x_k)$.

Diffusion Assisted EBM (DA-EBM). On the other hand, [Zhang et al., 2023] suggest to solve the problem by jointly modeling indexes and states. Here, the multi-level EBM is expressed as $p_k^\psi(x) = \exp(-E^\psi(k, x))/\mathcal{Z}^\psi$, where $E^\psi : \{0, \dots, K\} \times \mathbb{R}^d \to \mathbb{R}$ is a neural network and the normalizing constant $\mathcal{Z}^\psi = \sum_{k=0}^K \int_{\mathbb{R}^d} \exp(-E^\psi(k, x)) \mathrm{d}x$, which can be learn by maximizing the following maximum likelihood objective

$$\mathcal{L}^{\mathrm{DAEBM}}(\psi) = \mathbb{E}\left[\log p_k^\psi(X_k)\right], \quad k \sim \mathrm{U}(\{0, \dots, K\}), \quad X_k \sim p_k \,. \tag{D.123}$$

The gradient of objective (D.123) can be written as

$$\nabla_\psi \mathcal{L}^{\mathrm{DAEBM}}(\psi) = \mathbb{E}[\nabla_\psi E^\psi(k^-, X^-)] - \mathbb{E}[\nabla_\psi E^\psi(k^+, X_k^+)] \,, \tag{D.124}$$

where $(k^-, X^-) \sim p^\psi$, $k^+ \sim \mathrm{U}(\{0, \dots, K\})$ and $X_k^+ \sim p_{k^+}$. In the same fashion as Kim and Ye [2023], the authors suggest to perform the negative sampling from p^ψ by doing a Gibbs-within-Langevin procedure, whose transition kernel is summarized in Algorithm 13.

Our approach: using an annealed MCMC sampler as backbone

In this paper, we also suggest to learn a multi-level EBM approximating p, defined by $p_k^\psi(x) = \exp\left(-E^\psi(k, x)\right)/\mathcal{Z}_k^\psi$, where $E^\psi : \{0, \dots, K\} \times \mathbb{R}^d \to \mathbb{R}$ is a neural network and the normalizing constant $\mathcal{Z}_k^\psi = \int_{\mathbb{R}^d} \exp\left(-E^\psi(k, x)\right) \mathrm{d}x$. We propose to learn ψ by maximizing a simple yet novel maximum likelihood joint objective

$$\mathcal{L}^{\mathrm{ours}}(\psi) = \sum_{k=0}^K \mathcal{L}_k^{\mathrm{ours}}(\psi) \,, \; \mathcal{L}_k^{\mathrm{ours}}(\psi) = \mathbb{E}\left[\log p_k^\psi(X_k)\right], \quad X_k \sim p_k \,, \tag{D.125}$$

with single-level gradient given by

$$\nabla_\psi \mathcal{L}_k^{\mathrm{ours}}(\psi) = \mathbb{E}[\nabla_\psi E^\psi(k, X_k^-)] - \mathbb{E}[\nabla_\psi E^\psi(k, X_k^+)] \,, \; X_k^- \sim p_k^\psi \,, \; X_k^+ \sim p_k \,. \tag{D.126}$$

Unlike previous multi-level EBM algorithms, the negative sampling phase can be simply done in our framework by leveraging annealed MCMC algorithms presented in Appendix D.5 on the sequence of EBM densities $\{p^\psi(k, \cdot)\}_{k=0}^K$. This allows us to kill two birds in one stone as (i) we get negative samples for each single-level EBM and (ii) we overcome the negative sampling limitations of each EBM by leveraging the correlation between the consecutive levels. In practice, we use Replica Exchange as default annealed MCMC sampler because of its massively parallel capabilities. We summarize our training procedure in Algorithm 14. For ease of reading, we presented the continuous time analog of this approach in the main part of the paper, see Section 5.3.3.

Multi-level EBM parameterization as GMM tilting. To reduce the computational footprint of multi-level EBMs in the context of LRDS, we suggest to define our EBM model as a tilting of a learned GMM. Let $\bar{\psi} = \{w_j, \mathbf{m}_j, \Sigma_j\}_{j=1}^J$ be the parameters learned by GMM-LRDS (Algorithm 6) to approximate $\hat{\pi}^{\mathrm{ref}}$ and denote by $p_t^{\bar{\psi}}$ the t-th marginal density of the resulting reference process. Building on this, we define p_t^ψ as a tilting of $p_t^{\bar{\psi}}$ by

$$\log p_t^\psi(x) = \log p_t^{\bar{\psi}}(x) - \mathrm{NN}^\psi(t, x) \,, \tag{D.127}$$

Algorithm 14 Multi-level EBM training using annealed MCMC samplers

Input: Number of traing iterations $N \geq 1$, initial parameter ψ_0, batch size $B \geq 1$, number of MCMC steps $L \geq 1$, predefined annealed MCMC sampler

for $n \leftarrow 0$ **to** $N-1$ **do**

▷ Get B samples from p_k (in parallel, for $k = 0, \dots, K$)

$\{X_{k,b}^+\}_{b=1}^B \overset{\text{iid}}{\sim} p_k$

▷ Get B samples from each $p_k^{\psi_n}$ using the annealed MCMC sampler

$\{X_{k,b}^-\}_{b=1,k=0}^{B,K} \leftarrow \text{annealedMCMC}\left(\{p_k^{\psi_n}\}_{k=0}^K, B\right)$

▷ Apply a stochastic gradient descent on ψ_n

Compute the MC estimator $\hat{g}_n$ of the gradient of the loss $\mathcal{L}^{\text{ours}}(\psi_n)$ defined in (D.125)

$\hat{g}_n \leftarrow B^{-1} \sum_{k=0}^K \left(\sum_{b=1}^B \nabla_\psi E^{\psi_n}\left(k, X_{k,b}^-\right) - \sum_{b=1}^B \nabla_\psi E^{\psi_n}\left(k, X_{k,b}^+\right)\right)$

Update ψ_n to ψ_{n+1} with Adam optimizer based on $\hat{g}_n$

end

Output: Optimized parameter ψ_N

where $\text{NN}^\psi : [0,T] \times \mathbb{R}^d \to \mathbb{R}$ is a neural network with parameter ψ. In this scenario, we design the initial ψ_0 to ensure that $\text{NN}^{\psi_0} = 0$. In practice, in the case where the GMM shares very few similarities with $\hat{\pi}^{\text{ref}}$ close to $t = 0$, we rather consider

$$\log p_t^\psi(x) = \mathbf{1}_{t > t_{\text{lim}}}(t) \times \log p_t^{\bar{\psi}}(x) - \text{NN}^\psi(t,x) \,, \tag{D.128}$$

where $t_{\text{lim}} \in [0,T]$. For most of our numerical experiments, we found that $t^{\text{lim}} = 0.2$ brought satisfying results.

D.7 Additional metrics for variational diffusion-based methods

When specifically using annealed VI methods, one may consider additional metrics to evaluate the performance of the corresponding samplers. For completeness purpose, we provide in this section the expressions of the sampling metrics presented by Blessing et al. [2024], for all discrete time variational diffusion-based methods presented in this paper: RDS (including PIS and DDS), DIS and CMCD. All of these metrics are implemented in our code.

Given a fixed level of time discretization $K \geq 1$, these performance criteria require the evaluation of a variational *importance weight function* $w : (\mathbb{R}^d)^{K+1} \to \mathbb{R}_+$, assessing the quality of the variational approximation, that is specific to each variational setting.

- For RDS (including PIS and DDS), we define

$$w(y_{0:K}) = \frac{\mathcal{Z}^{\text{ref}}}{\mathcal{Z}} \frac{\mathrm{d}p_{0:K}^\theta(y_{0:K})}{\mathrm{d}p_{0:K}^\star(y_{0:K})} = \frac{\mathrm{d}p_{0:K}^\theta(y_{0:K})\gamma^{\text{ref}}(y_K)}{\mathrm{d}p_{0:K}^{\text{ref}}(y_{0:K})\gamma(y_K)} \,, \tag{D.129}$$

where $p^\theta \in \mathscr{P}^{(K+1)}$ and $p^{\text{ref}} \in \mathscr{P}^{(K+1)}$ are respectively defined as discrete time approximations of the path measures $\mathbb{P}^\theta$ and $(\mathbb{P}^{\text{ref}})^R$; see Appendix D.4.3 for more details.

- For DIS, we define

$$w(y_{0:K}) = \frac{1}{\mathcal{Z}} \frac{\mathrm{d}p_{0:K}^\theta(y_{0:K})}{\mathrm{d}p_{0:K}^\star(y_{0:K})} \,, \tag{D.130}$$

where $p^\theta \in \mathscr{P}^{(K+1)}$ and $p^\star \in \mathscr{P}^{(K+1)}$ are respectively defined as discrete time approximations of the path measures $\mathbb{P}^\theta$, induced by SDE (D.27), and $(\mathbb{P}^\star)^R$; see Appendix D.4.4 for more details.

- For CMCD, we define

$$w(y_{0:K}) = \frac{1}{\mathcal{Z}} \frac{\mathrm{d}p_{0:K}^\theta(y_{0:K})}{\mathrm{d}q_{0:K}^\theta(y_{0:K})} \tag{D.131}$$

where $p^\theta \in \mathscr{P}^{(K+1)}$ and $q^\theta \in \mathscr{P}^{(K+1)}$ are respectively defined as discrete time approximations of the path measures $\mathbb{P}^\theta$, induced by SDE (D.29), and $(\mathbb{Q}^\theta)^R$, where $\mathbb{Q}^\theta$ is induced by SDE (D.28); see Appendix D.4.5 for more details.

Note that, for each variational setting, w does not depend on the normalization constant of the target distribution : in particular, $\mathcal{Z}$ simplifies in the DIS and CMCD expressions. This also stands for $\mathcal{Z}^{\text{ref}}$ in the RDS setting.

D.7.1 'Reverse' performance criteria

Definition. Standard variational metrics, referred to as 'reverse' performance criteria by Blessing et al. [2024], are defined as expectations *with respect to the variational distribution.*

(a) The **Evidence Lower Bound** (ELBO), expected to be maximized, defined by

$$\mathrm{ELBO} = -\mathbb{E}[\log w(Y^\theta_{0:K})] \ , \ Y^\theta_{0:K} \sim p^\theta_{0:K} \ . \tag{D.132}$$

For any RDS setting with intractable $\mathcal{Z}^{\mathrm{ref}}$, this criterion may assess the performance of the RDS sampler itself, but cannot be used for numerical comparison with different variational settings.

(b) The **MC estimation of the normalizing constant** $\hat{\mathcal{Z}}_r$, expected to be close to $\mathcal{Z}$, defined by

$$\hat{\mathcal{Z}}_r = \mathbb{E}[w(Y^\theta_{0:K})] \ , \ Y^\theta_{0:K} \sim p^\theta_{0:K} \ . \tag{D.133}$$

For any RDS setting with intractable $\mathcal{Z}^{\mathrm{ref}}$, this criterion cannot be used.

(c) The **normalized Explained Sum of Squares** (nESS_r), expected to be close to 1, defined by

$$\mathrm{nESS}_r = \frac{\mathbb{E}[w(Y^\theta_{0:K})]^2}{\mathbb{E}[w(Y^\theta_{0:K})^2]} \ , \ Y^\theta_{0:K} \sim p^\theta_{0:K} \ . \tag{D.134}$$

Computation. Based on the results from Appendix D.4.2, we detail the simulation of $Y^\theta_{0:K}$ and the computation of $\log w(Y^\theta_{0:K})$ for each setting.

- **RDS setting**: $Y^\theta_0 \sim \pi^{\mathrm{base}}$, and for any $k \in \{0, \dots, K-1\}$,

$$Y^\theta_{k+1} = a_k Y^\theta_k + b_k\{s^{\mathrm{ref}}_{T-t_k}(Y^\theta_k) + \phi^\theta_{T-t_k}(Y^\theta_k)\} + \sqrt{c_k} Z_k \ , \ Z_k \sim \mathrm{N}(0, \mathrm{I}_d) \ , \tag{D.135}$$

and the importance weight function is given by

$$\log w(Y^\theta_{0:K}) = \sum_{k=0}^{K-1} \frac{w_k}{2} \left\|\phi^\theta_{T-t_k}(Y^\theta_k)\right\|^2 + \sum_{k=0}^{K-1} \sqrt{w_k} \phi^\theta_{T-t_k}(Y^\theta_k)^\top Z_k + \log \frac{\gamma^{\mathrm{ref}}}{\gamma}(Y^\theta_K) \ , \tag{D.136}$$

where $\{w_k, a_k, b_k, c_k\}_{k=0}^{K-1}$ depend on the noising scheme and the discretization setting.

- **DIS setting**: $Y^\theta_0 \sim \pi^{\mathrm{base}}$, and for any $k \in \{0, \dots, K-1\}$,

$$Y^\theta_{k+1} = a_k Y^\theta_k + b_k s^\theta_{T-t_k}(Y^\theta_k) + \sqrt{c_k} Z_k \ , \ Z_k \sim \mathrm{N}(0, \mathrm{I}_d) \tag{D.137}$$

and the importance weight function is given by

$$\log w(Y^\theta_{0:K}) = \sum_{k=0}^{K-1} \frac{w_k}{2} \|s^\theta_{T-t_k}(Y^\theta_k)\|^2 + \sum_{k=0}^{K-1} \sqrt{w_k} s^\theta_{T-t_k}(Y^\theta_k)^\top Z_k + \log \frac{\pi^{\mathrm{base}}(Y^\theta_0)}{\gamma(Y^\theta_K)} \ , \tag{D.138}$$

where $\{w_k, a_k, b_k, c_k\}_{k=0}^{K-1}$ come from the EI discretization of the VP noising scheme.

- **CMCD setting**: $Y^\theta_0 \sim \pi^{\mathrm{base}}$, and for any $k \in \{0, \dots, K-1\}$,

$$Y^\theta_{k+1} = Y^\theta_k + \frac{\sigma^2 \delta_k}{2}\{\nabla \log \pi_{t_k}(Y^\theta_k) + h^\theta_{t_k}(Y^\theta_k)\} + \sigma\sqrt{\delta_k} Z_k \ , \ Z_k \sim \mathrm{N}(0, \mathrm{I}_d) \ , \tag{D.139}$$

and the importance weight function is given by

$$\log w(Y^\theta_{0:K}) = \frac{1}{2} \sum_{k=0}^{K-1} w_k^{\mathrm{CMCD}} \left\|u^\theta_k\right\|^2 + \sum_{k=0}^{K-1} \sqrt{w_k^{\mathrm{CMCD}}} (u^\theta_k)^\top Z_k + \log \frac{\pi^{\mathrm{base}}(Y^\theta_0)}{\gamma(Y^\theta_K)} \ , \tag{D.140}$$
$$\text{with } u^\theta_k = \nabla \log \pi_{t_k}(Y^\theta_k) + \nabla \log \pi_{t_{k+1}}(Y^\theta_{k+1}) + h^\theta_{t_k}(Y^\theta_k) - h^\theta_{t_{k+1}}(Y^\theta_{k+1}) \ .$$

D.7.2 'Forward' performance criteria

Definition. As observed by Blessing et al. [2024], the 'reverse' variational metrics presented above may not be able to quantify mode collapse. Hence, the authors propose novel variational metrics, referred to as 'forward' performance criteria, that are defined as expectations *with respect to the ground truth distribution*.

(a) The **Evidence Upper Bound** (EUBO), expected to be minimized, defined by

$$\mathrm{EUBO} = \mathbb{E}[\log w(Y_{0:K})] \ , \ Y_{0:K} \sim p^\star_{0:K} \ . \tag{D.141}$$

For any RDS setting with intractable $\mathcal{Z}^{\mathrm{ref}}$, this criterion may assess the performance of the RDS sampler itself, but cannot be used for numerical comparison with different variational settings.

(b) The **MC estimation of the normalizing constant** $\hat{\mathcal{Z}}_f$, expected to be close to $\mathcal{Z}$, defined by

$$\hat{\mathcal{Z}}_f = \left(\mathbb{E}[w(Y_{0:K})^{-1}]\right)^{-1} \ , \ Y_{0:K} \sim p^\star_{0:K} \ . \tag{D.142}$$

For any RDS setting with intractable $\mathcal{Z}^{\mathrm{ref}}$, this criterion cannot be used.

(c) The **normalized Explained Sum of Squares** (nESS_f), expected to be close to 1, defined by

$$\mathrm{nESS}_f = \left(\mathbb{E}[w(Y_{0:K})^{-1}]\mathbb{E}[w(Y_{0:K})]\right)^{-1} \ , \ Y_{0:K} \sim p^\star_{0:K} \ . \tag{D.143}$$

Computation. Based on the results from Appendix D.4.2, we detail the simulation of $Y_{0:K}$ and the computation of $\log w(Y_{0:K})$ for each setting.

- **RDS setting with EM scheme**: $Y_K \sim \pi$, and for any $k \in \{0, \dots, K-1\}$,

$$Y_k = S_{k+1}(t_k)Y_{k+1} + S_{k+1}(t_k)\sigma_{k+1}(t_k)Z_k \ , \ Z_k \sim \mathrm{N}(0, \mathrm{I}_d) \ , \tag{D.144}$$

where $S_k(t) = \exp\left(\int_{T-t_k}^{T-t} f(u)\mathrm{d}u\right)$, $\sigma_k^2(t) = \int_{T-t_k}^{T-t} \frac{g^2(u)}{S_k(u)^2}\mathrm{d}u$, and the importance weight function is given by

$$\begin{aligned}\log w(Y_{0:K}) = &- \sum_{k=0}^{K-1} w_k^{\mathrm{EM}} \phi^\theta_{T-t_k}(Y_k)^\top \left\{ s^{\mathrm{ref}}_{T-t_k}(Y_k) + \frac{1}{2}\phi^\theta_{T-t_k}(Y_k) \right\} \\ &- \sum_{k=0}^{K-1} \sigma_{k+1}(t_k)\phi^\theta_{T-t_k}(Y_k)^\top Z_k + \sum_{k=0}^{K-1} \left(\frac{1}{S_{k+1}(t_k)} - 1 + f(T-t_k)\delta_k \right) \phi^\theta_{T-t_k}(Y_k)^\top Y_k \\ &+ \log \frac{\gamma^{\mathrm{ref}}}{\gamma}(Y_K) \ ,\end{aligned} \tag{D.145}$$

as a consequence of Lemma 75.

- **RDS setting with EI scheme (PBM)**: $Y_K \sim \pi$, and for any $k \in \{0, \dots, K-1\}$,

$$Y_k = \frac{Y_{k+1}}{a_k^{\mathrm{PBM}}} + \frac{\sqrt{c_k^{\mathrm{PBM}}}}{a_k^{\mathrm{PBM}}} Z_k \ , \ Z_k \sim \mathrm{N}(0, \mathrm{I}_d) \ , \tag{D.146}$$

and the importance weight function is given by

$$\begin{aligned}\log w(Y_{0:K}) = &- \sum_{k=0}^{K-1} w_k^{\mathrm{PBM}} \phi^\theta_{T-t_k}(Y_k)^\top \{ s^{\mathrm{ref}}_{T-t_k}(Y_k) + \frac{1}{2}\phi^\theta_{T-t_k}(Y_k) \} \\ &- \sum_{k=0}^{K-1} \sqrt{w_k^{\mathrm{PBM}}} \phi^\theta_{T-t_k}(Y_k)^\top Z_k + \log \frac{\gamma^{\mathrm{ref}}}{\gamma}(Y_K) \ ,\end{aligned} \tag{D.147}$$

as a consequence of Lemma 78.

- **RDS setting with EI scheme (VP)**: $Y_K \sim \pi$, and for any $k \in \{0, \dots, K-1\}$,

$$Y_k = \frac{Y_{k+1}}{a_k^{\mathrm{VP}}} + \frac{\sqrt{c_k^{\mathrm{VP}}}}{a_k^{\mathrm{VP}}} Z_k \ , \ Z_k \sim \mathrm{N}(0, \mathrm{I}_d) \ , \tag{D.148}$$

and the importance weight function is given by

$$\log w(Y_{0:K}) = -\sum_{k=0}^{K-1} w_k^{\mathrm{VP}} \phi^\theta_{T-t_k}(Y_k)^\top \{s^{\mathrm{ref}}_{T-t_k}(Y_k) + \frac{1}{2}\phi^\theta_{T-t_k}(Y_k)\} \\ - \sum_{k=0}^{K-1} \sqrt{w_k^{\mathrm{VP}}}\phi^\theta_{T-t_k}(Y_k)^\top Z_k + \log \frac{\gamma^{\mathrm{ref}}}{\gamma}(Y_K) \,, \tag{D.149}$$

as a consequence of Lemma 81.

- **DIS setting**: $Y_K \sim \pi$, and for any $k \in \{0, \dots, K-1\}$,

$$Y_k = \frac{Y_{k+1}}{a_k^{\mathrm{VP}}} + \frac{\sqrt{c_k^{\mathrm{VP}}}}{a_k^{\mathrm{VP}}} Z_k \,, \; Z_k \sim \mathrm{N}(0, \mathrm{I}_d) \,, \tag{D.150}$$

and the importance weight function is given by

$$\log w(Y_{0:K}) = -\frac{1}{2}\sum_{k=0}^{K-1} w_k^{\mathrm{VP}} \left\| s^\theta_{T-t_k}(Y_k) \right\|^2 - \sum_{k=0}^{K-1} \sqrt{w_k^{\mathrm{VP}}} s^\theta_{T-t_k}(Y_k)^\top Z_k + \log \frac{\pi^{\mathrm{base}}(Y_0)}{\gamma(Y_K)} \,, \tag{D.151}$$

as a consequence of Lemma 84.

- **CMCD setting**: $Y_K \sim \pi$, and for any $k \in \{0, \dots, K-1\}$,

$$Y_k = Y_{k+1} + \frac{\sigma^2 \delta_k}{2}\{\nabla \log \pi_{t_{k+1}}(Y_{k+1}) - h^\theta_{t_{k+1}}(Y_{k+1})\} + \sigma\sqrt{\delta_k} Z_k \,, \; Z_k \sim \mathrm{N}(0, \mathrm{I}_d) \,, \tag{D.152}$$

and the importance weight function is given by

$$\log w(Y_{0:K}) = -\frac{1}{2}\sum_{k=0}^{K-1} w_k^{\mathrm{CMCD}} \left\| u_k^\theta \right\|^2 - \sum_{k=0}^{K-1} \sqrt{w_k^{\mathrm{CMCD}}} (u_k^\theta)^\top Z_k + \log \frac{\pi^{\mathrm{base}}(Y_0)}{\gamma(Y_K)} \\ \text{with } u_k^\theta = \nabla \log \pi_{t_k}(Y_k) + \nabla \log \pi_{t_{k+1}}(Y_{k+1}) + h^\theta_{t_k}(Y_k) - h^\theta_{t_{k+1}}(Y_{k+1}) \,, \tag{D.153}$$

as a consequence of Lemma 87.

D.8 Implementation details

D.8.1 Details on target distributions

Gaussian mixture from Figure 5.2. The distribution from Figure 5.2 is a 16-dimensional mixture between $\mathrm{N}(\mathbf{m}_1, \Sigma_1)$ and $\mathrm{N}(\mathbf{m}_2, \Sigma_2)$ with weights 0.7 and 0.3 respectively. Let $\mathbf{1}_d$ be the d-dimensional vector with all components equal to 1, then we have

$$\mathbf{m}_1 = -0.6 \times \mathbf{1}_{16}, \quad \mathbf{m}_2 = +0.6 \times \mathbf{1}_{16} \,. \tag{D.154}$$

Moreover, denote R_θ a rotation matrix with angle θ between the first and last axes, then

$$\Sigma_1 = R_{\pi/4}\,\mathrm{diag}(10^{-2}, \dots, 10^{-2}, 10^{-1}) R^T_{\pi/4}, \quad \Sigma_2 = R_{\pi/6}\,\mathrm{diag}(10^{-1}, 10^{-2}, \dots, 10^{-2}) R^T_{\pi/6} \,. \tag{D.155}$$

In practice, we estimate the weight of the first mode w_1 by computing a Monte Carlo estimator of

$$\int \mathbf{1}_{\mathrm{N}(x;\mathbf{m}_1,\Sigma_1) > \mathrm{N}(x;\mathbf{m}_2,\Sigma_2)}(x)\pi(x)\mathrm{d}x \,. \tag{D.156}$$

Bi-modal Gaussian mixture. This Gaussian mixture is used in Figure 5.1 and also in Section 5.5. It is a d-dimensional mixture between $\mathrm{N}(-\mathbf{1}_d, \Sigma)$ and $\mathrm{N}(\mathbf{1}_d, \Sigma)$ with weights 2/3 and 1/3 respectively. The different values of Σ are recapped in Table D.1. Just like the previous experiment, we also use the formula (D.156) to estimate the strongest mode weight with Monte Carlo approach.

Table D.1: **Covariance settings for the bi-modal Gaussian mixture**. We denote by $\text{logdiag}(a, b)$ with $0 < a < b$ the diagonal matrix in $\mathbb{R}^{d\times d}$ whose diagonal is a log-linear interpolation between a and b. Moreover, $Q_d \in \mathbb{R}^{d\times d}$ is a random orthogonal matrix built by doing the QR decomposition of a random matrix distributed according to $\text{U}([0, 5]^{d\times d})$. The seed used to generate this last matrix is fixed at 42.

Type	Difficulty	Cond. number	Covariance
Diagonal	Isotropic	1	$(5 \times 10^{-2})^2 \times \text{I}_d$
Diagonal	Medium	10^2	$(5 \times 10^{-2})^2 \times \text{logdiag}(10^{-2}, 1)$
Diagonal	Hard	10^4	$(5 \times 10^{-2})^2 \times \text{logdiag}(10^{-4}, 1)$
Full	Medium	10^2	$(5 \times 10^{-2})^2 \times Q_d \times \text{logdiag}(10^{-2}, 1) \times Q_d^T$
Full	Hard	10^4	$(5 \times 10^{-2})^2 \times Q_d \times \text{logdiag}(10^{-4}, 1) \times Q_d^T$

Multi-modal Gaussian mixtures. Given a number of mixture components $L > 2$, we define a d-dimensional Gaussian mixture $x \in \mathbb{R}^d \mapsto \sum_{\ell=1}^{L} w_\ell \text{N}(x; \mathbf{m}_\ell, 0.5\,\text{I}_d)$, where the mean locations $\{\mathbf{m}_\ell\}_{\ell=1}^{L}$ are independently distributed according to $\text{U}\left([-L, L]^d\right)$ and the mixture weights $\{w_\ell\}_{\ell=1}^{L}$ are strictly increasing, defined with a constant geometrical increment such that $w_L/w_1 = 3$. To assess the mode weight recovery in this multi-modal setting, we compute the total variation distance between the exact mode weight histogram and the mode weight histogram computed from Monte Carlo approximation.

Rings distribution. Let $T : \mathbb{R}^+ \times [0, 2\pi] \to \mathbb{R}^2$ be the transformation from polar to cartesian coordinates. Let $\theta \sim \text{U}([0, 2\pi])$ and $R \sim \text{N}(r, \sigma^2)$ with $r, \sigma > 0$, we say that $T(R, \theta)$ is distributed as a ring of radius r and width σ. The Rings distribution is a mixture of rings with radiuses $r = 1, 3, 5$ and width $\sigma = 0.1$. The weights of each ring within the mixture are 1/9, 3/9 and 5/9. In this specific case, we use more MCMC chains and GMM components than the number of modes. The initial points of the MCMC samplers are fetched by drawing the same amount of uniformly distributed points on each ring.

Checkerboard distribution. We divide the square $[-4, 4]^2$ into 16 squares of size 2×2. This distribution is defined as a mixture of uniform distributions on those squares in an interleaving fashion. We assign a weight of 18.75% to each 4 squares on the left and 6.25% to each 4 squares on the right. The initial points of the MCMC samplers are taken as the middle of each square.

The ϕ^4 field system. The ϕ^4 model is a simplified framework often used as a continuous version of the Ising model, aiding in the exploration of phase transitions within statistical mechanics. As per Gabrié et al. [2022], we focus on a discretized version of this model, set on a one-dimensional grid with size $d = 32$. Each configuration is represented as a d-dimensional vector $(\phi_i)_{i=1}^d$. To further constrain the system, we fix the field to zero at both ends by setting $\phi_0 = \phi_{d+1} = 0$.

The negative log-density of the distribution is then expressed as follows

$$\ln \pi_h(\phi) = -\beta \left(\frac{ad}{2} \sum_{i=1}^{d+1} (\phi_i - \phi_{i-1})^2 + \frac{1}{4ad} \sum_{i=1}^{d} (1 - \phi_i^2)^2 + h\phi_i \right) . \tag{D.157}$$

We selected parameter values that make the system bimodal, setting $a = 0.1$ and the inverse temperature $\beta = 20$, while adjusting the value of h. We define w_+ as the statistical frequency of configurations where $\phi_{d/2} > 0$, and w_- as the frequency where $\phi_{d/2} < 0$. When $h = 0$, the system is symmetric under the transformation $\phi \to -\phi$, so we anticipate $w_+ = w_-$. However, for $h > 0$, the negative mode becomes more prevalent.

For large values of d, the relative likelihood of the modes can be approximated using Laplace expansions at the 0th and 2nd orders. Letting ϕ_+^h and ϕ_-^h represent the local maxima of (D.157), these approximations provide the following results respectively

$$\frac{w_-}{w_+} \approx \frac{\pi_h(\phi_-^h)}{\pi_h(\phi_+^h)} , \quad \frac{w_-}{w_+} \approx \frac{\pi_h(\phi_-^h) \times |\det H_h(\phi_-^h)|^{-1/2}}{\pi_h(\phi_+^h) \times |\det H_h(\phi_+^h)|^{-1/2}} , \tag{D.158}$$

where H_h is the Hessian of the function $\phi \to \ln \pi_h(\phi)$. We compute the modes ϕ_+^h and ϕ_-^h using a gradient descent on the distribution's potential. Since we do not have access to ground truth samples in practice, we compare the Laplace approximations defined in (D.158), considered as our ground truth, to a Monte Carlo estimation of w_-/w_+.

Table D.2: **Complexity of each algorithm during the overall sampling procedure.** We track the number of target evaluations and neural network evaluations. We consider that computing π is as expensive as computing $\nabla \log \pi$. We assume an infinite parallel computational capabilities. Note that the cost of the training procedure can be amortized to sample multiple times.

Method	Number of target evaluations	Number of neural network evaluations
SMC	$\mathcal{O}(KM)$	0
RE	$\mathcal{O}(M)$	0
PIS/DDS/DIS/CMCD	$\mathcal{O}(NK)$	$\mathcal{O}(NK)$
PDDS	$\mathcal{O}(A(N+KM))$	$\mathcal{O}(A(N+KM))$
iDEM	$\mathcal{O}(AN)$	$\mathcal{O}(ANK)$
LRDS	$\mathcal{O}(N)$	$\mathcal{O}(NK)$

Bayesian logistic regression models. Finally, we evaluate the performance of a Bayesian logistic model, defined for any pair $(x, y) \in \mathbb{R}^{\dim} \times \{0, 1\}$ by the likelihood $p(y|x; w, b) = \text{Bernoulli}(y; \sigma(w^T x + b))$, where $w \in \mathbb{R}^{\dim}$ is a weight vector, $b \in \mathbb{R}$ is an intercept and σ is the sigmoid function. Given a dataset $\mathcal{D} = \{(x_j, y_j)\}_{j=1}^M \subset \mathbb{R}^{\dim} \times \{0, 1\}$ of size M, we aim to sample from the posterior distribution $p(w, b|\mathcal{D}) \propto p(w, b) \prod_{j=1}^M p(y_j|x_j; w, b)$ where $p(w, b) = p(w)p(b)$ is a fixed prior distribution. Following Blessing et al. [2024], we consider four real-world settings of binary classification problem: **Ionosphere** ($\dim = 35$, $M = 351$), **Sonar** ($\dim = 61$, $M = 208$), **German Credit** ($\dim = 25, M = 1000$), **Breast Cancer** ($\dim = 31$, $M = 569$). Each of these datasets is randomly split into a training subset $\mathcal{D}_{\text{train}}$ of size $0.8M$ and a test subset $\mathcal{D}_{\text{test}}$ of size $0.2M$. In this setting, we define the target distribution $\pi = p(w, b|\mathcal{D}_{\text{train}})$ and evaluate the sampling quality by computing the (unnormalized) average predictive log-likelihood $\log p(w, b|\mathcal{D}_{\text{test}})$, which is expected to be maximized.

In each case, the prior is carefully chosen as a Gaussian distribution with the following parameters:

- **Ionosphere**: $p(w) = \text{N}(0, 5.25\,\text{I}_{34})$ and $p(b) = \text{N}(4.25, 0.25^2)$,
- **Sonar**: $p(w) = \text{N}(0, 4.5\,\text{I}_{60})$ and $p(b) = \text{N}(-2.5, 0.5^2)$,
- **German Credit**: $p(w) = \text{N}(0, 1.25\,\text{I}_{24})$ and $p(b) = \text{N}(3.25, 0.5^5)$,
- **Breast Cancer**: $p(w) = \text{N}(0, 3.75\,\text{I}_{30})$ and $p(b) = \text{N}(31, 2^2)$.

We draw the reader's attention to the fact that these posterior distributions don't exhibit explicit multi-modality features. Therefore, for EBM-LRDS, the tilting EBM parameterization is rather based on a Gaussian approximation of the target distribution than a GMM approximation. The initial point of the MCMC samplers are sampled from the prior distribution.

D.8.2 Computational comparison of the presented methods

This section aims at comparing the computational budget of each algorithm. In particular, we track the number of neural network evaluations and the number of target evaluations. For each algorithm, we take the following notations.

- For SMC : K (number of annealing steps), M (number of MCMC steps per level)
- For RE : M (total number of MCMC steps)
- For RDS (PIS/DDS/LRDS), DIS and CMCD : N (number of training steps), K (number of discretization steps)
- For PDDS : N (number of training steps), M (number of MCMC steps per level), K (number of discretization steps), A (number of adaptation steps)
- For iDEM : N (number of training steps), K (number of discretization steps), A (number of adaptation steps)

We detail the complexities of each algorithm in Table D.2. Note that, in LRDS, the complexity of MCMC sampling (to compute $\hat{\pi}^{\text{ref}}$) or the complexity of learning the reference process were ignored as they are negligeable compared to the training and sampling budgets; see Appendix D.8.3 for details.

D.8.3 Hyper-parameters of each sampling algorithm

In this section, we detail every hyper-parameter for each algorithm involved in the computations of the results in Section 5.5. The computationally-related parameters were chosen to ensure a comparable execution clock-time between the different algorithms on the same hardware.

Construction of the MCMC dataset for $\hat{\pi}^{\mathbf{ref}}$. We build the dataset of reference samples by running MCMC samplers initialized in the modes of the target distribution. We run 4 chains per mode. We use the Metropolis-Adjusted Langevin algorithm (MALA), see Algorithm 8, as default sampler, except for the Checkerboard distribution where we use the Random Walk Metropolis Hastings (RWMH) algorithm as the score is constant. In both cases, we adapt the step size automatically for 8192 warmup steps to aim at a 70% acceptance rate. The datasets are 60000 samples long.

Annealed MCMC methods (SMC, RE). For both algorithms, we incorporate prior knowledge into the base distribution by taking $\pi^{\text{base}} = \mathrm{N}(\mathbf{m}, \Sigma)$ where $\mathbf{m}$ and Σ are computed using maximum likelihood on samples from $\hat{\pi}^{\text{ref}}$. In each case, the annealed density path is a linearly-spaced geometric interpolation, where $\pi_k \propto (\pi^{\text{base}})^{(K-k)/K}\pi^{k/K}$. For SMC (see Algorithm 10), we use $K = 128$ annealing levels with $N = 1024$ particles and $L = 64$ MCMC steps per level (with a 4096 steps warmup) and set the ESS resampling threshold as $\alpha = 1$. For RE (see Algorithm 12), we use $K = 128$ annealing levels with 256 independent chains per level and a swapping frequency every $S = 8$ steps for a total of $M = 32$ steps (with a 8192 steps warmup). Both algorithms use MALA as MCMC backbone with a step-size automatically tuned to achieve 70% acceptance rate.

Variational diffusion-based methods (PIS, DDS, DIS, CMCD, LRDS). For PIS and DDS we use the implementation provided by Richter et al. [2023], while we use the implementation provided by Blessing et al. [2024] for DIS and CMCD. For PIS, DDS and DIS, we set the hyper-parameter σ as advised by Appendix D.9.2 by computing the mean and the variance of the samples from $\hat{\pi}^{\text{ref}}$ to add prior knowledge. For CMCD, we sample a version of π which is shifted by $\mathbf{m}$, defined as the mean computed on samples from $\hat{\pi}^{\text{ref}}$, and we set σ as the square root of the average variance estimated on those samples. For PIS, DIS and DDS, we use $K = 100$ linearly spaced time discretization steps while CMCD learns its own time discretization grid of size $K = 128$ with a final time set at $T = 10$.

For all of them, we perform 4096 optimization steps with a batch of size 2048. The neural network at stake is a Fourier MLP, as in Zhang and Chen [2022], with 4 layers of width 64. In the case of PIS, DDS and DIS, we use a target-informed parameterization by adding $\mathrm{NN}(t) \times \nabla \log \pi(x)$ (where NN is a time-dependent scalar neural network) to the Fourier MLP, as suggested by the respective authors. As recommended by Vargas et al. [2024], we do not consider this extra-parameterization in CMCD, since the drift of the generative process is already informed by π. We highlight that, by default, we do not use this target-informed parameterization in LRDS, since the reference process is specifically designed to be it-self target-informed, hence avoiding useless evaluations of the target score. Additionally, as recommended by Zhang and Chen [2022], we design the LRDS guidance network such that $\phi^{\theta_0} = 0$. This ensures that the very first sampling phase is driven solely using the reference process.

For all diffusion-based algorithms and target distributions, we ensured that the noising schedules were chosen to ensure that the target distribution gets successfully noised at time T. For LRDS, we use an exponential integration of the respective time-reversed SDE as default transition kernel; see Appendix D.4.3 and Appendix D.4.3.

Adaptive diffusion-based methods (iDEM, PDDS). For iDEM, we use the implementation provided by the original paper [Akhound-Sadegh et al., 2024]. In order to provide prior information leveraging $\hat{\pi}^{\text{ref}}$, (i) we standardize the target distribution by using the empirical mean and variance of $\hat{\pi}^{\text{ref}}$ (ii) we preload the buffer with samples from $\hat{\pi}^{\text{ref}}$. Following the design choices of the original paper, we use a Variance Exploding noising scheme with a geometric variance schedule σ_t. Because we standardized the target distribution, we decided to take $\sigma_T = 5$ for each target distribution to ensure that the distribution is properly noised. We used the same hyper-parameters as in their Gaussian mixture experiment with 400 epochs instead of 1000 to ensure computational fairness. For PDDS, we also use the implementation provided by the original paper [Phillips et al., 2024]. In order to provide prior information leveraging $\hat{\pi}^{\text{ref}}$, we standardized the target distribution by leveraging the empirical mean and variance of $\hat{\pi}^{\text{ref}}$ thus bypassing the original VI step of their algorithm. We also significantly increased the default computational budget by using 128 discretization steps, 2048 particles and training for 100000 steps with a batch size of 512. The rest of the hyper-parameters are the default ones.

LRDS reference fitting details. For GMM-LRDS, the EM algorithm is taken from Pedregosa et al. [2011]. We use a diagonal covariance for Rings distribution, Checkerboard distribution, diagonal Gaussian mixtures and Bayesian posterior distributions, and use a full covariance otherwise. In the full covariance case, we regularize the covariance to ensure their positivity. For EBM-RDS, we fit multi-level EBMs with Algorithm 14 using the Replica Exchange annealed sampler (see Algorithm 12) as backbone. This sampler has the advantage of being massively parallel and thus suited for GPU computations. We perform 200 epochs with a batch size of 32 for each noise level. To increase gradient accuracy and enhance efficiency, we accumulate the gradients over 10 steps where we keep the same negative samples but update the positive ones. We perform 32 MCMC steps to sample a batch of negative samples with a swap happening every 8 steps. We keep 16 MCMC steps out of 32 to compute the expectations. To compensate the short length of our MCMC chains, we leverage a persistent buffer to kickstart the chains at each noise level. Lastly, we utilise the GMM tilting EBM parameterization detailed at the end of Appendix D.6 where the neural network parameterized as $x \mapsto \mathrm{NN}(t,x)^T x$ as suggested by Salimans and Ho [2021]. We leverage the GMM tilting initialization of the EBM to perform exact MCMC sampling at the very first gradient step. Moreover, we preconditioned the network as advised by Karras et al. [2022] by leveraging the Gaussian approximation of $\hat{\pi}^{\mathrm{ref}}$.

D.9 Additional results

D.9.1 Link between the target and the reference processes

Let $\pi^{\mathrm{ref}} \in \mathscr{P}(\mathbb{R}^d)$ be an arbitrary distribution. We recall that under mild assumptions on π^{ref}, see e.g., Cattiaux et al. [2023], the time-reversal of $\mathbb{P}^{\mathrm{ref}}$ is associated to the SDE

$$\mathrm{d}Y_t^{\mathrm{ref}} = -f(T-t)Y_t^{\mathrm{ref}}\mathrm{d}t + g^2(T-t)s_{T-t}^{\mathrm{ref}}(Y_t^{\mathrm{ref}})\mathrm{d}t + g(T-t)\mathrm{d}B_t \ , \ Y_0^{\mathrm{ref}} \sim \mathbb{P}_T^{\mathrm{ref}} \ . \tag{D.159}$$

Link with Schrödinger Bridge. Since $\mathbb{P}^\star$ and $\mathbb{P}^{\mathrm{ref}}$ are based on the same noising diffusion process, these path measures are linked by the relation $\mathbb{P}^\star = \pi \otimes \mathbb{P}^{\mathrm{ref}}_{|0}$. Hence, using the KL chain rule [Léonard, 2014a], it is clear that $\mathbb{P}^\star$ is solution to the following minimization problem over path measure space

$$\arg\min\{\mathrm{KL}(\mathbb{P} \mid \mathbb{P}^{\mathrm{ref}}) \ : \ \mathbb{P} \in \mathscr{P}(\mathbf{C}_T), \mathbb{P}_0 = \pi\} \ , \tag{D.160}$$

which is often referred to as a half-Schrödinger bridge problem in stochastic control literature.

Link with Doob's h-transform. This relation on path measure space can also be written as $\mathbb{P}^\star = \frac{\mathrm{d}\pi}{\mathrm{d}\pi^{\mathrm{ref}}} \cdot \mathbb{P}^{\mathrm{ref}}$, see Appendix D.4.1. Therefore, solely based on the SDE (D.159) describing $(\mathbb{P}^{\mathrm{ref}})^R$, $(\mathbb{P}^\star)^R$ may be expressed as a Doob's h-transform of $(\mathbb{P}^{\mathrm{ref}})^R$ via the SDE

$$\mathrm{d}Y_t = -f(T-t)Y_t\mathrm{d}t + g^2(T-t)\{s_{T-t}^{\mathrm{ref}}(Y_t) + h_{T-t}(Y_t)\}\mathrm{d}t + g(T-t)\mathrm{d}B_t \ , Y_0 \sim \mathbb{P}_T^\star \ , \tag{D.161}$$

where the Doob's control function $h : [0,T] \times \mathbb{R}^d \to \mathbb{R}^d$ is defined, for any $y_t \in \mathbb{R}^d$ and any $t \in [0,T]$, by

$$h_{T-t}(y_t) = \nabla \log \mathbb{E}_{(\mathbb{P}^{\mathrm{ref}})^R_{T|t}}\left[\frac{\mathrm{d}\pi}{\mathrm{d}\pi^{\mathrm{ref}}}(Y_T) \mid Y_t = y_t\right] = \nabla \log \mathbb{E}_{\mathbb{P}^{\mathrm{ref}}_{0|T-t}}\left[\frac{\mathrm{d}\pi}{\mathrm{d}\pi^{\mathrm{ref}}}(X_0) \mid X_t = y_t\right] \ . \tag{D.162}$$

Additionally, we have for any $x_t \in \mathbb{R}^d$,

$$\begin{aligned}
\frac{p_t^\star}{p_t^{\mathrm{ref}}}(x_t) &= \frac{1}{p_t^{\mathrm{ref}}(x_t)} \int_{\mathbb{R}^d} p_{t|0}^\star(x_t|x_0)\mathrm{d}\pi(x_0) \\
&= \int_{\mathbb{R}^d} \frac{\mathrm{d}\pi}{\mathrm{d}\pi^{\mathrm{ref}}}(x_0) \frac{p_{t|0}^{\mathrm{ref}}(x_t|x_0)\mathrm{d}\pi^{\mathrm{ref}}(x_0)}{p_t^{\mathrm{ref}}(x_t)} \\
&= \mathbb{E}_{\mathbb{P}^{\mathrm{ref}}_{0|t}}\left[\frac{\mathrm{d}\pi}{\mathrm{d}\pi^{\mathrm{ref}}}(X_0) \mid X_t = x_t\right] \ .
\end{aligned}$$

By combining previous computations, we obtain $h_t = \nabla \log p_t^\star / p_t^{\mathrm{ref}} = \phi_t$ for any $t \in [0,T]$.

D.9.2 Optimal setting of isotropic Gaussian reference distribution

The goal of this section is to explain how to set the hyperparameter $\sigma \in (0,\infty)$ in PIS and DDS settings so as to 'optimize' the reference distribution $\pi^{\mathrm{ref}} = \mathrm{N}(0, \sigma^2 \mathrm{I}_d)$, when targeting a multi-modal distribution. The

same reasoning can also be applied in the DIS setting to set the base distribution π^{base}, chosen as an isotropic Gaussian distribution.

Let $\pi \in \mathscr{P}(\mathbb{R}^d)$ be our target distribution. Assume that we are given a *diagonal Gaussian approximation* $\tilde{\pi} = \mathrm{N}(\mathbf{m}, \Sigma)$ of π, with mean $\mathbf{m} \in \mathbb{R}^d$ and covariance $\Sigma = \mathrm{diag}(\Gamma^2) \in \mathbb{R}^{d \times d}$, where $\Gamma \in (0, \infty)^d$. We propose to set π^{ref} as 'close' as possible to $\tilde{\pi}$ in PIS and DDS by solving the following variational problem

$$\arg\min\{\mathrm{KL}(\tilde{\pi} \mid \mathrm{N}(0, \sigma^2 \mathrm{I}_d)) \,:\, \sigma \in (0, \infty)\} \,. \tag{D.163}$$

Let $\sigma > 0$. We have

$$\mathrm{KL}(\tilde{\pi} \mid \mathrm{N}(0, \sigma^2 \mathrm{I}_d)) = \frac{1}{2}\left[\log \frac{|\sigma^2 \mathrm{I}_d|}{|\Sigma|} - d + \sigma^{-2}\mathbf{m}^T\mathbf{m} + \mathrm{Tr}(\sigma^{-2}\Sigma)\right] \tag{D.164}$$

$$= \frac{1}{2}\left[d\log(\sigma^2) - \sum_{j=1}^{d}\log\Gamma_j^2 - d + \sigma^{-2}\mathbf{m}^T\mathbf{m} + \sigma^{-2}\sum_{j=1}^{d}\Gamma_j^2\right] \,. \tag{D.165}$$

Moreover, the derivative with respect to σ^2 is given by

$$\frac{\partial}{\partial\sigma^2}\mathrm{KL}(\tilde{\pi} \mid \mathrm{N}(0, \sigma^2 \mathrm{I}_d)) = \frac{1}{2}\left[d\sigma^{-2} - \sigma^{-4}\left(\mathbf{m}^T\mathbf{m} + \sum_{j=1}^{d}\Gamma_j^2\right)\right] \,. \tag{D.166}$$

In particular, the optimal solution of this variational problem is

$$\sigma_\pi^2 = d^{-1}\left(\mathbf{m}^T\mathbf{m} + \sum_{j=1}^{d}\Gamma_j^2\right) \,, \tag{D.167}$$

which motivates us to set $\pi^{\text{ref}} = \mathrm{N}(0, \sigma_\pi^2 \mathrm{I}_d)$.

In practice, when we do not have samples from the target distribution, we rather rely on a diagonal Gaussian approximation of the empirical distribution $\hat{\pi}^{\text{ref}}$ (obtained with local MCMC samplers) and set $\pi^{\text{ref}} = \mathrm{N}(0, \sigma^2_{\hat{\pi}^{\text{ref}}} \mathrm{I}_d)$.

Application to Gaussian mixtures. In the particular case where π is a Gaussian mixture, we have an analytical formula for $\tilde{\pi}$.

Let $\pi = \sum_{j=1}^{J} w_j \mathcal{N}(\mathbf{m}_j, \Sigma_j)$ be our target distribution with $w_{1:J} \in \Delta_J$, where the j-th component has mean $\mathbf{m}_j \in \mathbb{R}^d$ and covariance $\Sigma_j = \mathrm{diag}(\lambda_j) \in \mathbb{R}^{d \times d}$ with $\lambda_j \in (0, \infty)^d$. Consider $X \sim \pi$ and $Z \sim \mathcal{M}(w_1, \ldots, w_J)$. We first have

$$\mathbb{E}[X] = \mathbb{E}\left[\mathbb{E}\left[X \mid Z\right]\right] = \sum_{j=1}^{J} w_j \mathbf{m}_j \,. \tag{D.168}$$

We define the *diagonal variance* of π by $\mathrm{diagVar}(\pi) = (\mathrm{Var}[X]_1, \ldots, \mathrm{Var}[X]_d) \in (0, \infty)^d$ where $\mathrm{Var}[X]_i = \mathbb{E}[(X_i - \mathbb{E}[X]_i)^2]$ for any $i \in \{1, \ldots, d\}$. By the law of total variance, we have

$$\mathrm{Var}[X]_i = \mathbb{E}\left[\mathrm{Var}\left[X \mid Z\right]_i\right] + \mathrm{Var}\left[\mathbb{E}\left[X \mid Z\right]\right]_i \tag{D.169}$$

$$= \sum_{j=1}^{J} w_j (\lambda_j^2)_i + \sum_{j=1}^{J} w_j \left((\mathbf{m}_j)_i - \mathbb{E}\left[X\right]_i\right)^2 \,. \tag{D.170}$$

By characterisation of Gaussian distributions, $\mathrm{N}(\mathbb{E}[X], \mathrm{diagVar}(\pi))$ is the closest diagonal Gaussian distribution to π in the Kullback-Leibler sense. Therefore, in this particular setting, $\tilde{\pi}$ may be defined by $\mathbf{m} = \mathbb{E}[X]$ and $\Gamma^2 = \mathrm{diagVar}(\pi)$. We use this approach when computing σ_π^2 in the numerical experiment commented in Figure 5.1.

D.9.3 Failure of local MCMC samplers on multimodal distributions

As we show in Figure D.1, *local* MCMC samplers such as the *Random Walk Metropolis Hastings* (RWMH) algorithm [Metropolis et al., 1953], the *Metropolis-adjusted Langevin Algorithm* (MALA) [Roberts and Tweedie, 1996], the *Hamiltonian Monte Carlo* (HMC) algorithm [Duane et al., 1987, Brooks et al., 2011] or the *No-U Turn Sampler* (NUTS) [Hoffman and Gelman, 2014] tend to produce Markov chains that get trapped in modes. The resulting samples are thus not representative of the global landscape.

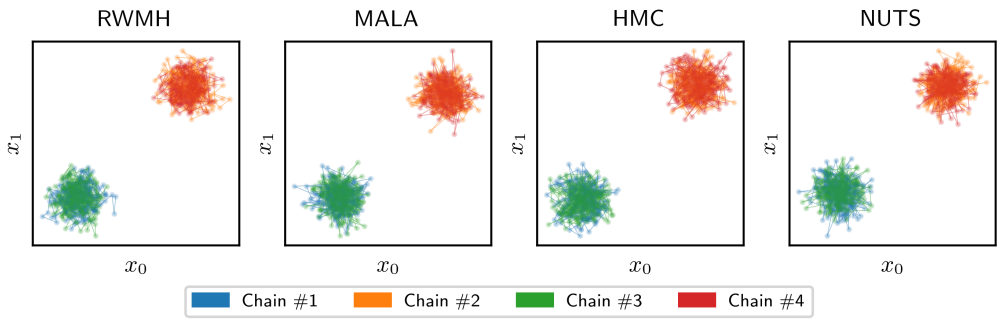


Figure D.1: **Samples from different MCMC samplers when sampling from a bi-modal Gaussian mixture in 2 dimensions**. There are 4 different chains (in different colors). The MCMC samplers ran for 4096 warmup steps before producing those 1024 samples. We only display 1 sample every 4 steps.

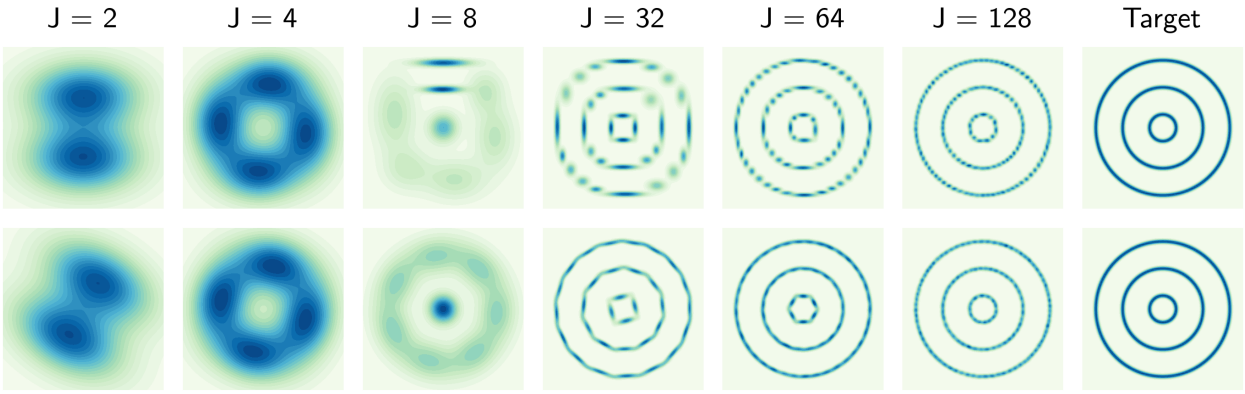


Figure D.2: **Increasingly expressive GMM approximations of the rings distribution. (Top)** Approximation using diagonal covariances **(Bottom)** Approximation using full covariances

D.9.4 Limitations of Gaussian Mixture Models

Figure D.2 shows the progressive GMM approximation of the Rings distribution by leveraging an increasing number of components. This figure shows that, in this case, getting a good approximation of the distribution requires significantly more components that the number of modes (here, 3), thus highlighting Gaussian Mixture models' approximation power limitations.

D.9.5 Further experimental results

In this section, we display detailed results for the experimental settings presented in Section 5.5, as well as additional results that assess the robustness and the superiority of LRDS compared to its competitors for a large diversity of settings. When ground truth samples are available (*i.e.*, in all settings except ϕ^4 and Bayesian regression), we display statistical estimations of the integral probability metrics defined in Appendix D.2, namely, the regularized Wasserstein distance with regularization $\varepsilon = 10^{-3}$ ($W_{2,\varepsilon}$), the Maximum Mean Discrepancy (MMD) and the sliced Kolmogorov-Smirnov distance (KS).

High-dimensional bi-modal Gaussian mixtures

Below, we display the results of all considered samplers, including LRDS, when targeting the bi-modal Gaussian mixtures defined in Appendix D.8.1 with increasing dimension $d \in \{8, 16, 32, 64\}$. To assess the sampling quality at a global level, we compute for each sampler the absolute error 'Mode Err.' when estimating the strongest mode weight (in %, comprised between 0%, the best, and 66.7%, the worst) via Monte Carlo approximation. All metrics are computed based on 8192 samples.

Table D.3: **Results for bi-modal Gaussian mixtures with diagonal covariance and medium conditioning**, averaged over 16 sampling runs (same setting as Table 5.2). Bold font indicates best result, orange cells refer to settings with uninformative mode weight estimation (*i.e.*, uniform mixture), red cells denote mode collapse. N/A denotes settings where results could not be obtained due to numerical issues. The MMD and KS results are displayed with 100-factor rescaling.

	$d = 16$				$d = 32$				$d = 64$			
Algorithm	Mode Err. ↓	$W_{2,\varepsilon}$ ↓	MMD ↓	KS ↓	Mode Err. ↓	$W_{2,\varepsilon}$ ↓	MMD ↓	KS ↓	Mode Err. ↓	$W_{2,\varepsilon}$ ↓	MMD ↓	KS ↓
SMC	11.4%±9.1%	0.38±0.00	13.91±10.68	10.80±7.84	15.8%±8.5%	0.70±0.01	18.68±9.67	14.63±7.59	15.2%±7.5%	1.16±0.01	17.87±8.72	14.14±6.71
RE	16.5%±1.3%	0.38±0.00	20.57±1.78	15.20±1.31	15.9%±1.4%	0.70±0.00	19.23±2.03	14.54±1.55	17.0%±1.4%	1.17±0.00	20.36±1.83	15.68±1.39
LV-PIS	6.0%±3.4%	0.43±0.00	7.63±3.07	6.86±2.49	33.2%±0.1%	1.01±0.03	34.90±0.58	28.43±1.52	33.0%±0.1%	1.62±0.02	34.03±0.71	29.33±0.75
LV-DDS	11.8%±9.3%	0.40±0.01	13.83±10.23	11.29±8.06	31.5%±2.9%	0.86±0.04	33.96±3.29	28.00±2.65	33.1%±0.1%	1.49±0.02	34.45±0.61	28.95±0.89
LV-DIS	16.4%±0.50%	0.42±0.00	21.36±0.75	16.16±0.54	16.5%±0.40%	0.77±0.00	21.13±0.96	16.18±0.76	16.8%±0.50%	1.24±0.01	20.71±0.81	15.93±0.56
LV-CMCD	32.2%±15.4%	0.60±0.06	46.89±17.35	33.46±13.03	50.1%±8.8%	1.96±0.52	76.17±13.71	48.79±8.03	16.3%±10.6%	3.47±0.31	53.15±1.85	41.79±2.64
iDEM	33.3%±0.0%	1.75±0.17	53.82±0.36	36.86±0.92	66.7%±0.0%	4.16±0.22	85.52±0.53	61.10±1.26	11.7%±0.4%	117.82±0.14	90.13±0.12	N/A
PDDS	**0.8%±0.6%**	0.40±0.00	**1.66±0.68**	**2.59±0.29**	66.7%±0.0%	11.22±0.08	105.11±0.24	N/A	N/A	N/A	N/A	N/A
GMM-LRDS	1.7%±0.6%	0.38±0.00	2.58±0.96	2.69±0.66	**2.7%±0.8%**	0.71±0.00	**3.64±1.07**	**3.38±0.76**	**4.1%±0.6%**	1.19±0.00	**5.25±0.98**	**4.50±0.70**

Table D.4: **Results for bi-modal Gaussian mixtures with diagonal covariance and isotropic conditioning**, averaged over 16 sampling runs. Bold font indicates best result, orange cells refer to settings with uninformative mode weight estimation (*i.e.*, uniform mixture), red cells denote mode collapse. N/A denotes settings where results could not be obtained due to numerical issues. The MMD and KS results are displayed with 100-factor rescaling.

	$d = 16$				$d = 32$				$d = 64$			
Algorithm	Mode Err. ↓	$W_{2,\varepsilon}$ ↓	MMD ↓	KS ↓	Mode Err. ↓	$W_{2,\varepsilon}$ ↓	MMD ↓	KS ↓	Mode Err. ↓	$W_{2,\varepsilon}$ ↓	MMD ↓	KS ↓
SMC	16.0%±10.1%	0.65±0.01	19.11±12.55	14.06±8.74	12.3%±9.6%	1.18±0.01	14.57±11.46	11.03±8.03	11.0%±8.8%	1.94±0.00	12.98±10.51	9.80±7.38
RE	15.2%±1.2%	0.66±0.00	18.72±1.66	13.42±1.23	16.1%±1.3%	1.19±0.00	19.50±1.70	14.14±1.26	16.5%±1.3%	1.94±0.00	19.46±1.71	14.39±1.24
LV-PIS	1.9%±1.2%	0.66±0.00	2.66±1.45	2.63±0.89	1.3%±0.6%	1.20±0.00	2.15±0.68	2.23±0.42	2.8%±0.6%	1.98±0.00	2.94±0.90	2.81±0.61
LV-DDS	**0.7%±0.5%**	0.65±0.00	**0.86±0.70**	**1.59±0.30**	**0.8%±0.5%**	1.19±0.00	**1.09±0.60**	**1.56±0.24**	1.6%±0.8%	1.95±0.00	**2.13±0.97**	**2.23±0.61**
LV-DIS	16.4%±1.3%	0.66±0.00	19.99±1.73	14.16±1.21	16.7%±0.7%	1.20±0.00	20.12±1.00	14.54±0.70	16.6%±0.6%	1.96±0.00	19.79±0.90	14.59±0.65
LV-CMCD	0.90%±0.60%	0.68±0.00	2.84±0.55	2.92±0.45	1.7%±0.7%	1.24±0.00	2.77±0.36	2.76±0.36	1.1%±0.8%	2.05±0.00	3.43±0.75	3.15±0.65
iDEM	66.7%±0.0%	2.23±0.17	83.92±0.50	59.68±2.16	33.3%±0.0%	2.48±0.15	49.83±0.35	35.02±0.62	**0.3%±0.2%**	179.86±0.12	97.17±1.05	N/A
PDDS	0.9%±0.6%	0.70±0.00	1.27±1.25	2.50±0.49	0.9%±0.6%	1.22±0.00	1.53±0.87	2.41±0.31	0.9%±0.7%	1.96±0.00	0.85±1.25	2.05±0.56
GMM-LRDS	1.6%±0.6%	0.66±0.00	2.07±0.99	2.23±0.54	2.8%±0.5%	1.19±0.00	3.72±0.89	3.23±0.58	4.6%±0.6%	1.96±0.00	5.65±0.88	4.56±0.61

Table D.5: **Results for bi-modal Gaussian mixtures with diagonal covariance and hard conditioning**, averaged over 16 sampling runs. Bold font indicates best result, orange cells refer to settings with uninformative mode weight estimation (*i.e.*, uniform mixture), red cells denote mode collapse. N/A denotes settings where results could not be obtained due to numerical issues. The MMD and KS results are displayed with 100-factor rescaling.

	$d = 16$				$d = 32$				$d = 64$			
Algorithm	Mode Err. ↓	$W_{2,\varepsilon}$ ↓	MMD ↓	KS ↓	Mode Err. ↓	$W_{2,\varepsilon}$ ↓	MMD ↓	KS ↓	Mode Err. ↓	$W_{2,\varepsilon}$ ↓	MMD ↓	KS ↓
SMC	12.4%±8.3%	0.24±0.00	15.40±10.16	12.00±7.57	16.5%±9.5%	0.47±0.00	20.22±11.66	15.71±8.58	11.4%±9.8%	0.80±0.00	13.88±11.92	11.16±8.79
RE	16.3%±1.4%	0.25±0.00	20.76±1.95	15.40±1.38	17.0%±1.7%	0.47±0.00	21.31±2.19	16.11±1.58	16.4%±1.4%	0.81±0.00	19.99±1.75	15.54±1.38
LV-PIS	8.4%±3.4%	0.45±0.00	13.52±2.50	12.48±2.01	32.2%±0.3%	1.29±0.02	33.89±0.60	29.92±0.57	32.5%±0.2%	2.24±0.04	33.91±0.76	30.49±0.61
LV-DDS	24.7%±8.8%	0.32±0.03	26.89±9.03	22.92±7.59	40.5%±13.9%	0.73±0.02	44.97±19.24	37.20±12.88	38.1%±15.4%	1.59±0.04	42.64±20.91	35.46±13.67
LV-DIS	16.6%±0.6%	0.38±0.02	23.00±0.87	19.69±0.69	16.7%±0.7%	0.65±0.03	23.01±1.12	19.33±0.99	16.8%±0.6%	1.06±0.02	22.94±0.84	18.97±0.65
LV-CMCD	21.5%±17.8%	4.37±1.19	62.31±7.46	N/A	31.5%±2.2%	1.54±0.35	46.02±13.29	37.50±1.30	44.1%±15.3%	0.87±0.03	51.17±19.36	41.30±14.11
iDEM	33.3%±0.0%	2.12±0.11	51.43±0.33	38.07±1.02	66.7%±0.0%	7.53±0.52	85.77±0.63	55.56±1.03	33.3%±0.0%	175.16±0.83	102.13±0.20	N/A
PDDS	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
GMM-LRDS	**2.1%±1.0%**	0.25±0.00	**3.41±1.26**	**3.46±0.90**	**1.7%±0.9%**	0.47±0.00	**3.00±1.12**	**2.99±0.80**	**4.5%±1.8%**	0.82±0.00	**6.12±2.21**	**5.25±1.62**

- Table D.3 corresponds to Gaussian mixtures with diagonal covariance and medium conditioning, completing the results of Table 5.2 presented in the main paper,
- Table D.4 corresponds to Gaussian mixtures with diagonal covariance and isotropic conditioning,
- Table D.5 corresponds to Gaussian mixtures with diagonal covariance and hard conditioning,
- Table D.6 corresponds to Gaussian mixtures with full covariance and medium conditioning,
- Table D.7 corresponds to Gaussian mixtures with full covariance and hard conditioning.

For all of these bi-modal settings, we observe that the mode weight estimation error is consistent with the values of probability metrics (except for the regularized Wasserstein distance, which is not discriminative between the methods). In particular, GMM-LRDS is on par or superior to all competitors in each setting, except the least challenging setting ('Isotropic' conditioning, see Table D.4), where LV-DDS is more performant.

Table D.6: **Results for bi-modal Gaussian mixtures with full covariance and medium conditioning**, averaged over 16 sampling runs. Bold font indicates best result, orange cells refer to settings with uninformative mode weight estimation (*i.e.*, uniform mixture), red cells denote mode collapse. N/A denotes settings where results could not be obtained due to numerical issues. The MMD and KS results are displayed with 100-factor rescaling.

	$d = 8$				$d = 16$				$d = 32$			
Algorithm	Mode Err. ↓	$W_{2,\varepsilon}$ ↓	MMD ↓	KS ↓	Mode Err. ↓	$W_{2,\varepsilon}$ ↓	MMD ↓	KS ↓	Mode Err. ↓	$W_{2,\varepsilon}$ ↓	MMD ↓	KS ↓
SMC	17.3%±7.1%	0.17±0.00	21.55±8.84	16.07±6.33	13.5%±4.5%	0.38±0.00	16.35±5.47	12.52±4.05	10.3%±5.8%	0.70±0.00	12.18±6.67	9.72±5.15
RE	16.6%±1.0%	0.17±0.00	21.34±1.54	15.30±1.11	16.4%±1.4%	0.38±0.00	20.65±1.90	15.23±1.34	16.3%±1.2%	0.70±0.00	19.90±1.74	15.04±1.24
LV-PIS	6.8%±2.4%	0.20±0.00	9.89±2.78	8.67±1.98	25.4%±12.6%	0.52±0.06	28.21±13.13	23.37±10.61	33.1%±0.2%	1.07±0.39	35.02±0.70	27.85±3.09
LV-DDS	**1.6%±0.8%**	0.17±0.00	**2.64±0.87**	**2.63±0.53**	3.3%±0.9%	0.39±0.00	4.83±1.08	4.27±0.75	4.0%±1.1%	0.75±0.00	6.85±1.23	5.95±1.01
LV-DIS	16.5%±0.5%	0.22±0.00	22.50±0.88	17.33±0.66	16.3%±1.1%	0.46±0.02	21.65±1.55	16.93±0.13	16.7%±0.6%	0.82±0.01	21.95±0.94	17.16±0.72
LV-CMCD	16.6%±0.5%	0.18±0.00	21.76±1.04	16.06±0.81	20.8%±12.7%	1.23±0.96	36.92±5.06	26.70±4.59	16.6%±0.50%	0.75±0.00	20.89±0.95	15.90±0.71
iDEM	66.7%±0.0%	2.06±0.19	85.27±0.40	61.42±1.09	66.7%±0.0%	3.95±0.20	84.48±0.53	62.98±1.10	66.7%±0.0%	6.89±0.24	85.60±0.41	62.87±1.31
PDDS	**1.3%±1.1%**	0.19±0.00	**1.52±1.58**	**2.65±0.77**	**1.9%±0.9%**	0.40±0.00	**2.63±1.53**	**3.09±0.77**	**0.7%±0.6%**	0.73±0.00	**1.56±0.89**	**2.61±0.37**
GMM-LRDS	**1.3%±0.6%**	0.17±0.00	**1.97±0.85**	**2.36±0.52**	**1.8%±0.5%**	0.38±0.00	**2.65±0.87**	**2.72±0.58**	2.4%±0.5%	0.71±0.00	3.48±0.89	3.24±0.61

Table D.7: **Results for bi-modal Gaussian mixtures with full covariance and hard conditioning**, averaged over 16 sampling runs. Bold font indicates best result, orange cells refer to settings with uninformative mode weight estimation (*i.e.*, uniform mixture), red cells denote mode collapse. N/A denotes settings where results could not be obtained due to numerical issues. The MMD and KS results are displayed with 100-factor rescaling.

	$d = 8$				$d = 16$				$d = 32$			
Algorithm	Mode Err. ↓	$W_{2,\varepsilon}$ ↓	MMD ↓	KS ↓	Mode Err. ↓	$W_{2,\varepsilon}$ ↓	MMD ↓	KS ↓	Mode Err. ↓	$W_{2,\varepsilon}$ ↓	MMD ↓	KS ↓
SMC	14.2%±15.0%	0.10±0.00	17.83±19.13	13.83±13.58	10.1%±6.9%	0.24±0.00	12.91±8.77	10.03±6.21	17.9%±10.2%	0.47±0.00	21.61±12.58	16.96±9.50
RE	16.1%±1.2%	0.11±0.00	20.99±1.86	15.12±1.33	16.8%±1.2%	0.25±0.00	21.54±1.76	15.89±1.22	16.3%±1.3%	0.47±0.00	20.28±1.79	15.39±1.32
LV-PIS	10.9%±1.1%	0.20±0.01	15.75±1.22	14.13±0.83	2.2%±1.5%	0.43±0.00	10.36±0.80	10.50±0.70	32.7%±0.3%	1.34±0.03	34.25±0.75	30.15±0.78
LV-DDS	3.2%±1.8%	0.12±0.01	5.53±1.63	5.36±1.05	7.5%±4.1%	0.29±0.00	11.39±4.88	9.91±3.53	10.5%±7.8%	0.62±0.00	16.62±8.01	14.10±6.05
LV-DIS	16.9%±0.5%	0.28±0.03	24.00±0.90	21.85±0.90	16.6%±0.6%	0.51±0.03	23.85±0.96	22.1±0.77	16.8%±0.6%	0.91±0.07	24.34±0.91	22.76±0.77
LV-CMCD	16.6%±0.5%	1.98±0.02	18.76±0.56	34.42±1.16	16.7%±0.6%	3.67±0.02	19.86±0.52	36.03±0.10	16.7%±0.5%	6.05±0.02	22.65±0.42	35.88±0.99
iDEM	66.6%±0.0%	0.23±0.00	84.95±0.61	60.96±1.39	66.7%±0.0%	5.49±0.12	86.81±0.47	59.71±1.59	66.7%±0.0%	10.33±0.09	81.57±0.65	53.73±1.36
PDDS	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
GMM-LRDS	**1.8%±0.7%**	0.11±0.00	**2.85±1.00**	**3.22±0.67**	**1.0%±0.7%**	0.25±0.00	**2.22±0.91**	**2.60±0.62**	**2.2%±1.3%**	0.47±0.00	**3.19±1.63**	**3.13±1.17**

Execution time. To demonstrate the applicability of LRDS in practice, we provide in Figure D.3 the execution clock-time of each sampling method for several Gaussian mixture sampling settings. The computations were all ran on the same V100 GPU. We notice that (i) the initial MCMC sampling step to build the reference dataset, then used to initialize each sampler, and the reference fitting time from GMM-LRDS are completely negligeable compared to the training time, and that (ii) the training time of GMM-LRDS is on par with previous variational diffusion-based approaches.

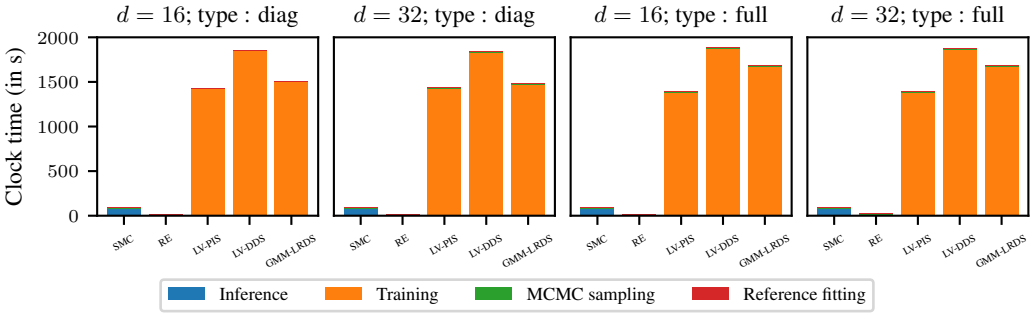


Figure D.3: **Execution clock-time for bi-modal Gaussian mixtures with *diagonal* covariance (Left) and *full* covariance (Right)**, averaged over the conditioning settings.

Ablation study on GMM-LRDS: effect of the reference distribution. To asses the robustness of GMM-LRDS with respect to the setting of the reference distribution, we conduct an ablation study that reveals the impact of modifying the location and the entropy of the reference modes. To do so, we set $d = 8$ and define the target distribution $\pi : x \in \mathbb{R}^d \mapsto (2/3)\mathrm{N}(x; -\mathbf{1}_d, 0.05) + (1/3)\mathrm{N}(x; \mathbf{1}_d, 0.05)$, as detailed in Appendix D.8.1. Instead of defining the reference distribution in GMM-LRDS as a Gaussian mixture fitted on MCMC samples, we propose to consider a flexible reference distribution given by the Gaussian mixture $x \in \mathbb{R}^d \mapsto (2/3)\mathrm{N}(x; \mathbf{m}_1, 0.05\alpha^2\mathrm{I}_d) + (1/3)\mathrm{N}(x; \mathbf{m}_2, 0.05\alpha^2\mathrm{I}_d)$ where $(\mathbf{m}_1, \mathbf{m}_2) \in \mathbb{R}^d \times \mathbb{R}^d$ and $\alpha > 0$.

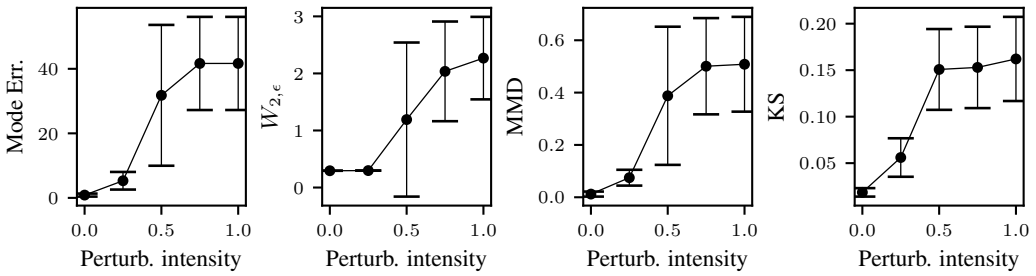


Figure D.4: **Results of GMM-LRDS for an 8-dimensional bi-modal Gaussian mixture when varying the location of the reference modes**. The performance degrades as soon as the reference modes and the target modes are further from each other.

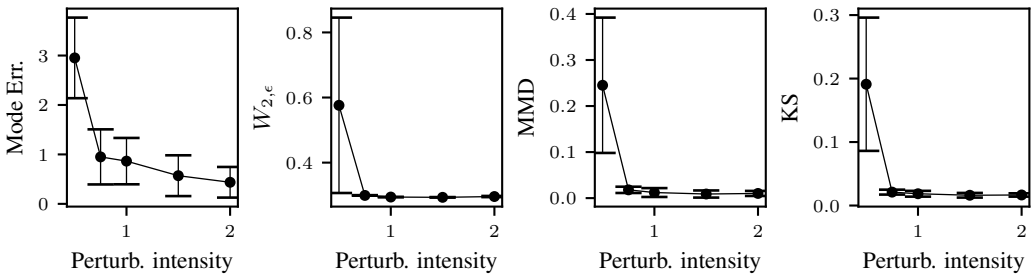


Figure D.5: **Results of GMM-LRDS for an 8-dimensional bi-modal Gaussian mixture when varying the variance of the reference modes.** Except for small reference mode variance, the performance is unchanged.

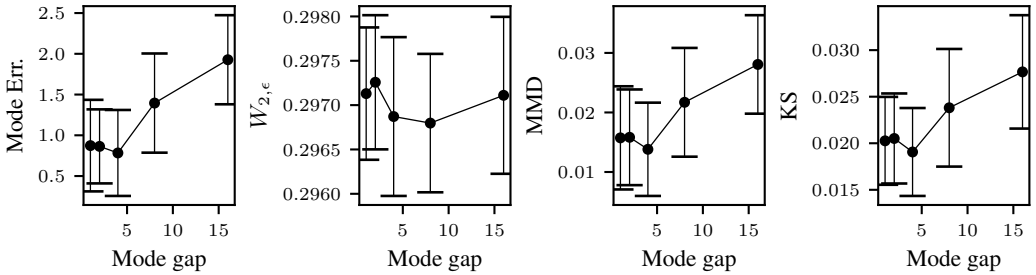


Figure D.6: **Results of GMM-LRDS for an 8-dimensional bi-modal Gaussian mixture when varying the distance between the modes of the target distribution.** The performance does not significantly degrades, despite the increasing sampling complexity.

To observe the impact of the location of the reference modes on GMM-LRDS, we take $\alpha = 1$ (same variance setting as the target) and $(\mathbf{m}_1, \mathbf{m}_2) \sim \mathrm{N}(-\mathbf{1}_d, \sigma_{\mathbf{m}}^2 \mathrm{I}_d) \otimes \mathrm{N}(\mathbf{1}_d, \sigma_{\mathbf{m}}^2 \mathrm{I}_d)$ where $\sigma_{\mathbf{m}} \in \{0, 0.25, 0.5, 0.75, 1.0\}$ is a mean perturbation parameter. Hence, the higher $\sigma_{\mathbf{m}}$, the further the reference modes might be with respect to the target modes. We display in Figure D.4 the results of GMM-LRDS for this setting, averaged over 16 sampling runs. This ablation study notably reveals the major need of precision on the location of the target modes (in practice, brought by MCMC sampling) to build the reference distribution.

On the other hand, to understand the effect of the entropy of the reference modes, we take $(\mathbf{m}_1, \mathbf{m}_2) = (-\mathbf{1}_d, \mathbf{1}_d)$ (same mean setting as the target), and $\alpha \in \{0.5, 0.75, 1, 1.5, 2\}$, which can be seen as a variance perturbation parameter. We display in Figure D.5 the results of GMM-LRDS for this setting, averaged over 16 sampling runs. Interestingly, our ablation study demonstrates that GMM-LRDS works well as soon as the support of the reference distribution includes the support of the target distribution. In practice, this is verified by MCMC sampling.

Ablation study on GMM-LRDS: effect of the distance between the target modes. To asses the robustness of GMM-LRDS with respect to the complexity of the target distribution, we conduct a second ablation study that illustrates the behaviour of GMM-LRDS for target with higher energy barrier. To do so, we set $d = 8$ and define the target distribution $\pi : x \in \mathbb{R}^d \mapsto (2/3)\mathrm{N}(x; -a\,\mathbf{1}_d, 0.05) + (1/3)\mathrm{N}(x; a\,\mathbf{1}_d, 0.05)$, where

Table D.8: **Results of GMM-LRDS for bi-modal Gaussian mixtures with diagonal covariance and various conditioning settings ($d = 128$), averaged over 16 sampling runs.** The MMD and KS results are displayed with 100-factor rescaling.

Isotropic conditioning				Medium conditioning				Hard conditioning			
Mode Err. ↓	$W_{2,\varepsilon}$ ↓	MMD ↓	KS ↓	Mode Err. ↓	$W_{2,\varepsilon}$ ↓	MMD ↓	KS ↓	Mode Err. ↓	$W_{2,\varepsilon}$ ↓	MMD ↓	KS ↓
6.6%±0.6%	3.04±0.00	7.84±0.92	6.16±0.67	6.8%±0.9%	1.86±0.00	8.33±1.29	6.81±0.98	6.2%±3.6%	1.32±0.00	8.09±3.83	6.73±2.92

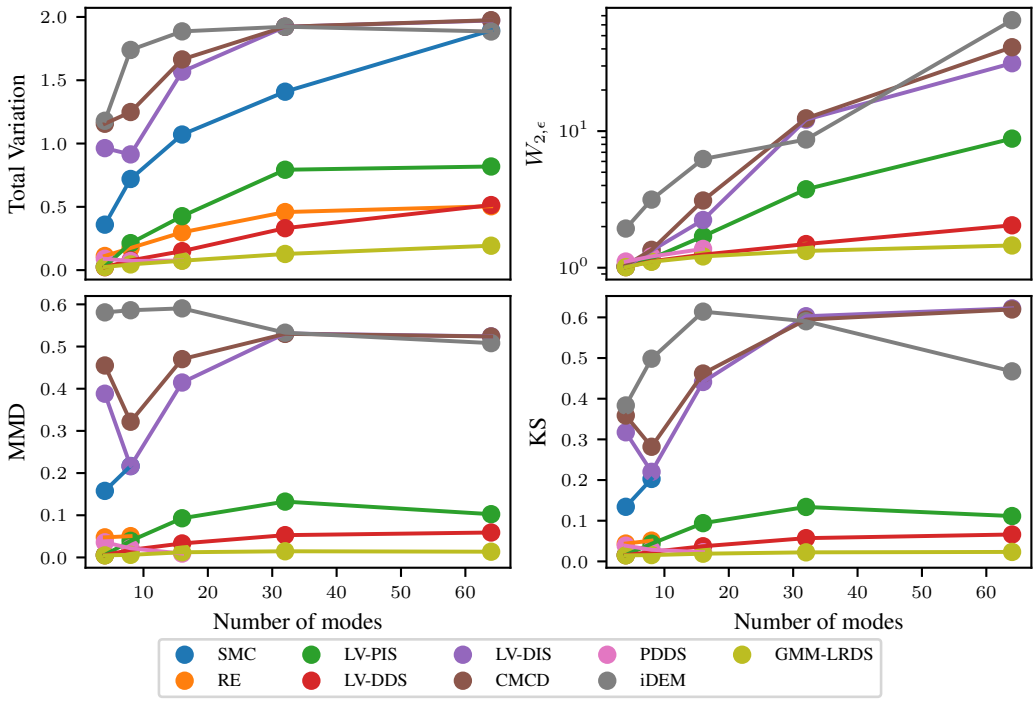


Figure D.7: **Results for multi-modal Gaussian mixtures**, averaged over 16 sampling runs. (**Top Left**): Total variation distance between weight histograms. (**Top Right**): Results with regularized Wasserstein distance. (**Bottom Left**): Results with MMD distance. (**Bottom Right**): Results with KS distance. Incomplete curves refer to settings with missing results due to numerical issues.

$a \in \{1, 2, 4, 8, 16\}$ indicates the complexity level of the target. For each target, we conduct 16 sampling runs of GMM-LRDS, and display the results in Figure D.6. Our numerics show that the performance GMM-LRDS remains consistent with increasing a, proving the interest of our method for complex multi-modal targets.

Limitation of LRDS: high dimension. Although LRDS outperforms its competitors when sampling from challenging bi-modal Gaussian mixtures, its performance tends to decrease when the dimension is large, like most of samplers. To highlight this limitation, we provide in Table D.8 the results of GMM-LRDS for various mixture settings with $d = 128$, where we observe that it fails to recover a good estimation of the strongest mode weight (roughly, 10% of relative error).

Multi-modal Gaussian mixtures

We display in Figure D.7 the results of all considered samplers, including LRDS, when targeting the multi-modal Gaussian mixtures defined in Appendix D.8.1 with fixed dimension $d = 8$ and increasing number of modes $L \in \{4, 8, 16, 32, 64\}$. To assess the sampling quality at a global level, we dispense for each sampler the total variation distance between the true weight histogram and the weight histogram obtained via Monte Carlo approximation. All metrics are computed based on 8192 samples. We notably observe that GMM-LRDS has the best performance compared to all other sampling methods, independently of the number of modes in the target distribution. In particular, Figure D.7 demonstrates that GMM-LRDS is able to recover both global features and local features for a complex multi-modal distribution.

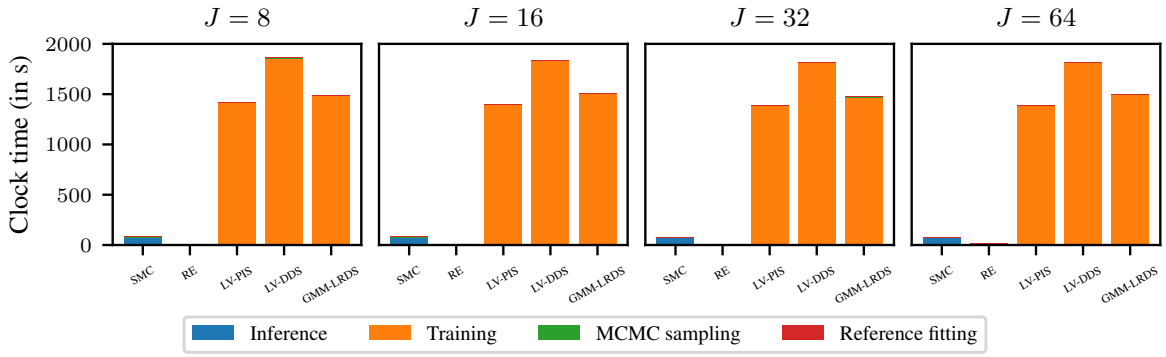


Figure D.8: **Execution clock-time for multi-modal Gaussian mixtures with increasing number of mixture components** (from left to right).

Execution time. To demonstrate the applicability of LRDS in practice, we provide in Figure D.8 the execution clock-time of each sampling method for all multi-modal Gaussian mixture sampling settings. The conclusions are the same as in the bi-modal setting: GMM-LRDS has equivalent training time to previous variational diffusion-based approaches, with negligeable extra cost due to reference fitting.

Limitation of LRDS: high number of modes. As depicted in Figure D.7 (top left), the recovery of the mode weights for a multi-modal target distribution gets worse as soon as the number of modes is large. Although GMM-LRDS performs better than its competitors, it suffers from this limitation.

Rings and Checkerboard distributions

In Figure D.9 and Figure D.10, we illustrate the impact of the number J of Gaussian mixture components when running GMM-LRDS to sample from Rings and Checkerboard distributions, detailed in Appendix D.8.1,by taking $J \in \{1, 8, 16, 32, 64\}$. In particular, we consider mixture models with diagonal covariance (first two rows) and full covariance (last two rows), fitted on MCMC samples from the target via EM algorithm. For both settings, we observe that setting J large enables to recover with more precision the support of the target distribution, leading consequently to better performance of GMM-LRDS. In the special Rings setting, where the target energy landscape is narrow and complex, we also notice that mixtures with full covariance provide better estimation of the support of the distribution. On the other hand, we remark that GMM-LRDS struggles to recover the geometry of the Checkerboard distribution, while EBM-LRDS is more performant, see Figure 5.6.

ϕ^4 field system

Table D.9 shows that all competing algorithms fail at recovering the weight ratio of the ϕ^4 system as they are highly prone to mode collapse either on the 'negative' mode or on the 'positive' mode.

Bayesian logistic regression models

Finally, we display in Table D.10 the results of all considered samplers, including LRDS, when targeting the Bayesian posterior distributions obtained from real-world data and defined in Appendix D.8.1. To assess the sampling quality, we compute the average predictive log-likelihood (*i.e.*the expected posterior distribution on the test dataset) via Monte Carlo approximation with 8192 samples. Since these distributions do not exhibit explicit multi-modal characteristics, we observe that standard MCMC-based techniques such as SMC and RE perform better than deep learning-based approaches. Interestingly, LRDS is on par or superior to previous diffusion-based approaches. For completeness, we also provide in Figure D.11 the execution clock-time of all samplers in this setting.

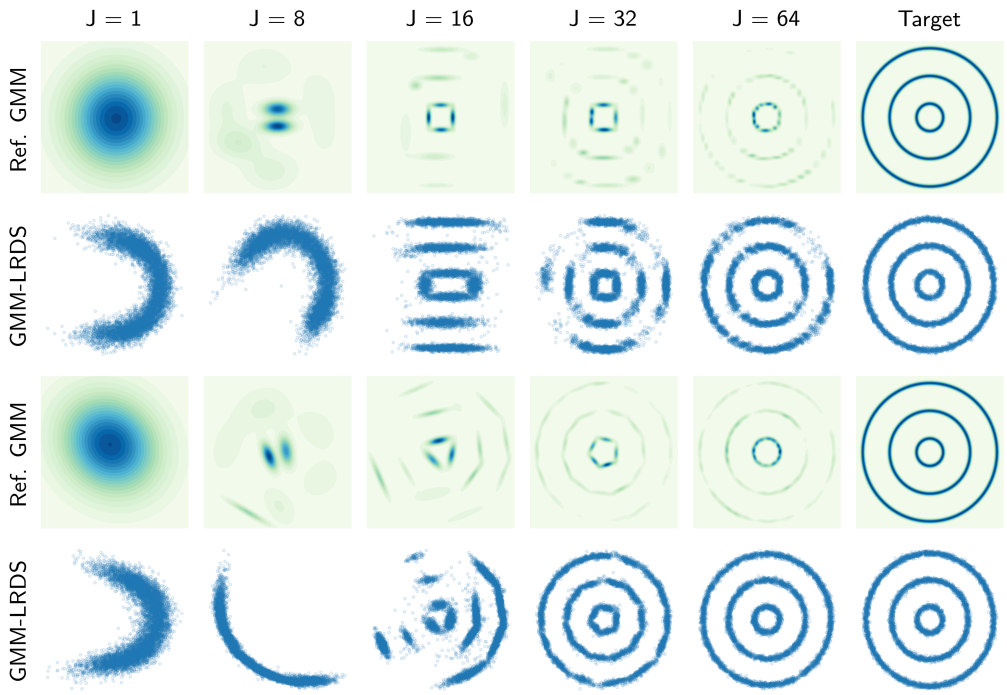


Figure D.9: **Results of GMM-LRDS for Rings distribution with increasingly expressive GMM for reference distribution. (Top two rows)** GMM density and GMM-LRDS samples for **diagonal** covariance parameterization **(Bottom two rows)** GMM density and GMM-LRDS samples for **full** covariance parameterization.

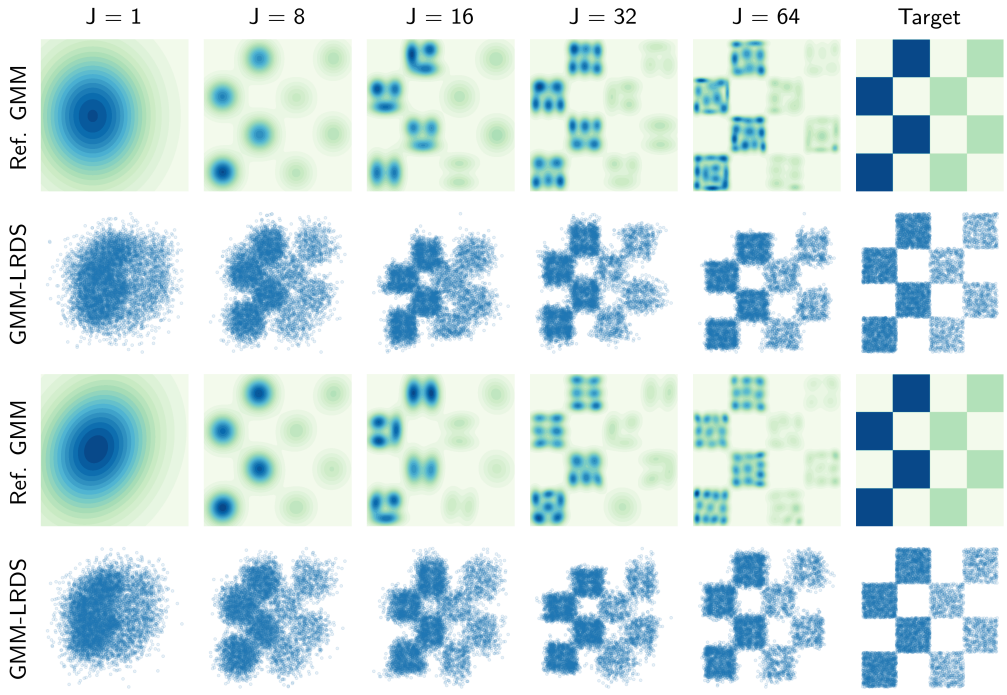


Figure D.10: **Results of GMM-LRDS for Checkerboard distribution with increasingly expressive GMM for reference distribution. (Top two rows)** GMM density and GMM-LRDS samples for **diagonal** covariance parameterization **(Bottom two rows)** GMM density and GMM-LRDS samples for **full** covariance parameterization.

Table D.9: **Estimation of the weight ratio for the ϕ^4 system** with different values of h, averaged over 16 runs. The Laplace approximations, that stand for ground truth, are displayed at the bottom of the table. Orange cells refer to settings with uninformative mode weight estimation (*i.e.*, uniform mixture), red cells denote mode collapse. N/A denotes settings where results could not be obtained due to numerical issues.

Algorithm	$h = 0$	$h = 9 \times 10^{-4}$	$h = 2 \times 10^{-3}$	$h = 2.5 \times 10^{-3}$	$h = 3.5 \times 10^{-3}$
SMC	$1.28_{\pm 1.19}$	$1.71_{\pm 1.47}$	$1.51_{\pm 1.38}$	$1.60_{\pm 1.52}$	$1.57_{\pm 1.43}$
RE	$1.00_{\pm 0.12}$	$1.01_{\pm 0.12}$	$1.04_{\pm 0.13}$	$1.11_{\pm 0.13}$	$1.07_{\pm 0.14}$
LV-PIS	∞	∞	∞	∞	∞
LV-DDS	$0.00_{\pm 0.00}$	∞	$0.00_{\pm 0.00}$	∞	∞
LV-DIS	$1.07_{\pm 0.03}$	$1.05_{\pm 0.03}$	$1.05_{\pm 0.06}$	$1.16_{\pm 0.12}$	$1.17_{\pm 0.07}$
CMCD	$1.01_{\pm 0.05}$	$0.98_{\pm 0.07}$	$1.01_{\pm 0.07}$	$0.96_{\pm 0.07}$	$0.96_{\pm 0.02}$
iDEM	$0.59_{\pm 0.01}$	$35.9_{\pm 1.43}$	$98.6_{\pm 10.6}$	$1.81_{\pm 0.03}$	$170_{\pm 17.8}$
PDDS	N/A	N/A	N/A	N/A	N/A
Laplace 0^{th} order	1.00	1.35	1.95	2.30	3.22
Laplace 2^{nd} order	1.00	1.34	1.90	2.23	3.08

Table D.10: **Average predictive log-likelihood** for Bayesian posterior distributions obtained from logistic regression tasks, averaged over 16 runs. Bold font indicates best performance.

Algorithm	**Ionosphere** ($d = 36$) ↑	**Sonar** ($d = 62$) ↑	**German Credit** ($d = 26$) ↑	**Breast Cancer** ($d = 32$) ↑
SMC	$-139.4_{\pm 0.3}$	$-191.4_{\pm 0.2}$	$\mathbf{-122.2_{\pm 0.1}}$	$\mathbf{-89.2_{\pm 0.5}}$
RE	$-139.5_{\pm 0.2}$	$-191.3_{\pm 0.1}$	$\mathbf{-122.2_{\pm 0.1}}$	$-90.1_{\pm 0.3}$
LV-PIS	$-156.0_{\pm 1.0}$	$-202.2_{\pm 0.2}$	$-153.7_{\pm 2.2}$	$-203.9_{\pm 3.3}$
LV-DDS	$-147.7_{\pm 0.5}$	$-195.2_{\pm 0.7}$	$-151.2_{\pm 1.3}$	$-128.4_{\pm 0.9}$
LV-DIS	$-224.3_{\pm 22.6}$	$-318.7_{\pm 47.0}$	$-206.5_{\pm 1.0}$	$-152.1_{\pm 1.9}$
CMCD	$-970.6_{\pm 2.7}$	$-1645.4_{\pm 6.2}$	$-171.9_{\pm 0.0}$	$-175.3_{\pm 5.7}$
iDEM	$-153.7_{\pm 0.2}$	$-4634.1_{\pm 12.9}$	$-132.4_{\pm 0.1}$	$-320.8_{\pm 0.9}$
PDDS	$-139.6_{\pm 0.3}$	$-191.3_{\pm 0.1}$	$\mathbf{-122.2_{\pm 0.1}}$	$-102.2_{\pm 0.1}$
GMM-LRDS	$\mathbf{-138.0_{\pm 0.5}}$	$\mathbf{-190.5_{\pm 1.0}}$	$-129.0_{\pm 2.7}$	$-100.1_{\pm 2.1}$
EBM-LRDS	$-148.0_{\pm 15.4}$	$-191.8_{\pm 0.9}$	$-131.2_{\pm 5.2}$	$-92.7_{\pm 4.0}$

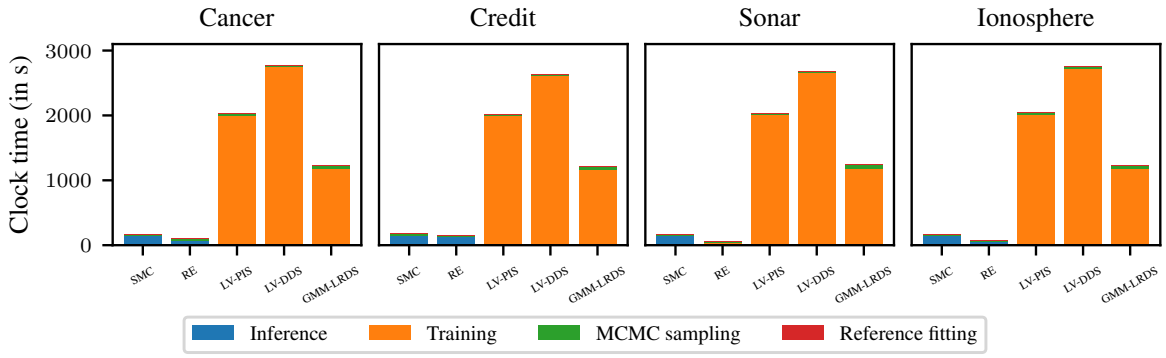


Figure D.11: Execution clock-time for Bayesian posterior distributions obtained from logistic regression tasks 'Ionosphere', 'Sonar', 'German Credit' and 'Breast Cancer' (from left to right).

Appendix E

Appendix of Chapter 6

Organization of the supplementary

The appendix is organized as follows. Appendix E.1 summarizes general facts that will be useful for proofs and corresponding computations. In Appendix E.2, we describe the general framework of noising diffusion processes, as well as the particular case of Variance-Preserving (consistently used in our experiments) and Variance-Exploding schemes: we notably detail there the formulas related to SDE/ODE integrators and to the computation of the log-determinant terms arising from the use of deterministic transitions in DM-aMC samplers (see Section 6.4). In Appendix E.3, we provide the proofs of all theoretical results dispensed in Section 6.4. Finally, we precisely detail our experimental setting in Appendix E.4, along with additional numerical results.

E.1 Preliminaries

E.1.1 Useful lemmas

Lemma 90 (Power series expansion of the matrix logarithm)**.** *Let* $(\alpha, \beta) \in \mathbb{R}^2$*. For any matrix* $\mathrm{M} \in \mathbb{R}^{d\times d}$ *satisfying* $\|\mathrm{M}\| < \min(1/|\alpha|, 1/|\beta|)$*, the following identities hold*

$$\log\left[(\mathrm{I}_d - \beta\mathrm{M})^{-1}(\mathrm{I}_d + \alpha\mathrm{M})\right] = \sum_{i=1}^{\infty} \frac{\beta^i - (-1)^i\alpha^i}{i}\mathrm{M}^i, \quad \log\left[(\mathrm{I}_d + \beta\mathrm{M})^{-1}(\mathrm{I}_d - \alpha\mathrm{M})\right] = \sum_{i=1}^{\infty} \frac{(-1)^i\beta^i - \alpha^i}{i}\mathrm{M}^i,$$

where log *denotes the matrix logarithm.*

Proof. This is an immediate corollary from [Hall, 2000, Theorem 3.6]. □

Corollary 91. *Let* $(c_1, c_2, c_3) \in \mathbb{R}^3$ *with* $c_1 \neq 0$*. Let* $\mathrm{M} \in \mathbb{R}^{d\times d}$ *be a matrix satisfying* $\|\mathrm{M}\| < \min(1/|c_2|, 1/|c_3|)$*. Define the matrices*

$$\mathrm{M}_1 = c_1(\mathrm{I}_d + c_2\mathrm{M})^{-1}(\mathrm{I}_d - c_3\mathrm{M}) \ , \ \mathrm{M}_2 = c_1^{-1}(\mathrm{I}_d - c_3\mathrm{M})^{-1}(\mathrm{I}_d + c_2\mathrm{M}) \ .$$

Then we have

$$\log|\det \mathrm{M}_1| = d\log|c_1| + \sum_{i=1}^{\infty} \frac{(-1)^i c_2^i - c_3^i}{i}\mathrm{Tr}[\mathrm{M}^i] \ , \ \log|\det \mathrm{M}_2| = -d\log|c_1| + \sum_{i=1}^{\infty} \frac{c_3^i - (-1)^i c_2^i}{i}\mathrm{Tr}[\mathrm{M}^i] \ .$$

Proof. Consider such (c_1, c_2, c_3) and such matrix M. Note that the assumption on c_2 and c_3 guarantees the

invertibility of $\mathrm{I}_d + c_2\mathrm{M}$ and $\mathrm{I}_d - c_3\mathrm{M}$. Regarding M_1, we have

$$\begin{aligned}
\log|\det M_1| &= d\log|c_1| + \log\left|\det\left((\mathrm{I}_d + c_2\mathrm{M})^{-1}(\mathrm{I}_d - c_3\mathrm{M})\right)\right| \\
&= d\log|c_1| + \log\left|\det(\mathrm{I}_d + c_2\mathrm{M})^{-1}\det(\mathrm{I}_d - c_3\mathrm{M})\right| \\
&= d\log|c_1| - \log\left|\det(\mathrm{I}_d + c_2\mathrm{M})\right| + \log\left|\det(\mathrm{I}_d - c_3\mathrm{M})\right| \\
&= d\log|c_1| - \log\det(\mathrm{I}_d + c_2\mathrm{M}) + \log\det(\mathrm{I}_d - c_3\mathrm{M}) && \text{[Behrmann et al., 2019, Lemma 6]} \\
&= d\log|c_1| + \log\det\left((\mathrm{I}_d + c_2\mathrm{M})^{-1}(\mathrm{I}_d - c_3\mathrm{M})\right) \\
&= d\log|c_1| + \mathrm{Tr}\log\left((\mathrm{I}_d + c_2\mathrm{M})^{-1}(\mathrm{I}_d - c_3\mathrm{M})\right) . && \text{[Hall, 2000, Theorem 3.10]}
\end{aligned}$$

Hence, we obtain the first result of Corollary 91 by using the second statement of Lemma 90 with $\beta = c_2$ and $\alpha = c_3$. Similar computations with M_2 lead to the second result. □

E.1.2 Discrete time setting for diffusion models

Following Karras et al. [2024], Grenioux et al. [2024], we define the time discretization $\{t_k\}_{k=0}^K \subset [0,T]$ accordingly to the growth (in log-scale) of the noise level $t \mapsto \sigma(t)$.

Given fixed boundary values $\sigma_{\min} > 0$ and $\sigma_{\max} > 0$, we define for any $k \in \{0,\dots,K\}$ the timestep t_k by

$$\log\sigma(t_k)^2 = \log\sigma_{\min}^2 + \frac{k}{K}\log\left(\frac{\sigma_{\max}^2}{\sigma_{\min}^2}\right) \iff \sigma(t_k)^2 = \sigma_{\min}^2\left(\frac{\sigma_{\max}^2}{\sigma_{\min}^2}\right)^{k/K} \tag{E.1}$$

$$\iff t_k = \sigma^{-1}\left(\sigma_{\min}\left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)^{k/K}\right) . \tag{E.2}$$

Depending on the type of noising, the map σ^{-1} may be explicit or not; in the latter case, we approximate it via a dichotomy scheme. In practice, $\sigma_{\max} = \sigma(T)$ and $\sigma_{\min}$ is chosen very close to 0 such that $t_K \approx T$ and $t_0 \approx 0$; hence, we have $p_{t_K} \approx \pi^{\text{base}}$ and $p_{t_0} \approx \pi$. We denote $\delta_k = t_{k+1} - t_k$.

E.1.3 General results on SDE/ODE integration

A12 (Integrability conditions on f and g). *Coefficients $f : [0,T] \to \mathbb{R}$ and $g : [0,T] \to (0,\infty)$ are such that (a) f is integrable on $(0,T)$ and (b) g is integrable on $(0,T)$.*

Lemma 92 (SDE exponential integration). *Let $T > 0$ and $b \in \mathbb{R}^d$. Consider the SDE defined on $[0,T]$ by $\mathrm{d}Y_t = f(t)(Y_t + b)\mathrm{d}t + g(t)\mathrm{d}B_t$, where coefficients f and g verify **A**12. Then, for any pair of time-steps (s,t) such that $T \geq t > s \geq 0$, the conditional distribution of Y_t given $Y_s = y_s \in \mathbb{R}^d$, denoted by $q_{t|s}(\cdot|y_s)$, verifies*

$$q_{t|s}(\cdot|y_s) = \mathrm{N}\left(\exp(\textstyle\int_s^t f(u)\mathrm{d}u)y_s + \left(\exp(\int_s^t f(u)\mathrm{d}u) - 1\right)b, \int_s^t g^2(u)\exp(2\int_u^t f(r)\mathrm{d}r)\mathrm{d}u\,\mathrm{I}_d\right) .$$

Proof. Assume **A**12. Define the function $\zeta : t \in [0,T] \to \exp(-\int_0^t f(u)\mathrm{d}u)$ and consider the stochastic process $(Z_t)_{t\in[0,T]}$ defined by $Z_t = \zeta(t)Y_t$ for any $t \in [0,T]$. By Îto's formula, we have $\mathrm{d}Z_t = f(t)\zeta(t)b\mathrm{d}t + \zeta(t)g(t)\mathrm{d}B_t = -\dot{\zeta}(t)b\mathrm{d}t + \zeta(t)g(t)\mathrm{d}B_t$. Therefore, for any time-steps (s,t) such that $T \geq t > s \geq 0$, we have

$$\zeta(t)Y_t - \zeta(s)Y_s = \{\zeta(s) - \zeta(t)\}b + \textstyle\int_s^t \zeta(u)g(u)\mathrm{d}B_u ,$$

and then

$$Y_t = \exp(\textstyle\int_s^t f(u)\mathrm{d}u)Y_s + \left(\exp(\int_s^t f(u)\mathrm{d}u) - 1\right)b + \int_s^t g(u)\exp(\int_u^t f(r)\mathrm{d}r)\mathrm{d}B_u ,$$

which gives the result using Îto's isometry and that Y_s is independent from $(B_t - B_s)_{t\in[s,T]}$. □

The following lemma can be seen as the limit of Lemma 92 in the deterministic regime, *i.e.*, when $g(t) = 0$ for any $t \in [0,T]$.

Lemma 93 (ODE exponential integration). *Let $T > 0$ and $b \in \mathbb{R}^d$. Consider the ODE defined on $[0,T]$ by $\mathrm{d}Y_t = f(t)[Y_t + b]\mathrm{d}t$, where coefficient f verifies **A**12. Then, for any pair of time-steps (s,t) such that $T \geq t > s \geq 0$, the ODE solution Y_t given $Y_s = y_s \in \mathbb{R}^d$ is defined by*

$$Y_t = \exp\left(\textstyle\int_s^t f(u)\mathrm{d}u\right)y_s + \left(\exp\left(\int_s^t f(u)\mathrm{d}u\right) - 1\right)b .$$

Proof. Let $0 \leq s < t \leq T$, set $Z_t = \exp(-F(t))Y_t$, where $F(t) = \int_0^t f(u)\mathrm{d}u$, then

$$\mathrm{d}Z_t = f(t)\exp(-F(t))b\mathrm{d}t \ ,$$

which implies that

$$Z_t = Z + (\exp(-F(s)) - \exp(-F(t)))\, b \ ,$$

which gives the result. □

E.1.4 Review of score and energy matching methods

Consider the noising diffusion process given by the SDE (6.2). In this section, we review a selection of methods used to learn the scores, the time scores and/or the log-densities (*i.e.*, energies) of the marginal distributions $(p_t)_{t\in[0,T]}$ associated to this process, based on samples from the target distribution π with unnormalized density γ. While the presented score matching (and time score matching) techniques have widely been experimented within the diffusion model community, the evoked log-density estimation (also referred to as energy matching) approaches are much more recent, and only provide a small glimpse into the pretty young field of research to which they belong. In the following, we will denote $r(t) = S(t)\sigma(t)$, where coefficients S and σ are introduced in (6.3) to marginally characterize diffusion models. We adopt a consistent notation for neural networks: $\mathbf{U}_t^\theta$ is used to learn the log-density $\log p_t$, $\mathbf{s}_t^\theta$ to learn the score $\nabla \log p_t$, and $\mathbf{u}_t^\theta$ to learn the time score $\partial_t \log p_t$.

Denoising Score Matching (DSM) [Song et al., 2021b]. This is the standard score matching loss used in the diffusion-based generative modeling literature. It relies on the so-called Tweedie identity

$$\nabla \log p_t(x_t) = \mathrm{E}[\nabla \log q_{t|0}(x_t|X_0)] \ , \ X_0 \sim q_{0|t}(\cdot|x_t) \ ,$$

where $q_{t|0}$ is the *tractable* noising transition kernel between times 0 and t, see (6.7), and $q_{0|t}$ is the related denoising transition kernel, which verifies by Bayes property $q_{0|t}(x_0|x_t) \propto \gamma(x_0)q_{t|0}(x_t|x_0)$. This gives rise to the following objective for estimating the score function $(t,x) \mapsto \nabla \log p_t(x)$ by a neural network $(t,x) \mapsto \mathbf{s}_t^\theta(x)$

$$\mathcal{L}_{\mathrm{DSM}}(\theta) = \mathrm{E}\left[\left\|\mathbf{s}_t^\theta(X_t) + \frac{Z}{r(t)}\right\|_2^2\right] \ , \ X_t = S(t)X_0 + r(t)Z \ ,$$

where $t \sim \mathrm{U}(0,T)$, $X_0 \sim \pi$ and $Z \sim \mathrm{N}(0, \mathrm{I}_d)$. In practice, one rather uses a reweighted version of this objective given by

$$\tilde{\mathcal{L}}_{\mathrm{DSM}}(\theta) = \mathrm{E}\left[r^2(t)\left\|\mathbf{s}_t^\theta(X_t) + \frac{Z}{r(t)}\right\|_2^2\right] = \mathrm{E}\left[\left\|r(t)\mathbf{s}_t^\theta(X_t) + Z\right\|_2^2\right] \ .$$

To further reduce its variance with respect to the noise variable, one may consider applying the antithetic trick on Z variable and thus obtain the loss function

$$\tilde{\mathcal{L}}_{\mathrm{DSM}}^{\mathrm{anti}}(\theta) = \mathrm{E}\left[\frac{1}{2}\left\|r(t)\mathbf{s}_t^\theta(X_t) + Z\right\|_2^2 + \frac{1}{2}\left\|r(t)\mathbf{s}_t^\theta(X_t^-) - Z\right\|_2^2\right] \ ,$$
$$\text{with } X_t = S(t)X_0 + r(t)Z \ , \ X_t^- = S(t)X_0 - r(t)Z.$$

In practice, the DSM objective described above still exhibits high variance. To ensure its robustness and stability for large-scale applications, an equivalent objective, coined EDM, was proposed by Karras et al. [2022], which specifically relies on preconditioning guidelines for the neural network $\mathbf{s}^\theta$. In our experiments, the "DSM objective" will systematically refer to this specific EDM training loss, enhanced with the antithetic trick, whose success has been widely proven over the last few years for generative tasks.

Target Score Matching (TSM) [Bortoli et al., 2024a]. Alternatively, by operating a change-of-variable in the Tweedie's formula, the following identity also holds

$$\nabla \log p_t(x_t) = \frac{1}{S(t)}\mathrm{E}[\nabla \log \gamma(X_0)] \ , \ X_0 \sim q_{0|t}(\cdot|x_t) \ .$$

This gives rise to the following objective for estimating the score function $(t,x) \mapsto \nabla \log p_t(x)$ by a neural network $(t,x) \mapsto \mathbf{s}_t^\theta(x)$

$$\mathcal{L}_{\mathrm{TSM}}(\theta) = \mathrm{E}\left[\left\| \mathbf{s}_t^\theta(X_t) - \frac{\nabla \log \gamma(X_0)}{S(t)} \right\|_2^2\right] , \; X_t = S(t)X_0 + r(t)Z ,$$

where $t \sim \mathrm{U}(0,T)$, $X_0 \sim \pi$ and $Z \sim \mathrm{N}(0, \mathrm{I}_d)$. In practice, one rather uses a reweighted version of this objective given by

$$\tilde{\mathcal{L}}_{\mathrm{TSM}}(\theta) = \mathrm{E}\left[S^2(t)\left\| \mathbf{s}_t^\theta(X_t) - \frac{\nabla \log \gamma(X_0)}{S(t)} \right\|_2^2\right] = \mathrm{E}\left[\left\| S(t)\mathbf{s}_t^\theta(X_t) - \nabla \log \gamma(X_0) \right\|_2^2\right] ,$$

which itself can be improved via the antithetic trick as

$$\tilde{\mathcal{L}}^{\mathrm{anti}}_{\mathrm{TSM}}(\theta) = \mathrm{E}\left[\frac{1}{2}\left\| S(t)\mathbf{s}_t^\theta(X_t) - \nabla \log \gamma(X_0) \right\|_2^2 + \frac{1}{2}\left\| S(t)\mathbf{s}_t^\theta(X_t^-) - \nabla \log \gamma(X_0) \right\|_2^2\right] ,$$
$$\text{with } X_t = S(t)X_0 + r(t)Z \; , \; X_t^- = S(t)X_0 - r(t)Z.$$

In our experiments, the "TSM objective" will systematically refer to the training loss function $\tilde{\mathcal{L}}^{\mathrm{anti}}_{\mathrm{TSM}}$. As originally proposed by Bortoli et al. [2024a], this loss can also be combined with preconditioning schemes to reduce its variance in practice; however, since those are not compatible with the preconditioning directives from Karras et al. [2022], we do not integrate them in our numerical experiments.

Time Score Matching (tSM)[Guth et al., 2025, Yu et al., 2025]. Interestingly, the time score function has a similar decomposition

$$\partial_t \log p_t(x_t) = \mathrm{E}[\partial_t \log q_{t|0}(x_t|X_0)] \; , \; X_0 \sim q_{0|t}(\cdot|x_t) \; .$$

Since the conditional time derivative $\partial_t \log q_{t|0}$ is as tractable as the conditional score $\nabla \log q_{t|0}$, this gives rise to the following objective for estimating the time score function $(t,x) \mapsto \partial_t \log p_t(x)$ by a neural network $(t,x) \mapsto \mathbf{u}_t^\theta(x)$

$$\mathcal{L}_{\mathrm{tSM}}(\theta) = \mathrm{E}\left[\left\| \mathbf{u}_t^\theta(X_t) - u_{\mathrm{tSM}}^{\mathrm{target}}(t, X_0, Z) \right\|_2^2\right] \; , \; X_t = S(t)X_0 + r(t)Z \; ,$$

where $u_{\mathrm{tSM}}^{\mathrm{target}}(t,x_0,z) = -[\dot{r}(t)/r(t)]\left(d - \|z\|^2\right) + [\dot{S}(t)/r(t)]x_0^\top z$, $t \sim \mathrm{U}(0,T)$, $X_0 \sim \pi$ and $Z \sim \mathrm{N}(0,\mathrm{I}_d)$. Similarly, this objective may include antithetic trick and rewrite as

$$\mathcal{L}^{\mathrm{anti}}_{\mathrm{tSM}}(\theta) = \mathrm{E}\left[\frac{1}{2}\left\| \mathbf{u}_t^\theta(X_t) - u_{\mathrm{tsM}}^{\mathrm{target}}(t, X_0, Z) \right\|_2^2 + \frac{1}{2}\left\| \mathbf{u}_t^\theta(X_t^-) - u_{\mathrm{tsM}}^{\mathrm{target}}(t, X_0, -Z) \right\|_2^2\right] ,$$
$$\text{with } X_t = S(t)X_0 + r(t)Z \; , \; X_t^- = S(t)X_0 - r(t)Z.$$

As such, it has been observed that the derived objective exhibits very high variance, even more than score matching methods. While Guth et al. [2025] explore a reweighting precisely adjusted to the VE noising scheme, Yu et al. [2025] propose an alternative reweighting in the context of the VP noising scheme; we implement the latter formulation with the antithetic trick, to which the "tSM objective" will systematically refer in our experiments. Note that by including a change-of-variable into the expression of the time score, we may obtain a target-like version of the tSM objective given by

$$\mathcal{L}_{\mathrm{tTSM}}(\theta) = \mathrm{E}\left[\left\| \mathbf{u}_t^\theta(X_t) - u_{\mathrm{tTSM}}^{\mathrm{target}}(t, X_0, Z) \right\|_2^2\right] \; , \; X_t = S(t)X_0 + r(t)Z \; ,$$

where $u_{\mathrm{tTSM}}^{\mathrm{target}}(t,x_0,z) = -\left\{[\dot{S}(t)/S(t)]x_0 + [\sigma(t)\dot{r}(t)/r(t)]z\right\}^\top \nabla \log \gamma(x_0)$, $t \sim \mathrm{U}(0,T)$, $X_0 \sim \pi$ and $Z \sim \mathrm{N}(0,\mathrm{I}_d)$, along with its antithetic-like version

$$\mathcal{L}^{\mathrm{anti}}_{\mathrm{tTSM}}(\theta) = \mathrm{E}\left[\tfrac{1}{2}\left\| \mathbf{u}_t^\theta(X_t) - u_{\mathrm{tTSM}}^{\mathrm{target}}(t, X_0, Z) \right\|_2^2 + \tfrac{1}{2}\left\| \mathbf{u}_t^\theta(X_t^-) - u_{\mathrm{tTSM}}^{\mathrm{target}}(t, X_0, -Z) \right\|_2^2\right] ,$$
$$\text{with } X_t = S(t)X_0 + r(t)Z \; , \; X_t^- = S(t)X_0 - r(t)Z.$$

While this objective is enriched with the information of the target score $\nabla \log \gamma$, we did not use this objective in our experiments due to its variance instability during training procedure.

Log-density Fokker-Planck-Equation (LFPE) [Lai et al., 2023, Shi et al., 2024, Sun et al., 2024]. A key property of the noising SDE (6.2) is that the induced log-densities $(p_t)_{t\in[0,1]}$ can be described by a partial differential equation called the *Fokker-Planck equation* [Øksendal, 2003], whose formulation can be written as

$$\partial_t \log p_t(x) = \mathcal{F}[\log p](t,x) \stackrel{\text{def}}{=} \frac{1}{2}g^2(t)\left[\operatorname{div}(\nabla \log p_t)(x) + \|\nabla \log p_t(x)\|_2^2\right] - f(t)\{d + x^\top \nabla \log p_t(x)\}\,,$$

where div denotes the divergence operator defined by $\operatorname{div} \mathrm{F} = \operatorname{Tr}[\nabla \mathrm{F}]$. This gives rise to the following objective for estimating the log-density $(t,x) \mapsto \log p_t(x)$ by a neural network $(t,x) \mapsto \mathbf{U}_t^\theta(x)$

$$\mathcal{L}_{\text{LFPE}}(\theta) = \mathrm{E}\left[\left\|\partial_t \mathbf{U}_t^\theta(X_t) - \operatorname{sg}\left\{\mathcal{F}[\mathbf{U}^\theta](t,X_t)\right\}\right\|_2^2\right]\,,\; X_t = S(t)X_0 + r(t)Z\,,$$

where sg denotes the stop-gradient[1] operator with respect to parameter θ, $t \sim \mathrm{U}(0,T)$, $X_0 \sim \pi$ and $Z \sim \mathrm{N}(0, \mathrm{I}_d)$. In this case too, we can derive an objective based on the antithetic trick

$$\mathcal{L}^{\text{anti}}_{\text{LFPE}}(\theta) = \mathrm{E}\left[\frac{1}{2}\left\|\partial_t \mathbf{U}_t^\theta(X_t) - \operatorname{sg}\left\{\mathcal{F}[\mathbf{U}^\theta](t,X_t)\right\}\right\|_2^2 + \frac{1}{2}\left\|\partial_t \mathbf{U}_t^\theta(X_t^-) - \operatorname{sg}\left\{\mathcal{F}[\mathbf{U}^\theta](t,X_t^-)\right\}\right\|_2^2\right]\,,$$
$$\text{with } X_t = S(t)X_0 + r(t)Z\,,\; X_t^- = S(t)X_0 - r(t)Z.$$

In our experiments, the "LFPE objective" will always refer to the training loss $\mathcal{L}^{\text{anti}}_{\text{LFPE}}$.

Approximate LFPE (aLFPE) [Plainer et al., 2025]. In the case where the target term $\mathcal{F}[\mathbf{U}^\theta]$ is not detached with respect to θ in $\mathcal{L}_{\text{LFPE}}$, the main numerical burden lies in the computation of the divergence term $\operatorname{div}(\nabla \mathbf{U}^\theta)$ when the dimension is large. To reduce this overhead, Plainer et al. [2025] propose to instead consider a first-order statistical estimation of the residual term $\mathcal{R}^\theta(t,x) = \mathcal{F}[\mathbf{U}^\theta](t,x) - \partial_t \mathbf{U}_t^\theta(x)$ given by $\tilde{\mathcal{R}}^\theta(t,x) = \mathrm{E}_v[\tilde{\mathcal{R}}^\theta(t,x;v)]$, with $v \sim \mathrm{N}(0,\sigma^2\mathrm{I}_d)$ for a small $\sigma > 0$, where[2]

$$\begin{aligned}\tilde{\mathcal{R}}^\theta(t,x;v) = {}& \frac{1}{2}g^2(t)\left[\left(\frac{v}{\sigma}\right)^\top \frac{\nabla \mathbf{U}_t^\theta(x+v) - \nabla \mathbf{U}_t^\theta(x-v)}{2\sigma}\right] \\ & + \frac{1}{2}\left[\frac{1}{2}g^2(t)\left\|\nabla \mathbf{U}_t^\theta(x+v)\right\|_2^2 - f(t)\{d + (x+v)^\top \nabla \mathbf{U}_t^\theta(x+v)\} - \partial_t \mathbf{U}_t^\theta(x+v)\right] \\ & + \frac{1}{2}\left[\frac{1}{2}g^2(t)\left\|\nabla \mathbf{U}_t^\theta(x-v)\right\|_2^2 - f(t)\{d + (x-v)^\top \nabla \mathbf{U}_t^\theta(x-v)\} - \partial_t \mathbf{U}_t^\theta(x-v)\right]\,.\end{aligned}$$

This gives rise to the following objective

$$\mathcal{L}_{\text{aLFPE}}(\theta) = \mathrm{E}\left[\left(\sum_{i=1}^N \tilde{\mathcal{R}}^\theta(t,X_t;v_i^{X_t})\right)\left(\sum_{j=1}^N \tilde{\mathcal{R}}^\theta(t,X_t;v_j^{X_t})\right)\right]\,,\; X_t = S(t)X_0 + r(t)Z\,,$$

where $t \sim \mathrm{U}(0,T)$, $X_0 \sim \pi$, $Z \sim \mathrm{N}(0,\mathrm{I}_d)$ and $\{v_i^{X_t}, v_j^{X_t}\}_{j=1}^N$ are $2N$ independent samples from $\mathrm{N}(0,\sigma^2\mathrm{I}_d)$ defined for each input X_t. Overall, this formulation avoids the need of the divergence computation (while maintaining backpropagation through the scores), at the cost of non-negligible statistical error. Following the guidelines from Plainer et al. [2025], we consistently set in our experiments $\sigma = 0.0001$, but choose $N = 64$ (instead of $N = 1$ as originally proposed) to reduce the variance of the loss, and bring it into the most favorable setting. We also choose to keep the use of auto-differentiation to compute the time derivative $\partial_t \mathbf{U}^\theta$ instead of using finite difference approximation as suggested by Plainer et al. [2025], as it brings more stability during training. In our experiments, we will systematically refer to this version of $\mathcal{L}_{\text{aLFPE}}$ as the "aLFPE objective".

[1] While cited related works did not consider detaching the term $\mathcal{F}[\mathbf{U}^\theta]$ with respect to θ in their respective formulation, we made this choice to avoid backpropagation through both first and second-order derivatives of $\mathbf{U}^\theta$, which was computationally infeasible in the high-dimensional settings considered in this paper. Nonetheless, we emphasize that, in our early experiments, we observed unchanged results on pure log-density estimation tasks for small dimensional settings, thereby suggesting that our methodology remains sound.

[2] Even though this objective features an additional term compared to the one stated in Equation (12) from Plainer et al. [2025], it is consistent with the related code available at https://github.com/noegroup/ScoreMD. This extra term actually originates from the use of the antithetic trick on the Gaussian variable v.

Radon-Nikodym Estimator (RNE) [He et al., 2026]. Alternatively, a discrete-time formulation of the LFPE objective has been proposed to learn the log-densities $(\log p_t)_{t\in[0,T]}$, based on the Bayes's rule (ideally satisfied by DMs) stating that for any times $(s,t)\in[0,T]^2$ and any inputs x_s and x_t, we have $p_t(x_t)q_{s|t}(x_s|x_t) = p_s(x_s)q_{t|s}(x_t|x_s)$, where $q_{s|t}$ and $q_{t|s}$ correspond to related stochastic transition kernels (see Section 6.2.1). Enforcing this consistency with log-densities can thus be translated into the following objective for estimating the log-density $(t,x)\mapsto\log p_t(x)$ by a neural network $(t,x)\mapsto\mathbf{U}_t^\theta(x)$

$$\mathcal{L}_{\mathrm{RNE}}(\theta) = \mathrm{E}\left[\left\|\mathbf{U}_t^\theta(X_t) - \mathbf{U}_s^\theta(X_s) - \mathrm{sg}\{\log q_{t|s}(X_t|X_s) - \log q_{s|t}^\theta(X_s|X_t)\}\right\|_2^2\right] ,$$
$$\text{with}\quad X_s = S(s)X_0 + r(s)Z\ ,\ X_t = [S(t)/S(s)]X_s + r(t)\{1-\sigma^2(s)/\sigma^2(t)\}^{1/2}\tilde{Z}\ ,$$

where sg denotes the stop-gradient operator with respect to parameter θ, $(s,t)\sim\mathrm{U}(\{(t_k,t_{k+1}) : k\in[0,K-1]\})$ with $\{t_k\}_{k=0}^{K-1}$ being a discretization of time interval $[0,T]$, $X_0\sim\pi$ and $(Z,\tilde{Z})\sim\mathrm{N}(0,\mathrm{I}_d)\otimes\mathrm{N}(0,\mathrm{I}_d)$. While $q_{t|s}$ denotes a *noising* transition kernel[3], that is tractable by (6.7), $q_{s|t}^\theta$ is an approximate *denoising* transition kernel, that may be computed via the learned score $\nabla\mathbf{U}^\theta$, see for instance (6.8). In this case too, one may consider the variant featuring the antithetic trick

$$\begin{aligned}\mathcal{L}_{\mathrm{RNE}}^{\mathrm{anti}}(\theta) = \mathrm{E}\Bigg[&\frac{1}{2}\left\|\mathbf{U}_t^\theta(X_t) - \mathbf{U}_s^\theta(X_s) - \mathrm{sg}\{\log q_{t|s}(X_t|X_s) - \log q_{s|t}^\theta(X_s|X_t)\}\right\|_2^2\\ &+\frac{1}{2}\left\|\mathbf{U}_t^\theta(X_t^-) - \mathbf{U}_s^\theta(X_s^-) - \mathrm{sg}\{\log q_{t|s}(X_t^-|X_s^-) - \log q_{s|t}^\theta(X_s^-|X_t^-)\}\right\|_2^2\Bigg] ,\\ \text{with}\quad & X_s = S(s)X_0 + r(s)Z\ ,\ X_t = [S(t)/S(s)]X_s + r(t)\{1-\sigma^2(s)/\sigma^2(t)\}^{1/2}\tilde{Z}\ ,\\ \text{and}\quad & X_s^- = S(s)X_0 - r(s)Z\ ,\ X_t^- = [S(t)/S(s)]X_s^- + r(t)\{1-\sigma^2(s)/\sigma^2(t)\}^{1/2}\tilde{Z}\ .\end{aligned}$$

Contrary to the LFPE objective, the obtained loss function does not require backpropagation through the time derivative $\partial_t\mathbf{U}^\theta$, which represents a significant computational advantage. However, $\mathcal{L}_{\mathrm{RNE}}$ suffers from a severe bias-variance tradeoff with respect to the time gap $\delta = t-s$ for selected times s and t: if δ is too large, then the denoising approximation obtained via $q_{s|t}^\theta$ may be ineffective and bring much bias; on the other hand, if δ is too small, the resulting objective may be prone to high variance. While He et al. [2026] propose to use the Euler-Maruyama estimation for $q_{s|t}^\theta$, see (6.8), we rather consider the Exponential Integration, see Appendix E.2.2 and Appendix E.2.3 for the formulas, which provides better accuracy for larger gap δ. In our experiments, we will systematically refer to this version of $\mathcal{L}_{\mathrm{RNE}}^{\mathrm{anti}}$ as the "RNE objective".

E.2 Details on (de)noising diffusion processes

In this section, we consider a target probability distribution $\pi\in\mathscr{P}(\mathbb{R}^d)$ and a pair of time points (s,t) satisfying $T\geq t>s\geq 0$. We present technical derivations related to the integration of (de)noising diffusion processes under a unified framework, covering the generic setting (Appendix E.2.1), the Variance-Preserving scheme (Appendix E.2.2), and the Variance-Exploding scheme (Appendix E.2.3). Throughout, the notation $\mathbf{s}_t(x)$ and $\mathbf{H}_t(x)$ denotes, respectively, exact or approximate evaluation of the score $\nabla\log p_t(x)$ and the Hessian $\nabla^2\log p_t(x)$. This unified formulation allows our computations to encompass both idealized and practical regimes considered in this paper.

E.2.1 General noising scheme

Here, we consider the most general form of SDE (6.2), where f and g both verify **A**12.

Lemma 94 (Exact noising SDE integration - General case)**.** *The conditional distribution of X_t given $X_s = x_s\in\mathbb{R}^d$ is defined by the Gaussian kernel*

$$q_{t|s}(\cdot|x_s) = \mathrm{N}\left(\{S(t)/S(s)\}x_s, S(t)^2\{\sigma^2(t)-\sigma^2(s)\}\,\mathrm{I}_d\right)\ ,$$

where $S(t) = \exp(\int_0^t f(u)\mathrm{d}u)$ and $\sigma^2(t) = \int_0^t g^2(u)/S(u)^2\mathrm{d}u$.

Proof. This is an immediate corollary of Lemma 92. □

[3] While He et al. [2026] propose to replace $q_{t|s}$, though tractable, by its Euler-Maruyama estimation, our implementation relies rather on its exact formulation to avoid bringing additional approximation error into the loss.

Lemma 95 (Approximate denoising SDE integration - General case)**.** *Denote $\delta = t - s$. Then, the conditional distribution of X_t given $X_s = x_s \in \mathbb{R}^d$ may be approximated by the Gaussian kernel*

$$q_{s|t}(\cdot|x_t) = \mathrm{N}\left((1 - f(t)\delta)x_t + g^2(t)\delta\, \mathbf{s}_t\left(x_t\right), g^2(t)\delta\, \mathrm{I}_d\right) \ ,$$

Proof. This result is a straightforward application of the Euler-Maruyama scheme applied to SDE (6.5). □

Lemma 96 (Approximate noising ODE integration - General case)**.** *Denote $\delta = t - s$. Then, the solution at time t of the forward probability flow ODE (6.6) starting from $x_s \in \mathbb{R}^d$ at time s may be approximated in two ways:*

$$\tilde{\mathrm{T}}_{t|s}(x_s) = x_s + \delta v(s, x_s) \qquad \textit{(Euler method: explicit, 1st order)}$$
$$\mathrm{T}_{t|s}(x_s) = x_s + \delta v\left(\frac{s+t}{2}, \frac{x_s + \mathrm{T}_{t|s}(x_s)}{2}\right) \qquad \textit{(Midpoint method : implicit, 2nd order)}$$
$$\textit{where } v(u, x) = f(u)x - \frac{g^2(u)}{2}\mathbf{s}_u\left(x\right) \ .$$

Lemma 97 (Approximate denoising ODE integration - General case)**.** *Denote $\delta = t - s$. Then, the solution at time s of the backward probability flow ODE (6.6) starting from $x_t \in \mathbb{R}^d$ at time t may be approximated in two ways:*

$$\tilde{\mathrm{T}}_{s|t}(x_t) = x_t - \delta v(t, x_t) \qquad \textit{(Euler method : explicit, 1st order)}$$
$$\mathrm{T}_{s|t}(x_t) = x_t - \delta v\left(\frac{s+t}{2}, \frac{\mathrm{T}_{s|t}(x_t) + x_t}{2}\right) \qquad \textit{(Midpoint method: implicit, 2nd order)}$$
$$\textit{where } v(u, x) = f(u)x - \frac{g^2(u)}{2}\mathbf{s}_u\left(x\right) \ .$$

Remark on the mutual invertibility of the ODE integrators. It is easy to verify that the noising and denoising implicit Midpoint integrators described above are mutual inversible maps, *i.e.*, we have $\mathrm{T}_{s|t} \circ \mathrm{T}_{t|s} = \mathrm{T}_{t|s} \circ \mathrm{T}_{s|t} = \mathrm{Id}$. However, this is not the case for the Euler maps $\tilde{\mathrm{T}}_{s|t}$ and $\tilde{\mathrm{T}}_{t|s}$.

Lemma 98 (Formula for the Jacobian of the Midpoint integrators)**.** *Let $\delta > 0$, and let define the numerical constants $c_1(\delta)$, $c_2(\delta)$ and $c_3(\delta)$ as*

$$c_1(\delta) = \frac{1 + (\delta/2)f((s+t)/2)}{1 - (\delta/2)f((s+t)/2)} \ , \ c_2(\delta) = \frac{\delta}{4}\frac{g^2\left(\frac{s+t}{2}\right)}{1 - \frac{\delta}{2}f\left(\frac{s+t}{2}\right)} \ , \ c_3(\delta) = \frac{\delta}{4}\frac{g^2\left(\frac{s+t}{2}\right)}{1 + \frac{\delta}{2}f\left(\frac{s+t}{2}\right)} \ .$$

Consider the same notation as in Lemma 96 and Lemma 97. Assume that there exists $L > 0$ such that $\mathbf{s}_{(s+t)/2}$ is L-Lipschitz. Then for any positive step-size $\delta = t - s$ such that $\max\left(|c_2(\delta)|, |c_3(\delta)|\right) < 1/L$, the Jacobians of Midpoint integration maps $\mathrm{T}_{t|s}$ and $\mathrm{T}_{s|t}$, respectively denoted by $J_{t|s}$ and $J_{s|t}$, verify for any inputs $x_s \in \mathbb{R}^d$ and $x_t \in \mathbb{R}^d$

$$J_{t|s}(x_s) = c_1(\delta)\left(\mathrm{I}_d + c_2(\delta)A(x_s)\right)^{-1}\left(\mathrm{I}_d - c_3(\delta)A(x_s)\right) \ ,$$
$$J_{s|t}(x_t) = c_1(\delta)^{-1}\left(\mathrm{I}_d - c_3(\delta)B(x_t)\right)^{-1}\left(\mathrm{I}_d + c_2(\delta)B(x_t)\right) \ ,$$

where $A(x_s) = \mathbf{H}_{(s+t)/2}\left(\frac{x_s + \mathrm{T}_{t|s}(x_s)}{2}\right)$ and $B(x_t) = \mathbf{H}_{(s+t)/2}\left(\frac{x_t + \mathrm{T}_{s|t}(x_t)}{2}\right)$.

Proof. The result from Lemma 98 follows from the factorization of the following identities, inherited from the implicit expressions of $\mathrm{T}_{t|s}$ and $\mathrm{T}_{s|t}$,

$$J_{t|s}(x_s) = \left(\left(1 - \frac{\delta}{2}f\left(\frac{s+t}{2}\right)\right)\mathrm{I}_d + \frac{\delta}{4}g^2\left(\frac{s+t}{2}\right)A(x_s)\right)^{-1}\left(\left(1 + \frac{\delta}{2}f\left(\frac{s+t}{2}\right)\right)\mathrm{I}_d - \frac{\delta}{4}g^2\left(\frac{s+t}{2}\right)A(x_s)\right) \ ,$$
$$J_{s|t}(x_t) = \left(\left(1 + \frac{\delta}{2}f\left(\frac{s+t}{2}\right)\right)\mathrm{I}_d - \frac{\delta}{4}g^2\left(\frac{s+t}{2}\right)B(x_t)\right)^{-1}\left(\left(1 - \frac{\delta}{2}f\left(\frac{s+t}{2}\right)\right)\mathrm{I}_d + \frac{\delta}{4}g^2\left(\frac{s+t}{2}\right)B(x_t)\right) \ .$$

Here, the assumption on δ guarantees the invertibility of the matrices $\mathrm{I}_d + c_2(\delta)A(x_s)$ and $\mathrm{I}_d - c_3(\delta)B(x_t)$. □

Proposition 99 (Exact expression of the Jacobian log-determinants of the Midpoint integrators via power series)**.** *Consider the same notation as in Lemma 98. Assume that there exists $L > 0$ such that $\mathbf{s}_{(s+t)/2}$ is L-Lipschitz. Then, for any positive step-size $\delta = t - s$ such that $\max(|c_2(\delta)|, |c_3(\delta)|) < 1/L$, for any inputs $x_s \in \mathbb{R}^d$ and $x_t \in \mathbb{R}^d$, we have*

$$\log\left|\det J_{t|s}(x_s)\right| = \sum_{i=0}^{\infty} a_i(s,t)\,\mathrm{Tr}([A(x_s)]^i)\ ,$$
$$\log\left|\det J_{s|t}(x_t)\right| = \sum_{i=0}^{\infty} b_i(s,t)\,\mathrm{Tr}([B(x_t)]^i)\ ,$$

where $\{a_i(s,t), b_i(s,t)\}_{i=0}^{\infty}$ are numerical coefficients defined by

$$a_0(s,t) = -b_0(s,t) = d\log\left[\left|\frac{1+(\delta/2)f((s+t)/2)}{1-(\delta/2)f((s+t)/2)}\right|\right]\ ,$$
$$a_i(s,t) = \frac{\delta^i}{4^i}g^{2i}\left(\frac{s+t}{2}\right)\frac{(-1)^i\left(1+\frac{\delta}{2}f\left(\frac{s+t}{2}\right)\right)^i - \left(1-\frac{\delta}{2}f\left(\frac{s+t}{2}\right)\right)^i}{i\left(1-\frac{\delta^2}{4}f^2\left(\frac{s+t}{2}\right)\right)^i} \text{ for any } i \geq 1\ ,$$
$$b_i(s,t) = \frac{\delta^i}{4^i}g^{2i}\left(\frac{s+t}{2}\right)\frac{\left(1-\frac{\delta}{2}f\left(\frac{s+t}{2}\right)\right)^i - (-1)^i\left(1+\frac{\delta}{2}f\left(\frac{s+t}{2}\right)\right)^i}{i\left(1-\frac{\delta^2}{4}f^2\left(\frac{s+t}{2}\right)\right)^i} \text{ for any } i \geq 1\ .$$

Proof. Consider the Jacobian matrices $J_{t|s}(x_s)$ and $J_{s|t}(x_t)$ introduced in Lemma 98. Note that we have $\|A(x_s)\| < \min(1/|c_2(\delta)|, 1/|c_3(\delta)|)$ and $\|B(x_t)\| < \min(1/|c_2(\delta)|, 1/|c_3(\delta)|)$ based on the assumptions on $\mathbf{s}_{(s+t)/2}$ and δ. This allows us to apply Corollary 91 on $J_{t|s}(x_s)$ and $J_{s|t}(x_t)$, respectively with $\mathrm{M} = A(x_s)$ and $\mathrm{M} = B(x_t)$, to obtain their expansion series in a straightforward manner. □

Remark on the δ-assumption in Lemma 98 and Proposition 99. For any general noise schedule defined by coefficients f and g, the assumption $\max(|c_2(\delta)|, |c_3(\delta)|) < 1/L$ can be rephrased into $\delta = O(1/L)$, by considering limit approximations of coefficients $c_2(\delta)$ and $c_3(\delta)$ in the asymptotic regime $\delta \to 0$. Below, we present a rigorous expression of this upper bound on δ for the noising schemes considered in this paper, that is the *Variance-Preserving* approach (see Appendix E.2.2) and the *Variance-Exploding* approach (see Appendix E.2.3).

E.2.2 Variance-Preserving diffusion

Consider the noising SDE (6.2) where $f(t) = -g^2(t)/2$ and g being such that $\int_0^T g^2(s)\mathrm{d}sg1$, with arbitrary volatility coefficient $\sigma > 0$,

$$\mathrm{d}X_t = -\frac{g^2(t)X_t}{2}\mathrm{d}t + \sigma g(t)\mathrm{d}W_t\ ,\ X_0 \sim \pi\ . \tag{E.3}$$

This noising scheme, known as the *Variance-Preserving* (VP) scheme [Song et al., 2021b], is largely used in score-based generative models. In the following, we denote $\alpha_t = \int_0^t g^2(t)\mathrm{d}t$ for any $t \in [0,T]$.

On the choice of the g-schedule. Previous works have considered a linear schedule $g^2(t) = \beta_{\min}(1-t/T) + \beta_{\max}(t/T)$ where $\beta_{\min} = 0.1$, $\beta_{\max} \in \{10, 20\}$ and $T = 1$, see e.g., Song et al. [2021b] or cosine parameterization [Nichol and Dhariwal, 2021], which has been proved to perform better in generative modeling. In our sampling experiments, we did not observe any significant difference between these two settings. Hence, we fix the linear schedule to be the default setting for our numerics, and let σ be arbitrarily chosen.

Lemma 100 (Exact noising SDE integration - VP case)**.** *The conditional distribution of X_t given $X_s = x_s \in \mathbb{R}^d$ is defined by the Gaussian kernel*

$$q_{t|s}(\cdot|x_s) = \mathrm{N}\left(\sqrt{1-\lambda^f_{s,t}}x_s, \sigma^2\lambda^f_{s,t}\mathrm{I}_d\right)\ ,$$

with $\lambda^f_{s,t} = 1 - \exp(\alpha_s - \alpha_t)$. Since $p_T(x) = \int_{\mathbb{R}^d} p_{T|0}(x|x_0)\mathrm{d}\pi(x_0)$, it results that $p_T \approx \mathrm{N}(0, \sigma^2\,\mathrm{I}_d)$.

Proof. Lemma 92 applied on noising SDE (E.3). □

Based on the previous lemma, the interpolation coefficients in Equation (6.3) are given by

$$S(t) = \exp(-\alpha_t/2) \text{ and } \sigma(t) = \sigma\sqrt{1-\exp(-\alpha_t)}.$$

In particular, $t \mapsto \sigma(t)$ is not explicitly invertible. Following Lemma 100, the VP scheme is an 'ergodic' noising scheme, converging exponentially fast to the Gaussian distribution $\mathrm{N}(0, \sigma^2 \mathrm{I}_d)$; therefore, we have $\pi^{\text{base}} = \mathrm{N}(0, \sigma^2 \mathrm{I}_d)$ in this setting. Moreover, under mild assumptions on π, the denoising SDE (6.5) writes as

$$\mathrm{d}X_t = -\frac{g^2(t)}{2}\{X_t + 2\sigma^2 \nabla \log p_t(X_t)\}\mathrm{d}t + \sigma g(t)\mathrm{d}\tilde{B}_t, \; X_T \sim \pi^{\text{base}} \,. \tag{E.4}$$

To integrate this SDE (or the equivalent probability flow ODE), one could turn to the formulas introduced in Appendix E.2.2, by replacing general coefficients with VP coefficients. Instead, we propose to rely on Exponential Integration (EI) formulas dispensed in Lemma 92 (SDE case) and Lemma 93 (ODE case), that make exact the integration of the linear part of the drift.

Lemma 101 (Approximate denoising SDE EI-based integration - VP case)**.** *The conditional distribution of X_t given $X_s = x_s \in \mathbb{R}^d$ may be approximated by the Gaussian kernel*

$$q_{s|t}(\cdot|x_t) = \mathrm{N}\left(\sqrt{1+\lambda^b_{s,t}}x_t + 2\sigma^2\left\{\sqrt{1+\lambda^b_{s,t}} - 1\right\}\mathbf{s}_t\left(x_t\right), \sigma^2\lambda^b_{s,t}\,\mathrm{I}_d\right) \,,$$

with $\lambda^b_{s,t} = \exp(\alpha_t - \alpha_s) - 1$.

Proof. Lemma 92 applied on denoising SDE (E.4). □

Lemma 102 (Approximate noising ODE EI-based integration - VP case)**.** *The solution at time t of the forward probability flow ODE (6.6) starting from $x_s \in \mathbb{R}^d$ at time s may be approximated in two ways:*

$$\tilde{\mathrm{T}}_{t|s}(x_s) = \sqrt{1-\lambda^f_{s,t}}x_s + \sigma^2\left\{\sqrt{1-\lambda^f_{s,t}} - 1\right\}\mathbf{s}_s\left(x_s\right) \qquad \textit{(Euler method : explicit)}$$

$$\mathrm{T}_{t|s}(x_s) = \sqrt{1-\lambda^f_{s,t}}x_s + \sigma^2\left\{\sqrt{1-\lambda^f_{s,t}} - 1\right\}\mathbf{s}_{(s+t)/2}\left(\frac{x_s + \mathrm{T}_{t|s}(x_s)}{2}\right) \qquad \textit{(Midpoint method : implicit)}$$

Proof. Lemma 93 applied on forward time ODE (6.6). □

Lemma 103 (Approximate denoising ODE EI-based integration - VP case)**.** *The solution at time s of the probability flow ODE (6.6) starting from $x_t \in \mathbb{R}^d$ at time t may be approximated in two ways:*

$$\tilde{\mathrm{T}}_{s|t}(x_t) = \sqrt{1+\lambda^b_{s,t}}x_t + \sigma^2\left\{\sqrt{1+\lambda^b_{s,t}} - 1\right\}\mathbf{s}_t\left(x_t\right) \qquad \textit{(Euler method: explicit)}$$

$$\mathrm{T}_{s|t}(x_t) = \sqrt{1+\lambda^b_{s,t}}x_t + \sigma^2\left\{\sqrt{1+\lambda^b_{s,t}} - 1\right\}\mathbf{s}_{(s+t)/2}\left(\frac{\mathrm{T}_{s|t}(x_t) + x_t}{2}\right) \qquad \textit{(Midpoint : implicit)}$$

Proof. Lemma 93 applied on backward time ODE (6.6). □

Remark on the mutual invertibility of the ODE integrators. The noising and denoising implicit Midpoint integrators described above are mutual inversible maps, *i.e.*, $\mathrm{T}_{s|t} \circ \mathrm{T}_{t|s} = \mathrm{T}_{t|s} \circ \mathrm{T}_{s|t} = \mathrm{Id}$. This is due to the identity $(1+\lambda^b_{s,t})^{-1} = 1 - \lambda^f_{s,t}$. This is not the case for the Euler maps $\tilde{\mathrm{T}}_{s|t}$ and $\tilde{\mathrm{T}}_{t|s}$.

Simplification of δ-assumption in Lemma 98 and Proposition 99. Following the notation introduced in Lemma 98, we obtain simplifications of $c_2(\delta)$ and $c_3(\delta)$ in the specific VP case, for any positive step-size δ, that are given by

$$c_2(\delta) = \sigma^2\left(\frac{4}{\delta g^2\left((s+t)/2\right)} + 1\right)^{-1} \,, \; c_3(\delta) = \sigma^2\left(\frac{4}{\delta g^2\left((s+t)/2\right)} - 1\right)^{-1} \,.$$

Hence, for any given $L > 0$, if we have $\delta < 4/\{(\sigma^2 L + 1)g^2\left((s+t)/2\right)\}$, then it comes that $\max\left(|c_2(\delta)|, |c_3(\delta)|\right) < 1/L$. In particular, we may use this upper bound on δ as a more readable δ-assumption in Lemma 98 and Proposition 99.

Lemma 104 (Formula for the Jacobian of the Midpoint integrators - VP case)**.** *Let $\delta > 0$, and let define the numerical constants $c_1(\delta)$, $c_2(\delta)$ and $c_3(\delta)$ as*

$$c_1(\delta) = \sqrt{1-\lambda^f_{s,t}}\ ,\ c_2(\delta) = \frac{\sigma^2}{2}\left\{1-\exp\left(\frac{\alpha_s-\alpha_t}{2}\right)\right\}\ ,\ c_3(\delta) = \frac{\sigma^2}{2}\left\{\exp\left(\frac{\alpha_t-\alpha_s}{2}\right)-1\right\}\ .$$

Consider the same notation as in Lemma 102 and Lemma 103. Assume that there exists $L > 0$ such that $\mathbf{s}_{(s+t)/2}$ is L-Lipschitz. If we further assume that $(\alpha_t - \alpha_s) < 2\log\left(1 + 2/(L\sigma^2)\right)$, then the Jacobians of Midpoint integration maps $\mathrm{T}_{t|s}$ and $\mathrm{T}_{s|t}$, respectively denoted by $J_{t|s}$ and $J_{s|t}$, verify for any inputs $x_s \in \mathbb{R}^d$ and $x_t \in \mathbb{R}^d$

$$\begin{aligned} J_{t|s}(x_s) &= c_1(\delta)\left(\mathrm{I}_d + c_2(\delta)A(x_s)\right)^{-1}\left(\mathrm{I}_d - c_3(\delta)A(x_s)\right)\ , \\ J_{s|t}(x_t) &= c_1(\delta)^{-1}\left(\mathrm{I}_d - c_3(\delta)B(x_t)\right)^{-1}\left(\mathrm{I}_d + c_2(\delta)B(x_t)\right)\ , \end{aligned}$$

where $A(x_s) = \mathbf{H}_{(s+t)/2}\left(\frac{x_s + \mathrm{T}_{t|s}(x_s)}{2}\right)$ and $B(x_t) = \mathbf{H}_{(s+t)/2}\left(\frac{x_t + \mathrm{T}_{s|t}(x_t)}{2}\right)$.

Proof. The result from Lemma 104 follows from the factorization of the following identities, inherited from the implicit expressions of $\mathrm{T}_{t|s}$ and $\mathrm{T}_{s|t}$,

$$\begin{aligned} J_{t|s}(x_s) &= \left(\mathrm{I}_d - \frac{\sigma^2}{2}\left\{\sqrt{1-\lambda^f_{s,t}}-1\right\}A(x_s)\right)^{-1}\left(\sqrt{1-\lambda^f_{s,t}}\mathrm{I}_d + \frac{\sigma^2}{2}\left\{\sqrt{1-\lambda^f_{s,t}}-1\right\}A(x_s)\right)\ , \\ J_{s|t}(x_t) &= \left(\mathrm{I}_d - \frac{\sigma^2}{2}\left\{\sqrt{1+\lambda^b_{s,t}}-1\right\}B(x_t)\right)^{-1}\left(\sqrt{1+\lambda^b_{s,t}}\mathrm{I}_d + \frac{\sigma^2}{2}\left\{\sqrt{1+\lambda^b_{s,t}}-1\right\}B(x_t)\right)\ . \end{aligned}$$

Here, the additional assumption on the term $(\alpha_t - \alpha_s)$ may be seen as the EI-based analog to the assumption on the step size $\delta = t - s$ in Lemma 98. Indeed, if we have $(\alpha_t - \alpha_s) < 2\log\left(1 + 2/(L\sigma^2)\right)$, then it comes that $\max(|c_2(\delta)|, |c_3(\delta)|) < 1/L$, which thus guarantees the invertibility of the matrices $\mathrm{I}_d + c_2(\delta)A(x_s)$ and $\mathrm{I}_d - c_3(\delta)B(x_t)$. □

Proposition 105 (Exact expression of the Jacobian log-determinants of the Midpoint integrators via power series - VP case)**.** *Consider the same notation as in Lemma 104. Assume that there exists $L > 0$ such that $\mathbf{s}_{(s+t)/2}$ is L-Lipschitz. If we further assume that $(\alpha_t - \alpha_s) < 2\log\left(1 + 2/(L\sigma^2)\right)$, then, for any inputs $x_s \in \mathbb{R}^d$ and $x_t \in \mathbb{R}^d$, we have*

$$\begin{aligned} \log\left|\det J_{t|s}(x_s)\right| &= \sum_{i=0}^{\infty} a_i(s,t)\,\mathrm{Tr}([A(x_s)]^i)\ , \\ \log\left|\det J_{s|t}(x_t)\right| &= \sum_{i=0}^{\infty} b_i(s,t)\,\mathrm{Tr}([B(x_t)]^i)\ , \end{aligned}$$

where $\{a_i(s,t), b_i(s,t)\}_{i=0}^{\infty}$ are numerical coefficients defined by

$$\begin{aligned} a_0(s,t) &= -b_0(s,t) = \frac{d}{2}(\alpha_s - \alpha_t)\ , \\ a_i(s,t) &= \frac{\sigma^{2i}}{2^i i}\left(\exp\left(\frac{\alpha_s-\alpha_t}{2}\right)-1\right)^i\left(1-(-1)^i\exp\left(-i\frac{\alpha_s-\alpha_t}{2}\right)\right) \text{ for any } i \geq 1\ , \\ b_i(s,t) &= \frac{\sigma^{2i}}{2^i i}\left(\exp\left(-\frac{\alpha_s-\alpha_t}{2}\right)-1\right)^i\left(1-(-1)^i\exp\left(i\frac{\alpha_s-\alpha_t}{2}\right)\right) \text{ for any } i \geq 1\ . \end{aligned}$$

Proof. Similarly to the proof of Proposition 99, we combine the results of Lemma 104 and Corollary 91 to get the final result. Intermediary simplifications of the terms are omitted here to help the reading. □

E.2.3 Variance-Exploding diffusion

Consider the case where $f(t) = 0$. Then, SDE (6.2) simply writes as

$$\mathrm{d}X_t = g(t)\mathrm{d}W_t,\ X_0 \sim \pi\ . \tag{E.5}$$

This noising scheme is known as the *Variance-Exploding* (VE) scheme [Song et al., 2021b].

On the choice of the g-schedule. Following the guidelines from [Karras et al., 2022], we consider the geometric schedule

$$g^2(t) = \sigma_{\min}^2 \left(\frac{\sigma_{\max}^2}{\sigma_{\min}^2}\right)^t \log\left(\frac{\sigma_{\max}^2}{\sigma_{\min}^2}\right) ,$$

where $\sigma_{\min} \approx 0$ and $\sigma_{\max} g1$ can be arbitrarily chosen.

Lemma 106 (Exact noising SDE integration - VE case)**.** *The conditional distribution of X_t given $X_s = x_s \in \mathbb{R}^d$ is defined by the Gaussian kernel*

$$q_{t|s}(\cdot|x_s) = \mathrm{N}(x_s, \lambda_{s,t}\mathrm{I}_d) \ , \ \textit{with } \lambda_{s,t} = \sigma_{min}^2 \left(\frac{\sigma_{max}}{\sigma_{min}}\right)^{2s} \left(\left(\frac{\sigma_{max}}{\sigma_{min}}\right)^{2(t-s)} - 1\right)$$

Since $p_T(x) = \int_{\mathbb{R}^d} q_{T|0}(x|x_0)\mathrm{d}\pi(x_0)$, it results that $p_T \approx \mathrm{N}(0, \sigma_{max}^2 \mathrm{I}_d)$.

Proof. Lemma 92 applied on noising SDE (E.5). □

Based on the previous lemma, the interpolation coefficients in (6.3) are given by

$$S(t) = 1 \text{ and } \sigma(t) = \sigma_{\min}\sqrt{\left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)^{2t} - 1}.$$

In particular, $t \mapsto \sigma(t)$ is explicitly invertible, since we have

$$\sigma^{-1}(\sigma) = \frac{\log\left(\left(\frac{\sigma}{\sigma_{\min}}\right)^2 + 1\right)}{2\log\left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)} .$$

Time discretization. We define $t_{\text{start}} = \sigma^{-1}(\sigma_{\min})$ and $t_{\text{end}} = \sigma^{-1}(\sigma_{\max})$. With proper choices of $\sigma_{\min}$ and $\sigma_{\max}$, we have $t_{\text{start}} \approx 0$ and $t_{\text{end}} \approx T$. Following (E.1), the SNR-adapted time discretization $\{t_k\}_{k=0}^K$ between t_{start} and t_{end} is given by

$$t_k = \frac{\log\left(\left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)^{\frac{2k}{K}} + 1\right)}{2\log\left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)} .$$

Under mild assumptions on π, the denoising SDE (6.5) writes as

$$\mathrm{d}X_t = -g^2(t)\nabla \log p_t(X_t)\mathrm{d}t + g^2(t)\mathrm{d}\tilde{B}_t, \ X_T \sim \pi^{\text{base}} . \tag{E.6}$$

Similarly to the VP case (see Appendix E.2.2), we present below approximate transition kernels and maps based on the Exponential Integration (EI). Since the linear drift term is 0 here, the EI strategy amounts to exactly integrate the time-dependent coefficient associated to the (unknown) score drift term.

Lemma 107 (Approximate denoising SDE EI-based integration - VE case)**.** *The conditional distribution of X_t given $X_s = x_s \in \mathbb{R}^d$ may be approximated by the Gaussian kernel*

$$q_{s|t}(\cdot|x_t) = \mathrm{N}\left(x_t + \lambda_{s,t}\mathbf{s}_t\left(x_t\right), \lambda_{s,t}\,\mathrm{I}_d\right) .$$

Proof. Lemma 92 applied on denoising SDE (E.6). □

Lemma 108 (Approximate noising ODE EI-based integration - VE case)**.** *The solution at time t of the forward probability flow ODE (6.6) starting from $x_s \in \mathbb{R}^d$ at time s may be approximated in two ways:*

$$\tilde{\mathrm{T}}_{t|s}(x_s) = x_s - \frac{\lambda_{s,t}}{2}\mathbf{s}_s\left(x_s\right) \qquad \textit{(Euler method : explicit)}$$

$$\mathrm{T}_{t|s}(x_s) = x_s - \frac{\lambda_{s,t}}{2}\mathbf{s}_{(s+t)/2}\left(\frac{x_s + \mathrm{T}_{t|s}(x_s)}{2}\right) \qquad \textit{(Midpoint method : implicit)}$$

Proof. Lemma 93 applied on forward time ODE (6.6). □

Lemma 109 (Approximate denoising ODE EI-based integration - VE case)**.** *The solution at time s of the backward probability flow ODE (6.6) starting from $x_t \in \mathbb{R}^d$ at time t may be approximated in two ways:*

$$\tilde{\mathrm{T}}_{s|t}(x_t) = x_t + \frac{\lambda_{s,t}}{2}\mathbf{s}_t\left(x_t\right) \qquad \textit{(Euler method : explicit)}$$

$$\mathrm{T}_{s|t}(x_t) = x_t + \frac{\lambda_{s,t}}{2}\mathbf{s}_{(s+t)/2}\left(\frac{\mathrm{T}_{s|t}(x_t) + x_t}{2}\right) \qquad \textit{(Midpoint method : implicit)}$$

Proof. Lemma 93 applied on backward time ODE (6.6). □

Remark on the mutual invertibility of the ODE integrators. The noising and denoising implicit Midpoint integrators described above are mutual inversible maps, *i.e.*, $\mathrm{T}_{s|t} \circ \mathrm{T}_{t|s} = \mathrm{T}_{t|s} \circ \mathrm{T}_{s|t} = \mathrm{Id}$. This is not the case for the Euler maps $\tilde{\mathrm{T}}_{s|t}$ and $\tilde{\mathrm{T}}_{t|s}$.

Simplification of δ-assumption in Lemma 98 and Proposition 99. Following the notation introduced in Lemma 98, we obtain simplifications of $c_2(\delta)$ and $c_3(\delta)$ in the specific VE case, for any positive step-size δ, that are given by

$$c_2(\delta) = c_3(\delta) = \frac{\delta}{4}g^2\left(\frac{s+t}{2}\right) .$$

Hence, for any given $L > 0$, if we have $\delta < 4/\{Lg^2\left((s+t)/2\right)\}$, then it comes that $\max\left(|c_2(\delta)|, |c_3(\delta)|\right) < 1/L$. In particular, we may use this upper bound on δ as a more readable δ-assumption in Lemma 98 and Proposition 99.

Lemma 110 (Formula for the Jacobian of the Midpoint integrators - VE case)**.** *Let $\delta > 0$, and let define the numerical constants $c_1(\delta)$, $c_2(\delta)$ and $c_3(\delta)$ as*

$$c_1(\delta) = 1 \ , \ c_2(\delta) = c_3(\delta) = \frac{\lambda_{s,t}}{4} .$$

Consider the same notation as in Lemma 108 and Lemma 109. Assume that there exists $L > 0$ such that $\mathbf{s}_{(s+t)/2}$ is L-Lipschitz. If we further assume that $\lambda_{s,t} < 4/L$, then the Jacobians of Midpoint integration maps $\mathrm{T}_{t|s}$ and $\mathrm{T}_{s|t}$, respectively denoted by $J_{t|s}$ and $J_{s|t}$, verify for any inputs $x_s \in \mathbb{R}^d$ and $x_t \in \mathbb{R}^d$

$$J_{t|s}(x_s) = c_1(\delta)\left(\mathrm{I}_d + c_2(\delta)A(x_s)\right)^{-1}\left(\mathrm{I}_d - c_3(\delta)A(x_s)\right) ,$$
$$J_{s|t}(x_t) = c_1(\delta)^{-1}\left(\mathrm{I}_d - c_3(\delta)B(x_t)\right)^{-1}\left(\mathrm{I}_d + c_2(\delta)B(x_t)\right) ,$$

where $A(x_s) = \mathbf{H}_{(s+t)/2}\left(\frac{x_s + \mathrm{T}_{t|s}(x_s)}{2}\right)$ and $B(x_t) = \mathbf{H}_{(s+t)/2}\left(\frac{x_t + \mathrm{T}_{s|t}(x_t)}{2}\right)$.

Proof. The result from Lemma 110 follows from the factorization of the following identities, inherited from the implicit expressions of $\mathrm{T}_{t|s}$ and $\mathrm{T}_{s|t}$,

$$J_{t|s}(x_s) = \left(\mathrm{I}_d + \frac{\lambda_{s,t}}{4}A(x_s)\right)^{-1}\left(\mathrm{I}_d - \frac{\lambda_{s,t}}{4}A(x_s)\right) ,$$
$$J_{s|t}(x_t) = \left(\mathrm{I}_d - \frac{\lambda_{s,t}}{4}B(x_t)\right)^{-1}\left(\mathrm{I}_d + \frac{\lambda_{s,t}}{4}B(x_t)\right) .$$

Here, the additional assumption on the term $\lambda_{s,t}$ may be seen as the EI-based analog to the assumption on the step size $\delta = t - s$ in Lemma 98. Indeed, if we have $\lambda_{s,t} < 4/L$, then it comes that $\max(|c_2(\delta)|, |c_3(\delta)|) < 1/L$, which thus guarantees the invertibility of the matrices $\mathrm{I}_d + c_2(\delta)A(x_s)$ and $\mathrm{I}_d - c_3(\delta)B(x_t)$. □

Proposition 111 (Exact expression of the Jacobian log-determinants of the Midpoint integrators via power series - VE case)**.** *Consider the same notation as in Lemma 110. Assume that there exists $L > 0$ such that*

$\mathbf{s}_{(s+t)/2}$ *is L-Lipschitz. If we further assume that* $\lambda_{s,t} < 4/L$*, then, for any inputs* $x_s \in \mathbb{R}^d$ *and* $x_t \in \mathbb{R}^d$*, we have*

$$\log\left|\det J_{t|s}(x_s)\right| = \sum_{i=0}^{\infty} a_i(s,t)\,\mathrm{Tr}([A(x_s)]^i)\ ,$$
$$\log\left|\det J_{s|t}(x_t)\right| = \sum_{i=0}^{\infty} b_i(s,t)\,\mathrm{Tr}([B(x_t)]^i)\ ,$$

where $\{a_i(s,t), b_i(s,t)\}_{i=0}^{\infty}$ *are numerical coefficients defined by*

$$a_i(s,t) = b_i(s,t) = 0 \text{ for any even } i \in \mathbb{N}\ \ ,$$
$$a_i(s,t) = -b_i(s,t) = -2\frac{\lambda_{s,t}^i}{4^i} \text{ for any odd } i \in \mathbb{N}\ .$$

Proof. Similarly to the proof of Proposition 99, we combine the results of Lemma 110 and Corollary 91 to get the final result. Intermediary simplifications of the terms are omitted here to help the reading. □

E.3 Proofs of Section 6.4

In the main paper, we present our methodology to design diffusion-based aMC-BGs via deterministic transitions between noise levels, by relying on the general noising framework presented in Appendix E.2.1 to maintain a certain generality. We highlight that these results still hold within the specific EI-based framework of VP noising (see Appendix E.2.2) and VE noising (see Appendix E.2.3), based on the formulas introduced in the respective sections. We leave the proof for the reader.

Proof of Proposition 8. This is a restatement of the results of Lemma 96 and Lemma 97 in the case where $s = t_k$, $t = t_{k+1}$ and $\mathbf{s}_t = \nabla \log p_t$. The mutual invertibility property is immediate. □

We give below a formal version of **A**4.

A 13 (Score smoothness & discretization error - Formal version of **A**4)**.** *(a) There exists* $L_k > 0$ *such that* $\nabla \log p_{t_{k+1/2}}$ *is* L_k*-Lipschitz and (b) the step-size* δ_k *verifies*

$$\max\left(L_k\left|c_2(\delta_k)\right|, L_k\left|c_3(\delta_k)\right|, c_4(\delta_k, L_k)\right) < 1\ ,$$

where c_2 *and* c_3 *are given in Lemma 98, and* $c_4(\delta, L) = \frac{\delta}{2}\left(\left|f(t_{k+1/2}\right| + Lg^2(t_{k+1/2})/2\right)$.

Proof of Proposition 9. Assume **A**13. Fix current states x_k and x_{k+1}. Respectively, denote the sequences $\{\mathrm{T}^{(n)}_{k+1|k}(x_k)\}_{n\in\mathbb{N}}$ and $\{\mathrm{T}^{(n)}_{k|k+1}(x_{k+1})\}_{n\in\mathbb{N}}$ by $\{y_n\}_{n\in\mathbb{N}}$ and $\{z_n\}_{n\in\mathbb{N}}$. Respectively define the *forward* map $\Psi_{k+1|k} : \mathbb{R}^d \to \mathbb{R}^d$ and the *backward* map $\Psi_{k|k+1} : \mathbb{R}^d \to \mathbb{R}^d$ by

$$\Psi_{k+1|k}(y) = x_k + \delta_k v\left(t_{k+1/2}, \frac{x_k + y}{2}\right)\ ,\ \Psi_{k|k+1}(z) = x_{k+1} - \delta_k v\left(t_{k+1/2}, \frac{z + x_{k+1}}{2}\right)\ ,$$

such that $y_{n+1} = \Psi_{k+1|k}(y_n)$ and $z_{n+1} = \Psi_{k|k+1}(z_n)$ for any $n \in \mathbb{N}$. By combining **A**13-(a) and $c_4(\delta_k, L_k) < 1$ from **A**13-(b), it is easy to see that both maps $\Psi_{k+1|k}$ and $\Psi_{k|k+1}$ are contractive Lipschitz mappings. We directly obtain the result by application of Banach fixed-point theorem. □

Lemma 112 (Formula for the Jacobian of the IM integrator)**.** *Following the same notation as in Proposition 8, under* ***A****4, the Jacobians of* $\mathrm{T}_{k|k+1}$ *and* $\mathrm{T}_{k+1|k}$ *verify*

$$J_{\mathrm{T}_{k+1|k}}(x_k) = c_1\left(\mathrm{I}_d + c_2 A(x_k)\right)^{-1}\left(\mathrm{I}_d - c_3 A(x_k)\right)\ ,$$
$$J_{\mathrm{T}_{k|k+1}}(x_{k+1}) = c_1^{-1}\left(\mathrm{I}_d - c_3 B(x_{k+1})\right)^{-1}\left(\mathrm{I}_d + c_2 B(x_{k+1})\right)\ ,$$

where

$$A(x_k) = \mathbf{H}_{t_{k+1/2}}\left(\frac{x_k + \mathrm{T}_{k+1|k}(x_k)}{2}\right)\ ,\ B(x_{k+1}) = \mathbf{H}_{t_{k+1/2}}\left(\frac{x_{k+1} + \mathrm{T}_{k|k+1}(x_{k+1})}{2}\right)\ ,$$

$\mathbf{H}_{t_{k+1/2}}$ *is the Hessian of* $\log p_{t_{k+1/2}}$*, and* c_1*,* c_2*,* c_3 *are the numerical constants given in Lemma 98.*

Proof of Lemma 112. This is a restatement of Lemma 98 in the case where $s = t_k$, $t = t_{k+1}$, with exact score and Hessian functions ($\nabla \log p_{t_{k+1/2}}$ and $\mathbf{H}_{t_{k+1/2}}$) used for $\mathbf{s}$ and $\mathbf{H}$. In particular, the assumptions that are needed for this result are verified by **A**13. □

Note that similar constants to those introduced in Proposition 8 are derived for VP, respectively VE, noising scheme combined with exponential integration in Lemma 104, respectively Lemma 110, under small change in the assumption.

Proof of Proposition 10. Assume **A**13. Consider a prescribed fixed-point range $M \geq 1$ satisfying (6.25). We first consider the approximations of $A(x_k)$ and $B(x_{k+1})$, introduced in Lemma 112, obtained by replacing the intractable implicit map evaluations with their fixed-point estimations, see (6.23) and (6.24). This leads to the following expressions

$$A^{(M)}(x_k) = \mathbf{H}_{t_{k+1/2}}\left(\frac{x_k + \mathrm{T}^{(M)}_{k+1|k}(x_k)}{2}\right) , \; B^{(M)}(x_{k+1}) = \mathbf{H}_{t_{k+1/2}}\left(\frac{x_{k+1} + \mathrm{T}^{(M)}_{k|k+1}(x_{k+1})}{2}\right) .$$

By **A**13(a)-(b) we verify that we have $\|A(x_k)\| < \min(1/|c_2(\delta_k)|, 1/|c_3(\delta_k)|)$ and $\|B(x_{k+1})\| < \min(1/|c_2(\delta_k)|, 1/|c_3(\delta_k)|)$. Therefore, we encounter the same theoretical requirements as in the proof of Proposition 99 where $s = t_k$, $t = t_{k+1}$, with exact score and Hessian functions ($\nabla \log p_{t_{k+1/2}}$ and $\mathbf{H}_{t_{k+1/2}}$) used for $\mathbf{s}$ and $\mathbf{H}$, which allows us to define the *exact* expression of the Jacobian log-determinants

$$\log\left|\det J_{\mathrm{T}_{k+1|k}}(x_k)\right| = \sum_{i=0}^{\infty} a_i(t_k, t_{k+1}) \operatorname{Tr}([A(x_k)]^i) , \tag{E.7}$$

$$\log\left|\det J_{\mathrm{T}_{k+1|k}}(x_k)\right| = \sum_{i=0}^{\infty} b_i(t_k, t_{k+1}) \operatorname{Tr}([B(x_{k+1})]^i) , \tag{E.8}$$

using the numerical coefficients introduced in Proposition 99. On the other hand, we also have that $\left\|A^{(M)}(x_k)\right\| < \min(1/|c_2(\delta_k)|, 1/|c_3(\delta_k)|)$ and $\left\|B^{(M)}(x_{k+1})\right\| < \min(1/|c_2(\delta_k)|, 1/|c_3(\delta_k)|)$, which allows us to *exactly* define the following expansion series based on replacing Jacobian terms $A^{(M)}(x_k)$ and $B^{(M)}(x_{k+1})$

$$\sum_{i=0}^{\infty} a_i(t_k, t_{k+1}) \operatorname{Tr}([A^{(M)}(x_k)]^i) \text{ and } \sum_{i=0}^{\infty} b_i(t_k, t_{k+1}) \operatorname{Tr}([B^{(M)}(x_{k+1})]^i) . \tag{E.9}$$

Since we expect to have $A(x_k) \approx A^{(M)}(x_k)$ and $B(x_{k+1}) \approx B^{(M)}(x_{k+1})$, we may substitute the trace terms in (E.7) and (E.8) by those in (E.9) to approximate $\log\left|\det J_{\mathrm{T}_{k+1|k}}(x_k)\right|$ and $\log\left|\det J_{\mathrm{T}_{k|k+1}}(x_{k+1})\right|$. Finally, we obtain the result from Proposition 10 under additional approximation induced by the truncation of the power series at a given order $I \geq 1$, letting $a_{k,i} = a_i(t_k, t_{k+1})$ and $b_{k,i} = b_i(t_k, t_{k+1})$. □

In our experiments based on the VP noising scheme combined with exponential integration, we adapt the result from Proposition 10 by using the coefficients introduced in Proposition 105. Similarly, one could use the coefficients from Proposition 111 for the VE noising scheme combined with exponential integration.

E.4 Experimental details

E.4.1 Target details

Definition of the *TwoModes* target distribution. For our target π, we first consider the Gaussian mixture introduced in Grenioux et al. [2025], whose density is defined over $\mathbb{R}^d$ as

$$\gamma(x) = \tfrac{2}{3}\mathrm{N}(x; -a\mathbf{1}_d, \Sigma_1) + \tfrac{1}{3}\mathrm{N}(x; a\mathbf{1}_d, \Sigma_2) ,$$

where $\Sigma_1, \Sigma_2 \in \mathbb{R}^{d\times d}$ are diagonal covariance matrices. The diagonal entries of Σ_1 are given by $(\Sigma_1)_{i,i} = \frac{i}{d}\sigma^2_{\max} + \frac{d-i}{d}\sigma^2_{\min}$, and those of Σ_2 are the reverse of Σ_1: $(\Sigma_2)_{i,i} = (\Sigma_1)_{d-i,d-i}$, with $\sigma^2_{\max} = 0.2$ and $\sigma^2_{\min} = 0.01$ (hence, the conditioning number of each covariance matrix is 20). We vary the separation parameter $a \in \{1.0, 2.5, 5.0, 10.0\}$ and the dimension $d \in \{16, 32, 64, 128\}$. In our experiments, we rather consider the "standardized" version of γ (*i.e.*, with zero mean and unit covariance), given by the unnormalized density $x \mapsto \gamma(\Sigma_\pi^{1/2}x + \mathbf{m}_\pi)$, where $\mathbf{m}_\pi$ is the exact mean of π, and Σ_π is a diagonal covariance matrix whose entries correspond to the exact marginal variances of π along each coordinate. The mode weight absolute error metric corresponds to the absolute error $|\hat{w} - 2/3|$ when estimating the strongest mode weight via a Monte Carlo estimator $\hat{w}$ based on generated samples, see [Grenioux et al., 2025, Section 3.1] for more details.

Training method	LFPE+DSM	aLFPE+DSM	RNE+DSM
Value of λ_{reg}	$\{10^{-2}, 10^{-3}, 10^{-4}\}$	$\{10^{-2}, 10^{-3}, 10^{-4}\}$	$\{10, 1, 10^{-1}\}$

Table E.1: **Values of λ_{reg} for regularizing DSM objective with LFPE, aLFPE or RNE objective.**

Definition of the *ManyModes* target distribution. We also consider the d-dimensional Gaussian mixture with $L > 2$ components introduced in [Noble et al., 2025, Appendix H.1] defined for any $x \in \mathbb{R}^d$ by its density $\gamma(x) = \sum_{\ell=1}^{L} w_\ell \mathrm{N}(x; \mathbf{m}_\ell, 0.5\mathrm{I}_d)$, where the means $\{\mathbf{m}_\ell\}_{\ell=1}^{L}$ are sampled independently from $\mathrm{U}([-L, L]^d)$, and the weights $\{w_\ell\}_{\ell=1}^{L}$ form a strictly increasing geometric sequence such that $w_L/w_1 = 3$. We will consider $L \in \{16, 32, 64\}$ with fixed dimension $d = 32$. Moreover, we apply the same standardization procedure as for the *TwoModes* targets. To evaluate how well mode weights are recovered, we compute the Total Variation (TV) distance between the true mode weight histogram and its Monte Carlo estimate.

E.4.2 Training and sampling parameters

Diffusion model training details. As explained in Section 6.6.1, we consider two types of architectures $\mathcal{E}^\theta$ to learn the log-densities of DMs: (i) a pinned architecture, ensuring exact recovery of the target distribution π at t_0 and (ii) an hardcoded architecture, without any boundary condition fixed at training stage. For both of these models, we rely on an enhanced version of the score-like architecture advocated by Richter et al. [2023], denoted by $\mathbf{s}^\theta : (t, x) \in \mathbb{R}^{d+1} \to \mathbf{s}_t^\theta(x) \in \mathbb{R}^d$, which is a 4-layer 128-width fully connected network with GeLU activations, position-input preconditioning (based on target mean and scalar variance), time-input preconditioning (based on Fourier embedding) and time-input skip-connections at every layer. Our models are the following: (a) *Pinned*: given (6.26), we set $g_t^\theta(x) = \frac{1}{2}\left\|\mathbf{s}_t^\theta(x)\right\|_2^2$ and f^θ to be a scalar-to-scalar 4-layer 64-width fully connected network with GeLU activations and the same time-input preconditioning as in $\mathbf{s}^\theta$; (b) *Hardcoded*: we adopt the network preconditioning strategy proposed by Thornton et al. [2025] on $\mathbf{s}^\theta$.

For each setting of *TwoModes* and *ManyModes* distributions (varying by dimension, mode spacing or number of modes), we train both network architectures with the six objectives described in Appendix E.1.4, using parameterized log-density $\mathbf{U}_t^\theta = -\mathcal{E}_t^\theta$, parameterized score $\mathbf{s}_t^\theta = -\nabla_x \mathcal{E}_t^\theta$ and parameterized time score $\mathbf{u}_t^\theta = -\partial_t \mathcal{E}_t^\theta$. For TSM and tSM objectives, we consistently apply a score matching regularization by simply adding up the DSM objective term (hence with scale 1). Following the guidelines of cited energy matching works, we respectively implement LFPE+DSM, aLFPE+DSM and RNE+DSM objectives as a linear combination of the DSM objective (with scale 1) and the LFPE, aLFPE or RNE objective, multiplied by an hyperparameter λ_{reg} chosen in the range of values described in Table E.1. For the latter three methods, we systematically display, for each sampling/target setting, the best sampling results obtained among the considered values of λ_{reg}.

Regardless of the target distribution, all diffusion models are trained on datasets of size 60,000 for 1,000 epochs with a batch size of 1024. We use AdamW optimizer [Loshchilov and Hutter, 2019] and set the default learning rate as 10^{-4}, multiplied by a factor d^{-1} for score matching approaches (DSM, TSM, tSM) and d^{-2} for energy matching techniques (LFPE, aLFPE, RNE), following guidelines of related works. In particular, when training networks with TSM and RNE objectives, we initialize $\mathbf{U}^\theta$ based on the output of DSM training procedure. In the case of tSM, LFPE and aLPFE obectives, we observed that this DSM pretraining strategy was not beneficial, and even led to training instability (certainly due to arbitrary values taken by $\partial_t \mathbf{U}^\theta$ at initialization); to maintain a fair computational comparison between all methods, networks trained with tSM, LFPE and aLPFE objectives benefit from 1,000 additional training epochs (hence, 2,000 epochs in total). For each value of K considered in the sampling results, training is performed over a discrete time grid of $[0, T]$ with K levels defined by the SNR-discretization scheme evoked in Appendix E.1.2, rather than by uniform sampling in $[0, T]$. This means that a separate network is trained for each value of K, ensuring precise learning at the exact times where inference is conducted.

General remarks on annealed sampling methods. Since the considered targets are systematically standardized, we set the base distribution as their Gaussian approximation $\pi^{\text{base}} = \mathrm{N}(0, \mathrm{I}_d)$, for both tempering and diffusion-based approaches. For tempering paths, the sequence of densities defined by (6.18) is employed with a logarithmic schedule $\beta_k = 1 - \varepsilon^{k/K}$ and $\varepsilon = 10^{-5}$, following the recommendations of Grenioux et al. [2025]. For the standard SMC approach, we implemented the adaptive level selection strategy of Zhou et al. [2016], as also used in Grenioux et al. [2025]. However, within our computational upper bound of 256 intermediate levels, we did not observe any improvement over a fixed, logarithmic schedule. We therefore use the simpler fixed-schedule approach, which is also computationally cheaper. Similar adaptive strategies for optimizing the annealing schedule in RE have been proposed in the literature [Syed et al., 2021, 2022], but we restrict ourselves

here to non-adaptive settings to maintain comparability and investigate the impact of the number of intermediate levels directly. When using second-order approaches, *i.e.*, methods that require access to the Hessians of the bridging log-densities, we only exploit the diagonal of these Hessians to ensure a good compromise between accuracy and computational efficiency in high dimensional scenarios.

All SMC variants (also including diffusion-enhanced SMC samplers), as well as the standard AIS sampler, apply 160 MCMC steps (including 128 warm-up steps) for local exploration at each level $k \in \{0, \dots, K\}$, using *Metropolis-Adjusted Langevin Algorithm* (MALA) [Roberts and Tweedie, 1996]. Following the recommendations of Chopin et al. [2020], we do not perform resampling systematically, but instead apply it adaptively based on the current IS weights, using an effective sample size threshold of 30%.

For RE-based sampling methods, we perform a total of 24,576 MCMC steps (including 8,192 warm-up steps), with local exploration made via MALA and swaps occurring every 8 steps, thereby defining the computational budget of RE (with or without transition kernels) to be comparable to the footprint of the SMC setting[4] with the largest number of levels (*i.e.*, $K = 256$ where SMC performs the best), see the last row of Figure 6.3. When we target the diffusion density path, we leverage prior knowledge brought by the pre-trained diffusion model to initialize each intermediate level by simply simulating the denoising SDE (6.5), thus populating each level with target-informed particles. In Appendix E.4.4, we provide an ablation study in idealized setting **(A)** comparing the results of the related RE samplers when rather initially populating the levels with samples from π^{base} and observe that our score-informed strategy is more beneficial to RE.

For all variants of annealed samplers based on deterministic transitions, we use by default $M = 4$ fixed-point iterations, truncate the power series at order $I = 3$, and use 32 samples in the Hutchinson estimator when applicable. In Appendix E.4.4, we provide a precise ablation study of these hyperparameters in idealized setting **(A)**, to evaluate their individual impact on sampling performance. Finally, all local MALA steps are performed with an initial step size of 0.01; then, its is geometrically adapted during both warm-up and effective sampling based on local MH acceptance rates, targeting 70% acceptance.

Inference and sampling details. For diffusion-based methods, whether the path is learned or fixed, we adopt the SNR-adapted discretization from Appendix E.1.2 to establish the annealing levels : when combined with a learned path, this ensures consistency between learning and inference stages. For all AIS/SMC samplers, we use 8192 particles, and keep, for each particle, when it is available, the last 32 MCMC samples generated at the last level (properly reweighted using the associated importance weights) to compute the metrics. For all RE methods, we use 4 parallel RE chains; once the fixed global number of MCMC steps is reached, each of thse chains is subsampled by retaining only the last local MCMC state before each swap. For all annealed samplers, we repeat the sampling run 8 times to produce averaged results in the plots.

E.4.3 Additional targets and metrics

This section presents a series of extended experiments that complement the core findings of the paper by evaluating additional configurations and metrics. Specifically, Figures E.1 and E.2 expand upon Figure 6.1 on the *TwoModes* benchmark, considering various mode spacings and dimensions, and reporting sliced Wasserstein distance and absolute mode weight error, respectively. Similarly, Figures E.3, E.5, E.7 and E.9 generalize Figure 6.3 over more *TwoModes* settings by displaying the sliced Wasserstein distance results, while their analog for the mode weight error metric are respectively provided in Figures E.4, E.6, E.8 and E.10. To complete Figure 6.4, we illustrate, for the exact same settings, the failure of the annealed samplers when the DM log-densities are trained via the score/energy matching objectives introduced in Section 6.6.1 by reporting the sliced Wasserstein distance in Figures E.11 to E.34. Similarly, we display the results obtained for *ManyModes* settings in idealized setting **(A)** with Figure E.35 (sliced Wasserstein distance) and Figure E.36 (mode weight TV metric), and in realistic setting **(B)** for all considered score/energy matching objectives by reporting the related sliced Wasserstein distance in Figures E.37 to E.42. Finally, we compare in Figures E.43 and E.44 all learned density paths for 1D Gaussian mixtures introduced in Figure 6.5. We highlight that all of these additional experiments are fully consistent with the conclusions and observations from Sections 6.4 and 6.6.

[4] We however note that the version RE with deterministic transitions based on Hutchinson estimation requires twice as much sampling time as its other RE counterparts, making it the most demanding sampling method.

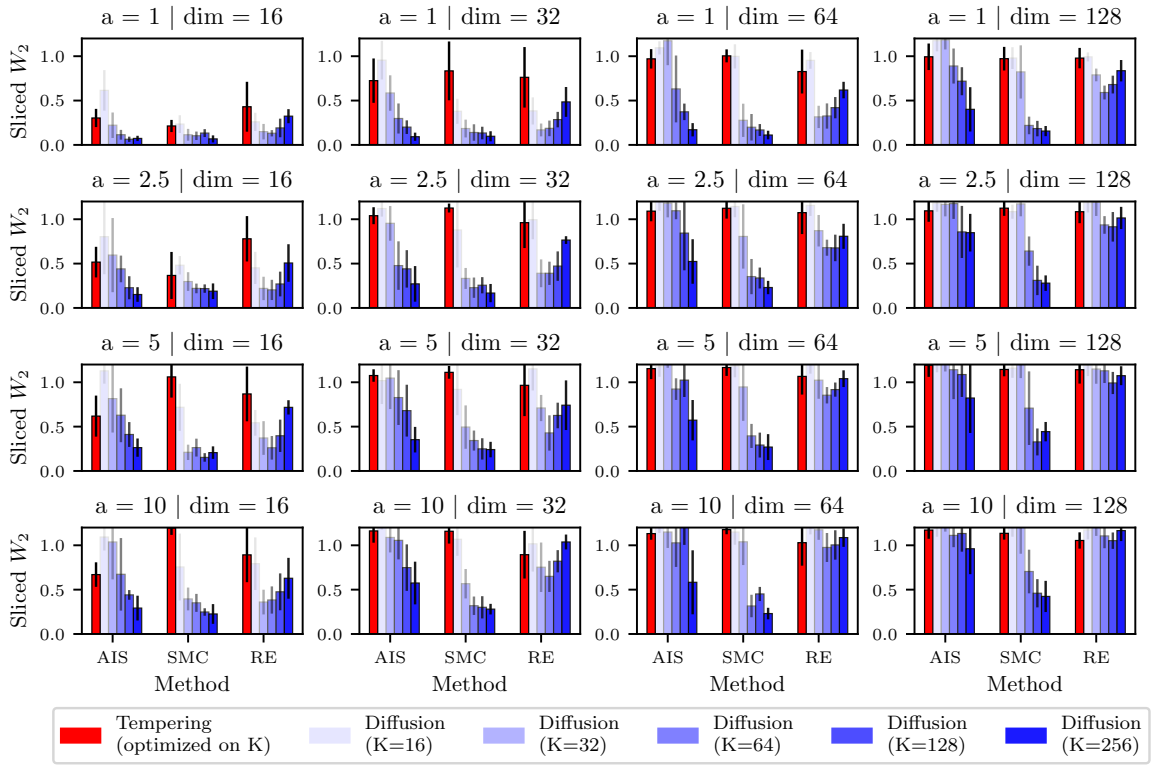


Figure E.1: **Sampling results via sliced Wasserstein distance for classic annealed samplers with diffusion (blue) and tempering (red) density paths**, when targeting *TwoModes* in different settings of mode spacing and dimension. This is complementary to Figure 6.1.

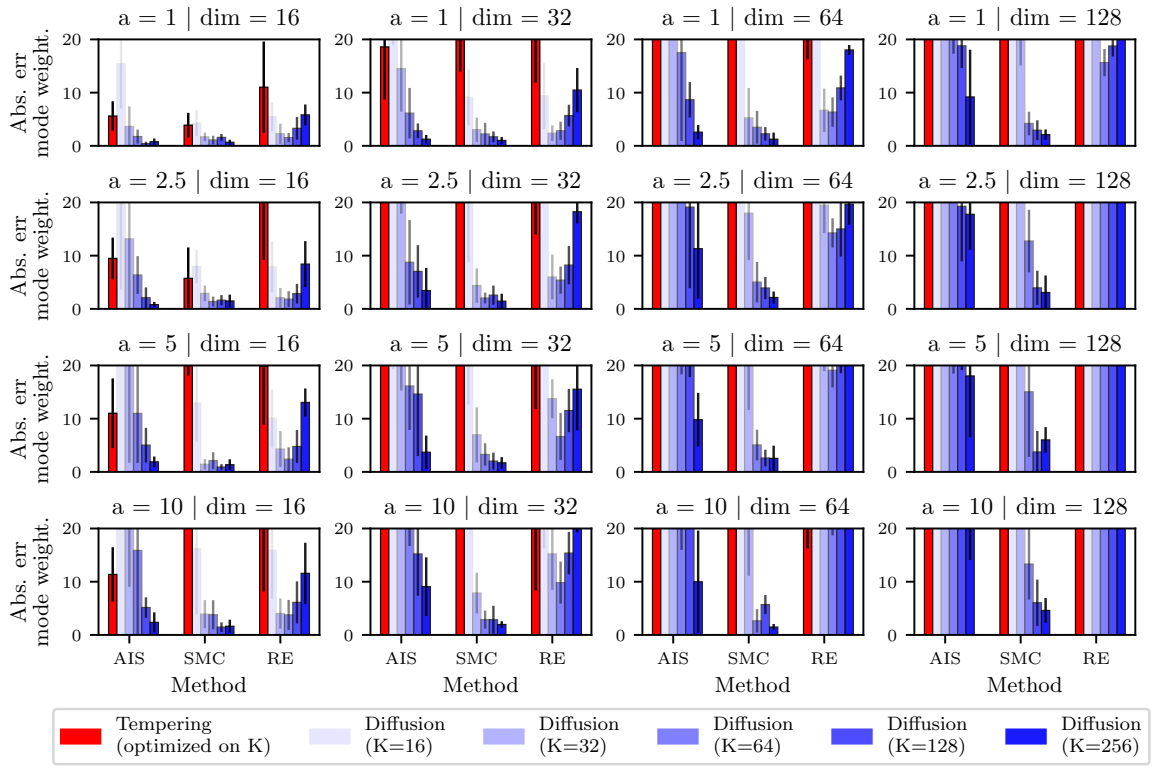


Figure E.2: **Sampling results via mode weight absolute error for classic annealed samplers with diffusion (blue) and tempering (red) density paths**, when targeting *TwoModes* in different settings of mode spacing and dimension. This is complementary to Figure 6.1 and Figure E.1.

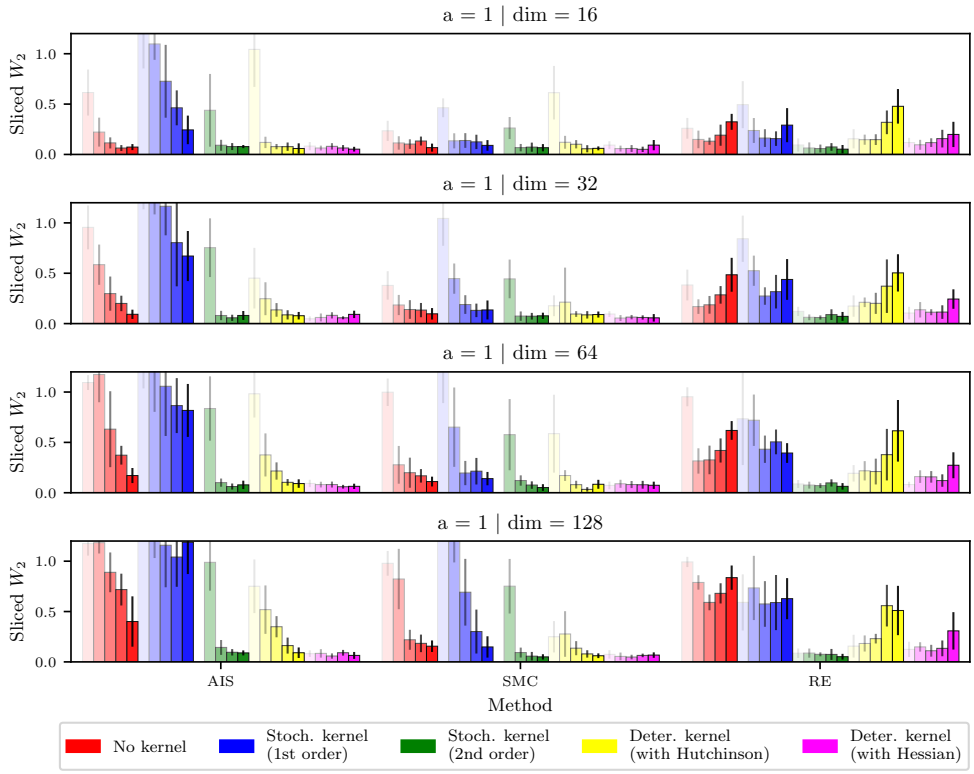


Figure E.3: **Diffusion-based aMC-BG results via sliced Wasserstein distance in idealized setting (A)**, when targeting *TwoModes* distribution with $a = 1$, for all dimensional settings. This is complementary to Figure 6.3.

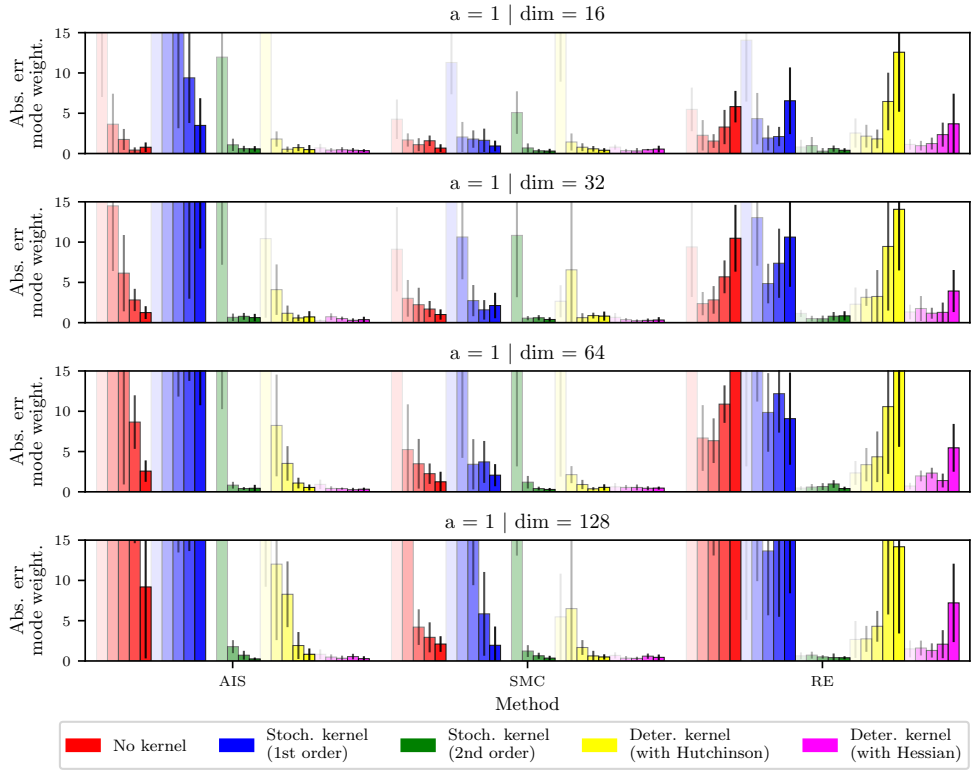


Figure E.4: **Diffusion-based aMC-BG results via mode weight absolute error in idealized setting (A)**, when targeting *TwoModes* distribution with $a = 1$, for all dimensional settings. This is complementary to Figure 6.3 and Figure E.3.

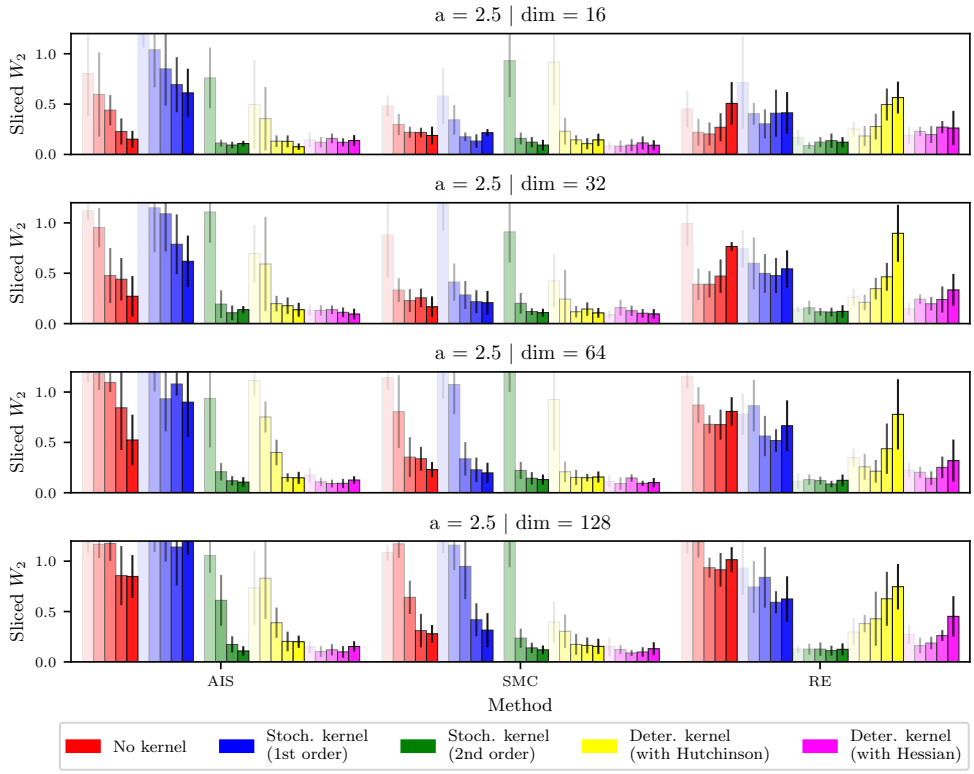


Figure E.5: **Diffusion-based aMC-BG results via sliced Wasserstein distance in idealized setting (A)**, when targeting *TwoModes* distribution with $a = 2.5$, for all dimensional settings. This is complementary to Figure 6.3.

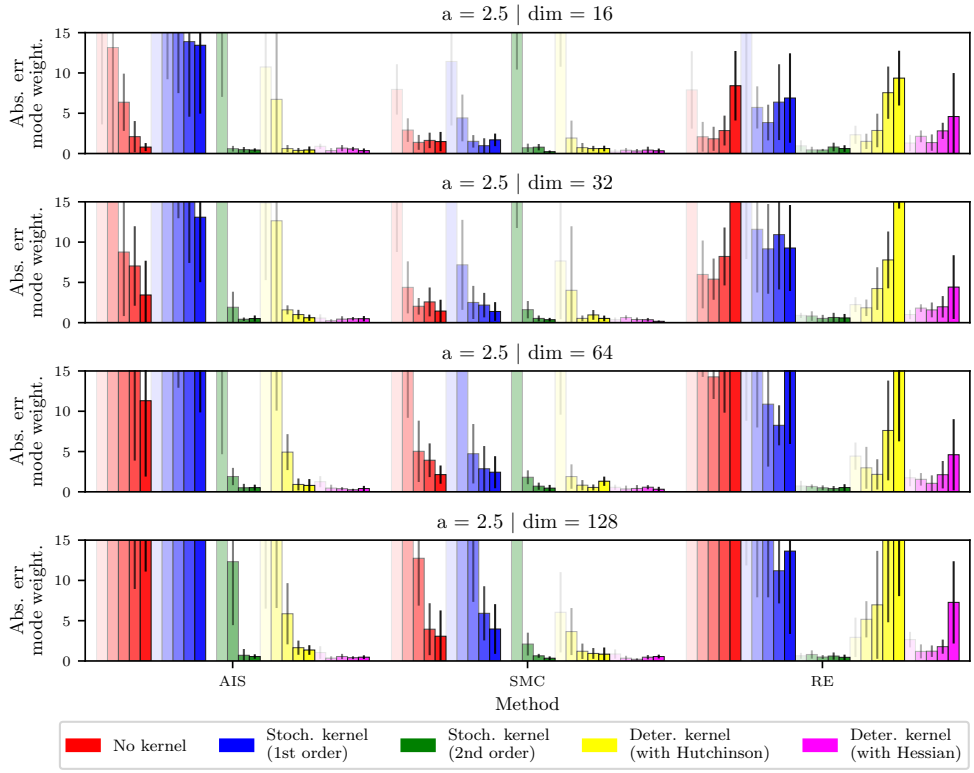


Figure E.6: **Diffusion-based aMC-BG results via mode weight absolute error in idealized setting (A)**, when targeting *TwoModes* distribution with $a = 2.5$, for all dimensional settings. This is complementary to Figure 6.3 and Figure E.5.

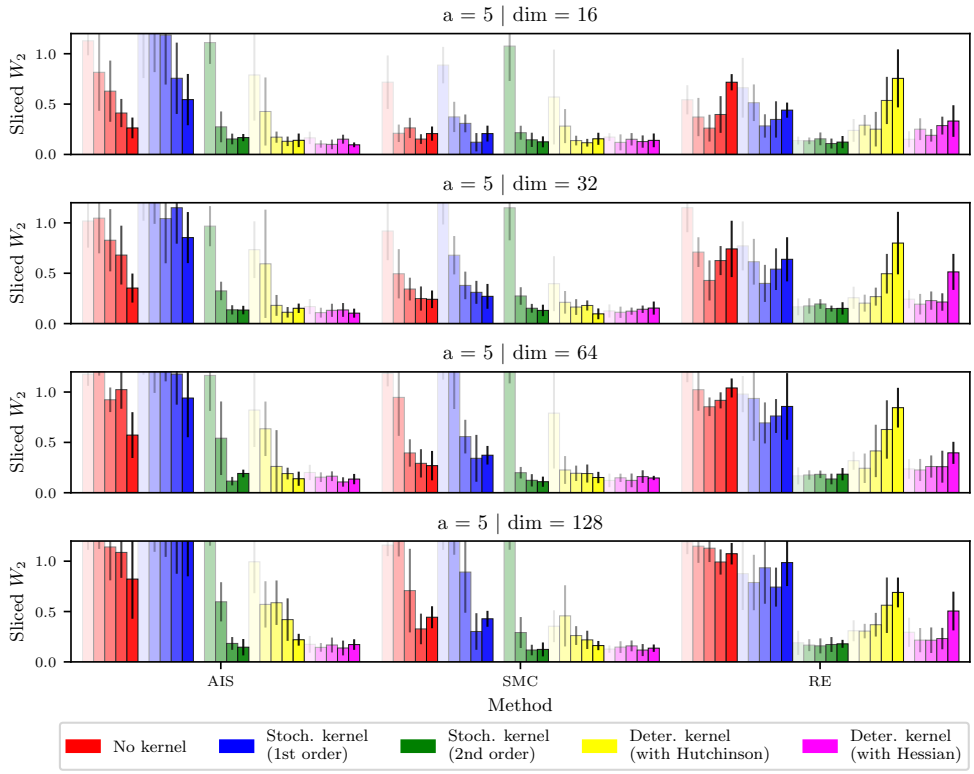


Figure E.7: **Diffusion-based aMC-BG results via sliced Wasserstein distance in idealized setting (A)**, when targeting *TwoModes* distribution with $a = 5$, for all dimensional settings. This is complementary to Figure 6.3.

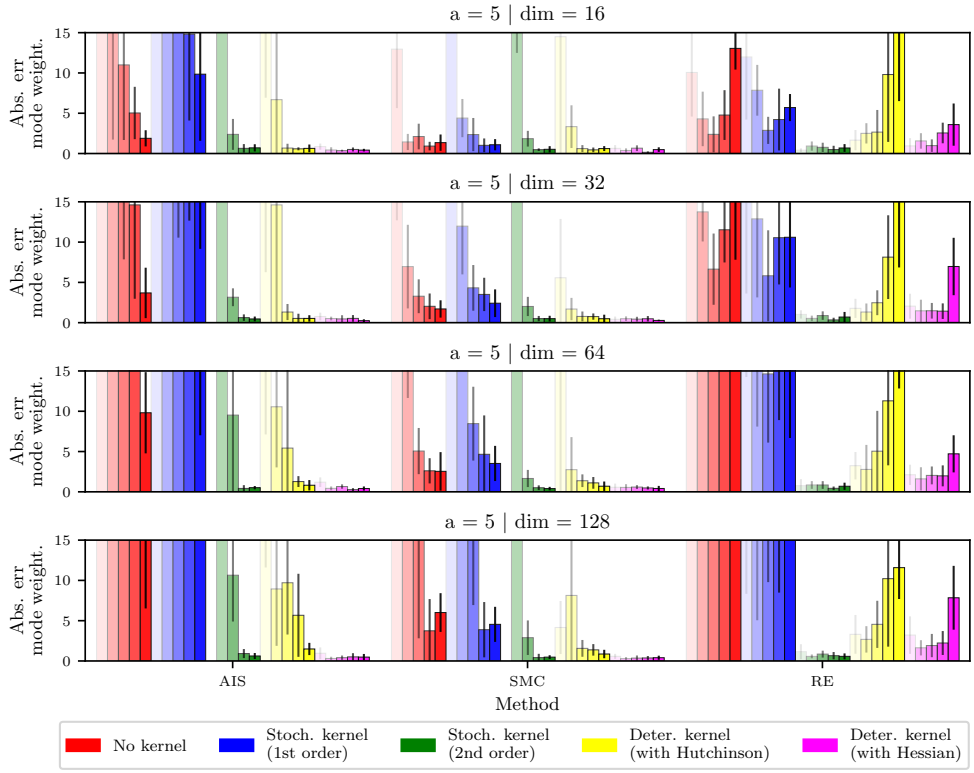


Figure E.8: **Diffusion-based aMC-BG results via mode weight absolute error in idealized setting (A)**, when targeting *TwoModes* distribution with $a = 5$, for all dimensional settings. This is complementary to Figure 6.3 and Figure E.7.

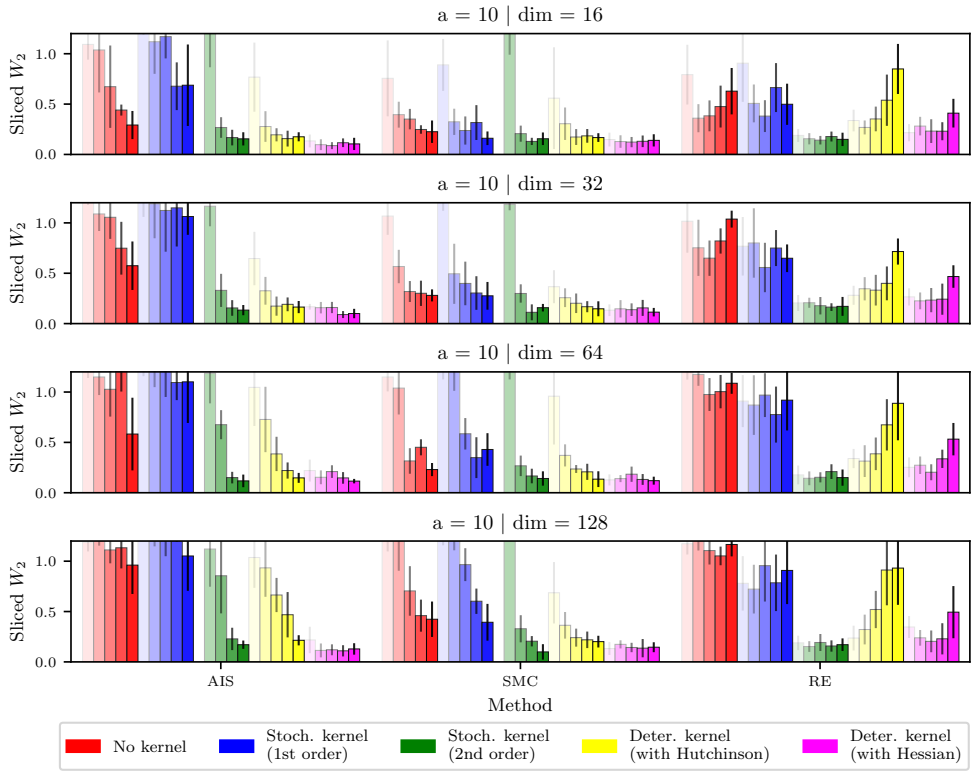


Figure E.9: **Diffusion-based aMC-BG results via sliced Wasserstein distance in idealized setting (A)**, when targeting *TwoModes* distribution with $a = 10$, for all dimensional settings. This is complementary to Figure 6.3.

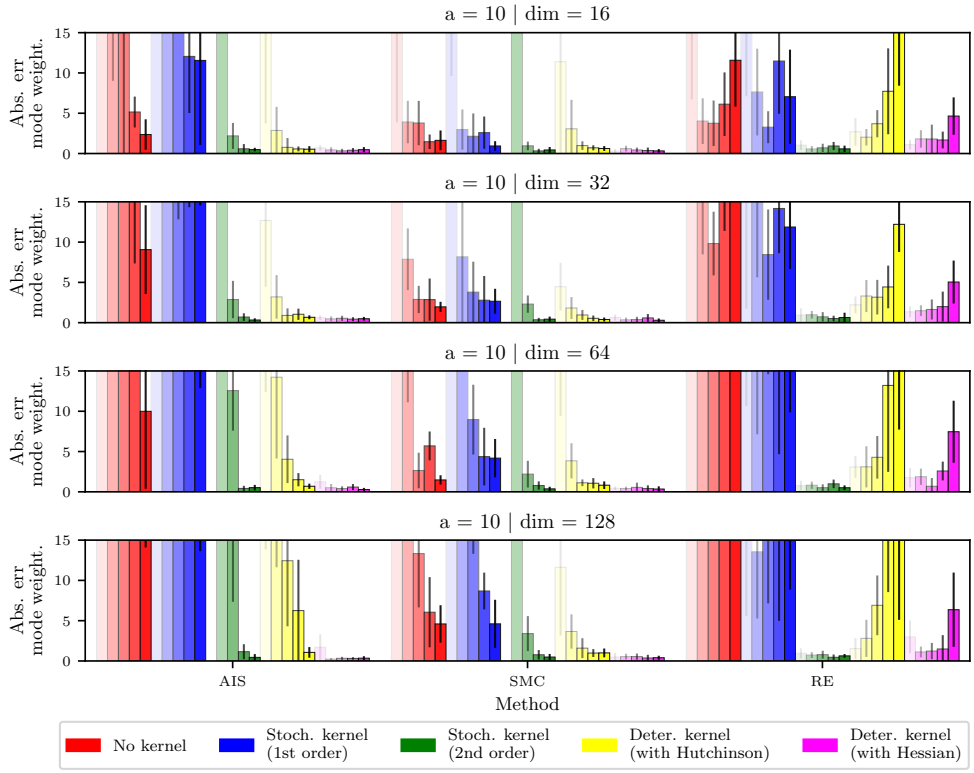


Figure E.10: **Diffusion-based aMC-BG results via mode weight absolute error in idealized setting (A)**, when targeting *TwoModes* distribution with $a = 10$, for all dimensional settings. This is complementary to Figure 6.3 and Figure E.9.

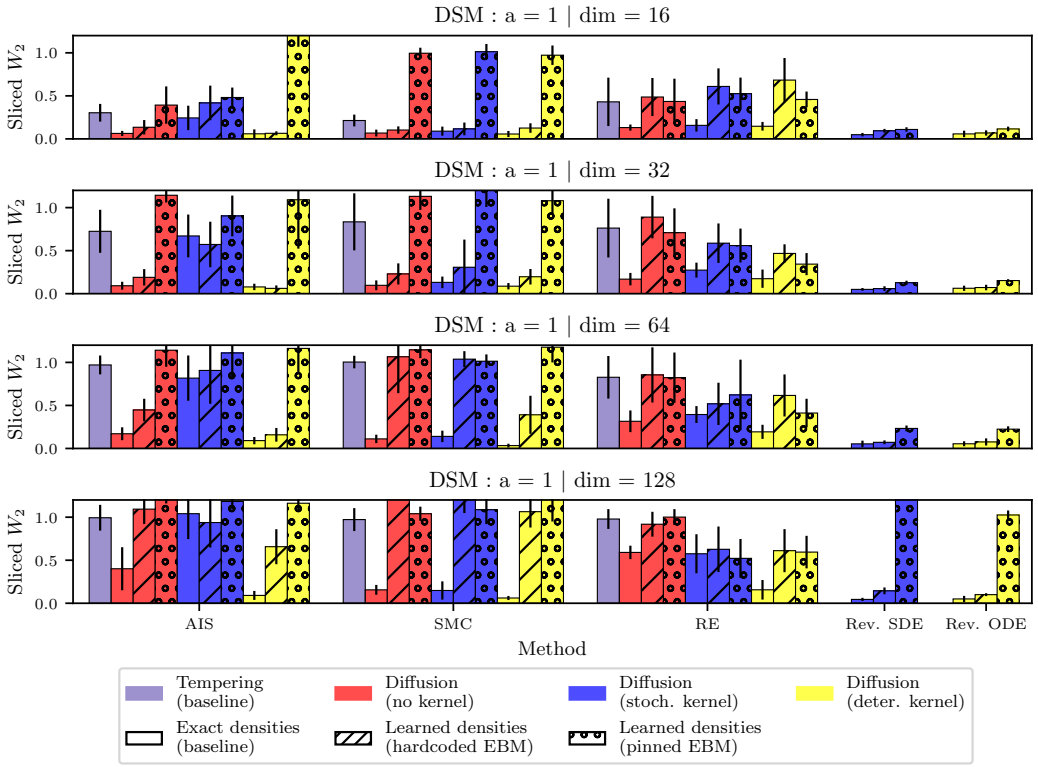


Figure E.11: **Diffusion-based aMC-BG results via sliced Wasserstein distance in realistic setting (B) with DSM objective**, when targeting *TwoModes* distribution with $a = 1$, for all dimensional settings. This is complementary to Figure 6.4.

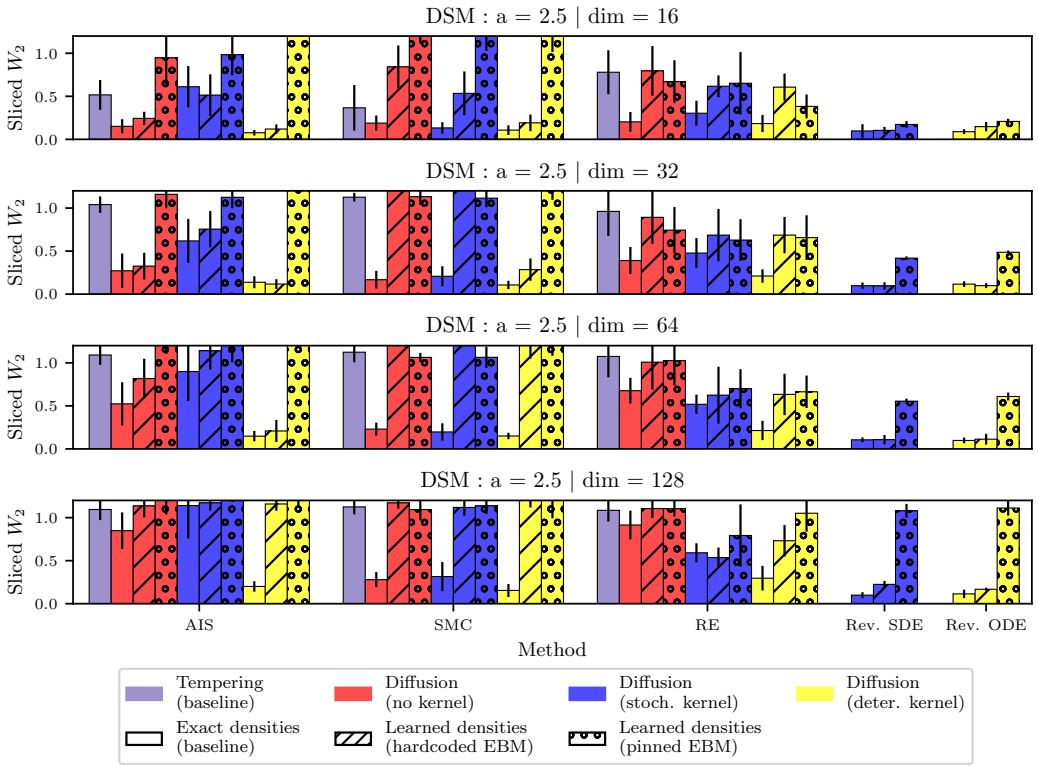


Figure E.12: **Diffusion-based aMC-BG results via sliced Wasserstein distance in realistic setting (B) with DSM objective**, when targeting *TwoModes* distribution with $a = 2.5$, for all dimensional settings. This is complementary to Figure 6.4.

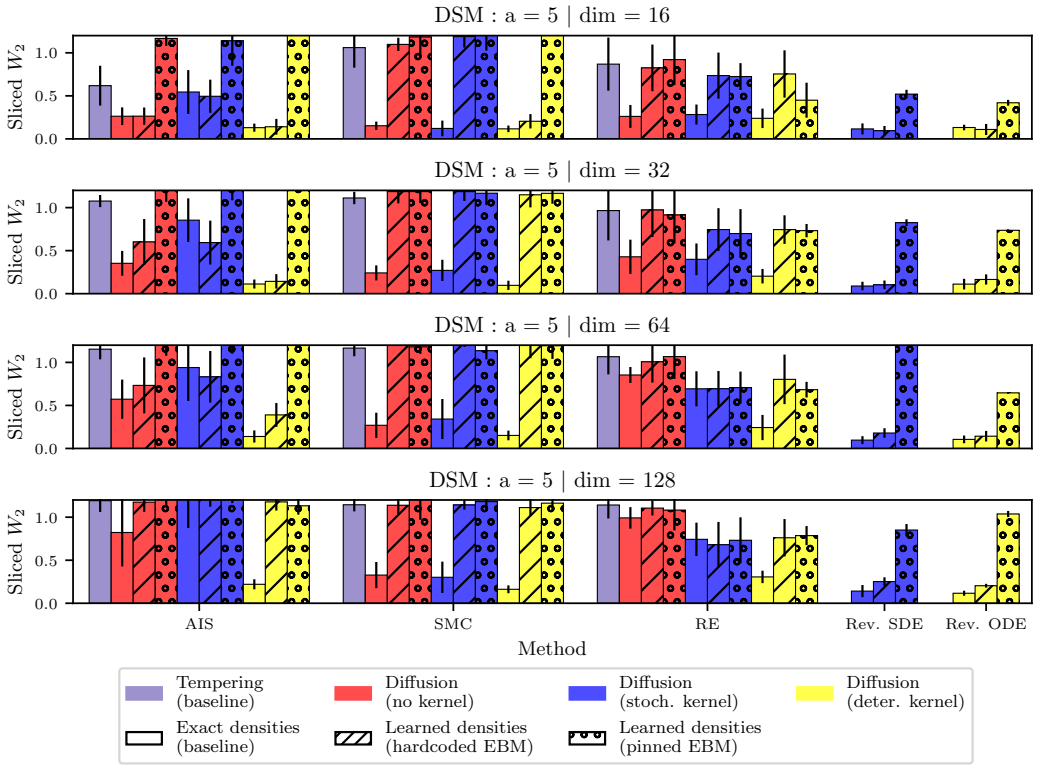


Figure E.13: **Diffusion-based aMC-BG results via sliced Wasserstein distance in realistic setting (B) with DSM objective**, when targeting *TwoModes* distribution with $a = 5$, for all dimensional settings. This is complementary to Figure 6.4.

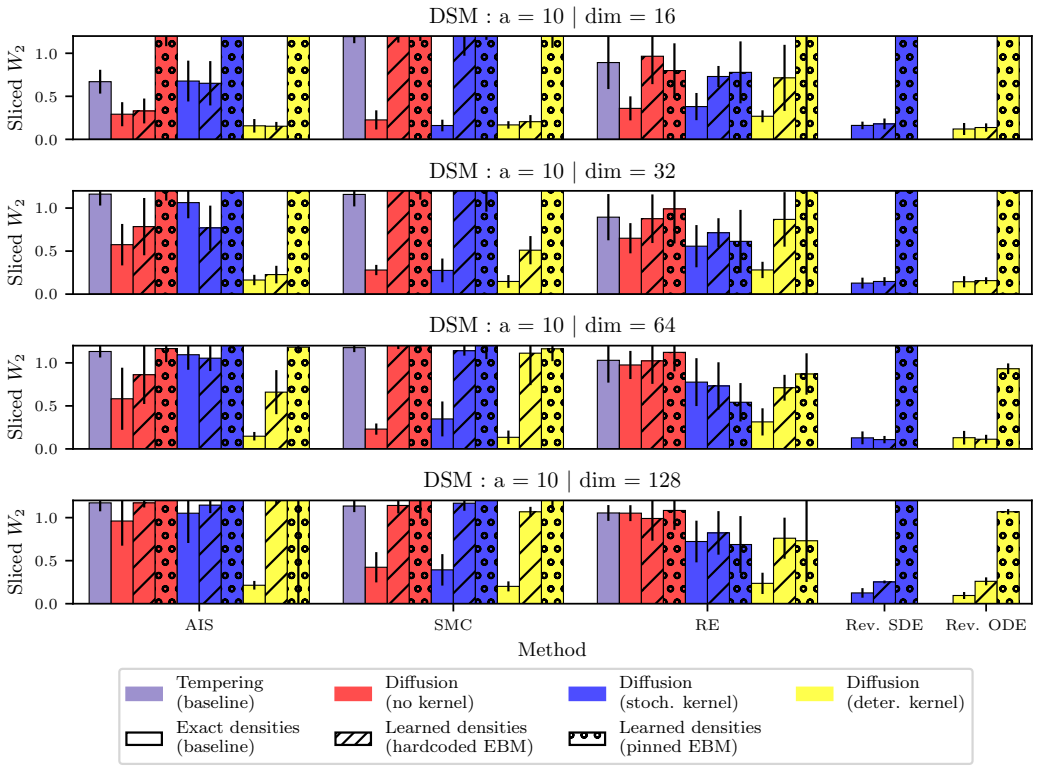


Figure E.14: **Diffusion-based aMC-BG results via sliced Wasserstein distance in realistic setting (B) with DSM objective**, when targeting *TwoModes* distribution with $a = 10$, for all dimensional settings. This is complementary to Figure 6.4.

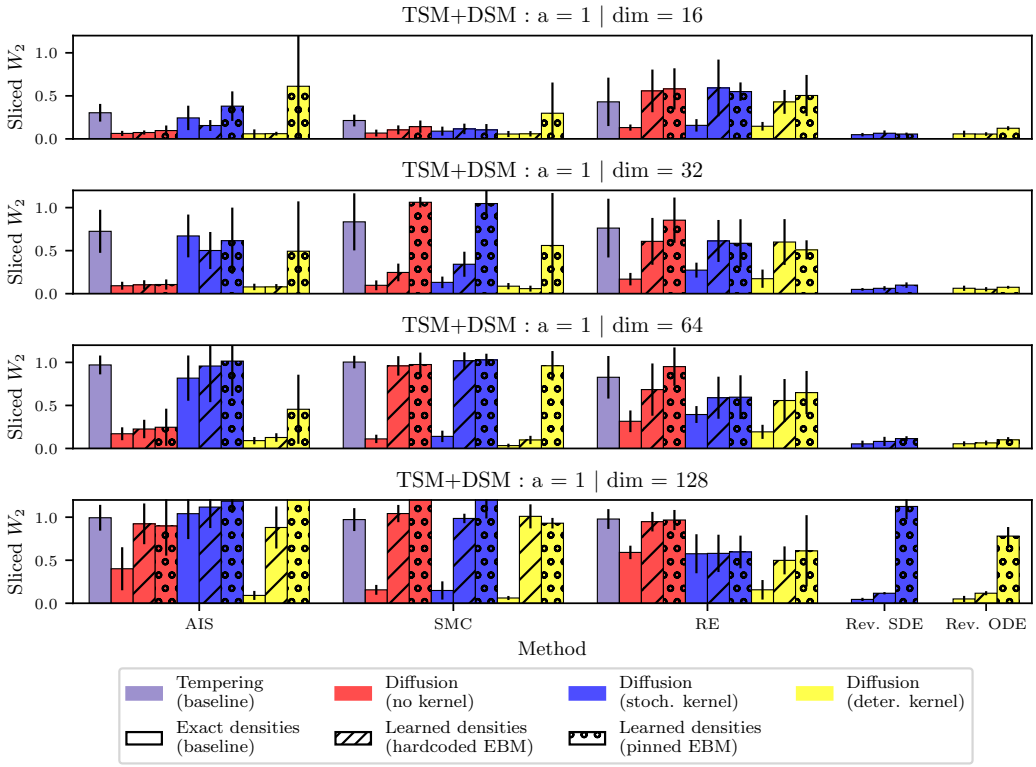


Figure E.15: **Diffusion-based aMC-BG results via sliced Wasserstein distance in realistic setting (B) with TSM+DSM objective**, when targeting *TwoModes* distribution with $a = 1$, for all dimensional settings. This is complementary to Figure 6.4.

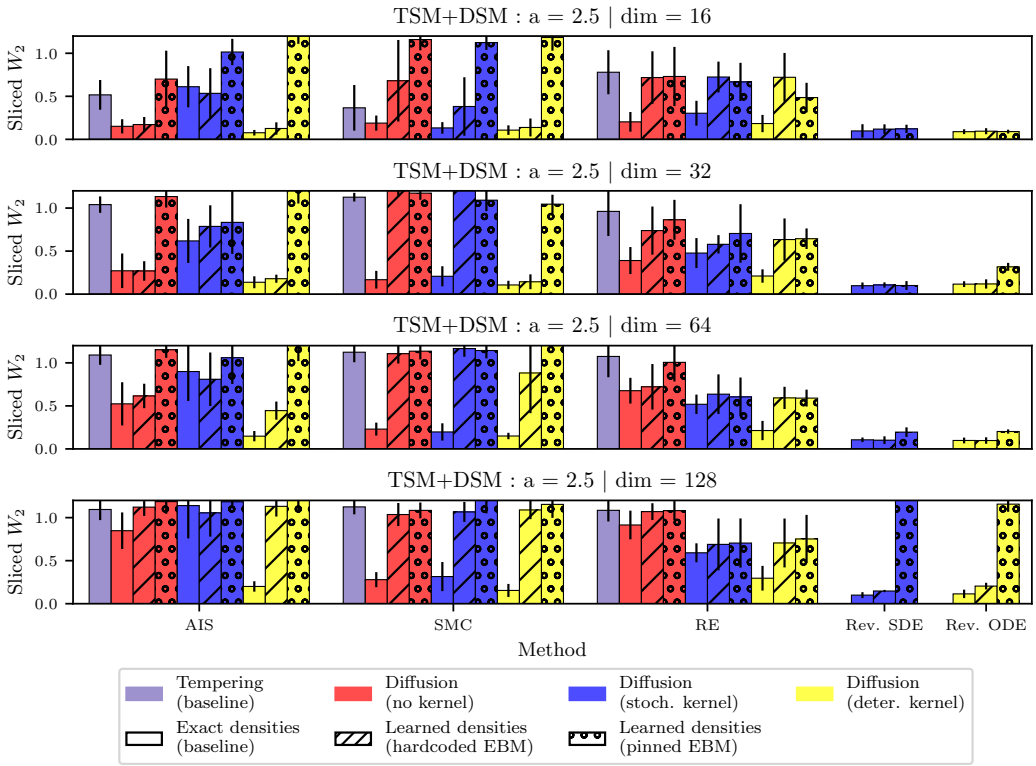


Figure E.16: **Diffusion-based aMC-BG results via sliced Wasserstein distance in realistic setting (B) with TSM+DSM objective**, when targeting *TwoModes* distribution with $a = 2.5$, for all dimensional settings. This is complementary to Figure 6.4.

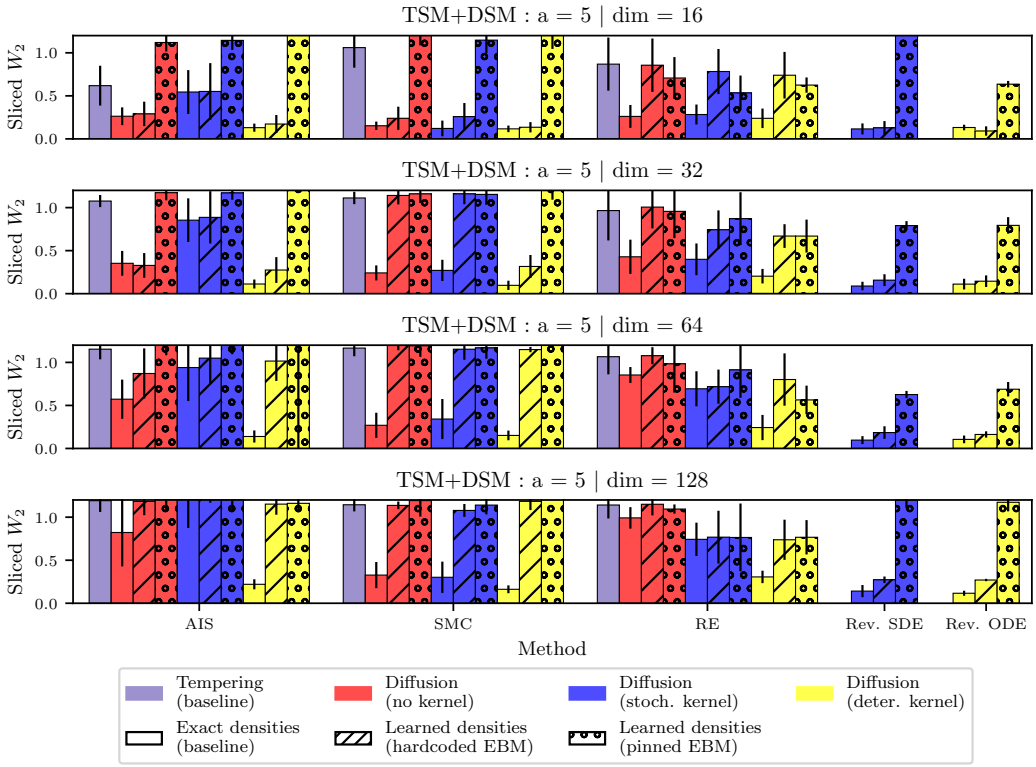


Figure E.17: **Diffusion-based aMC-BG results via sliced Wasserstein distance in realistic setting (B) with TSM+DSM objective**, when targeting *TwoModes* distribution with $a = 5$, for all dimensional settings. This is complementary to Figure 6.4.

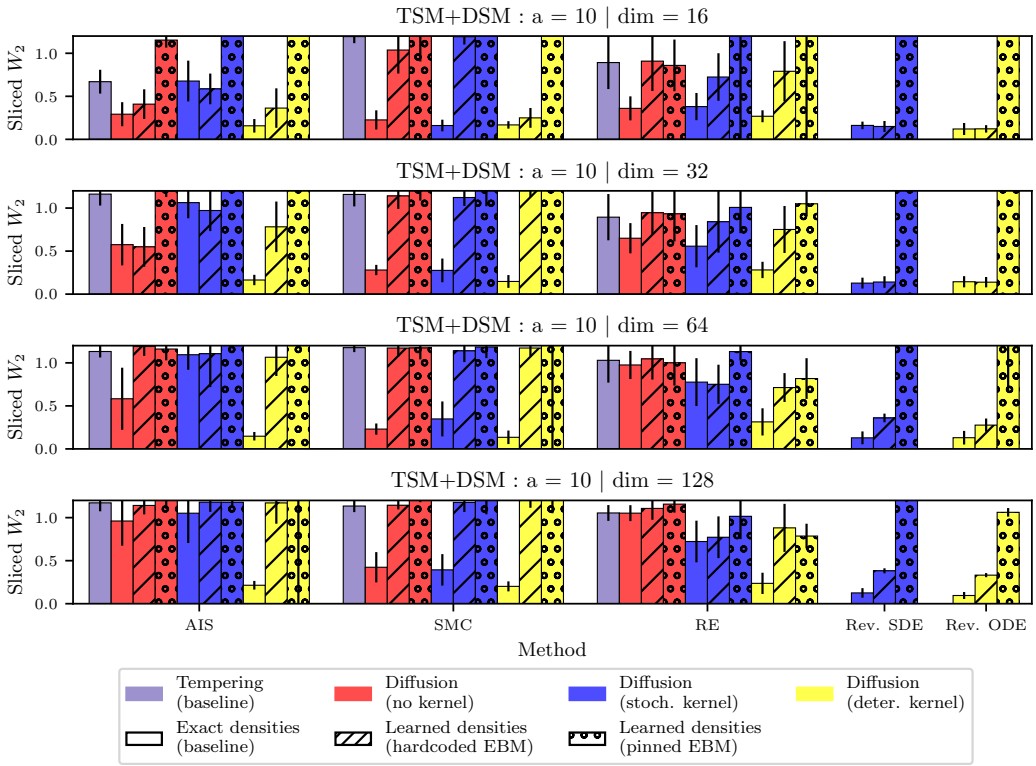


Figure E.18: **Diffusion-based aMC-BG results via sliced Wasserstein distance in realistic setting (B) with TSM+DSM objective**, when targeting *TwoModes* distribution with $a = 10$, for all dimensional settings. This is complementary to Figure 6.4.

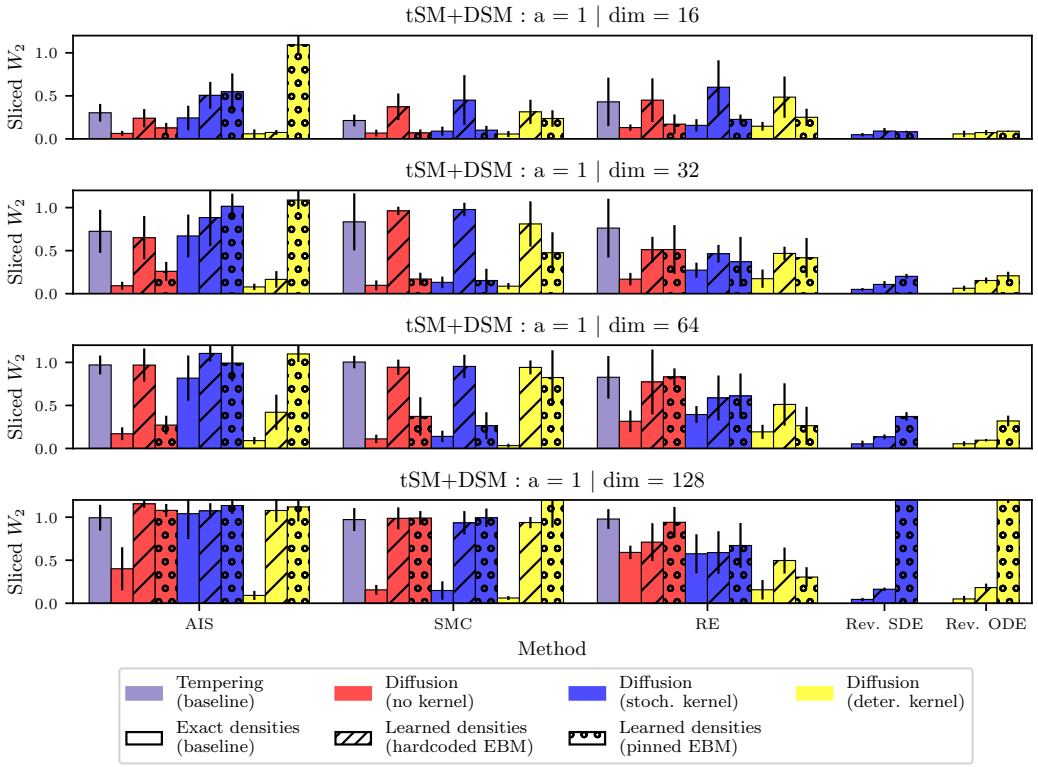


Figure E.19: **Diffusion-based aMC-BG results via sliced Wasserstein distance in realistic setting (B) with tSM+DSM objective**, when targeting *TwoModes* distribution with $a = 1$, for all dimensional settings. This is complementary to Figure 6.4.

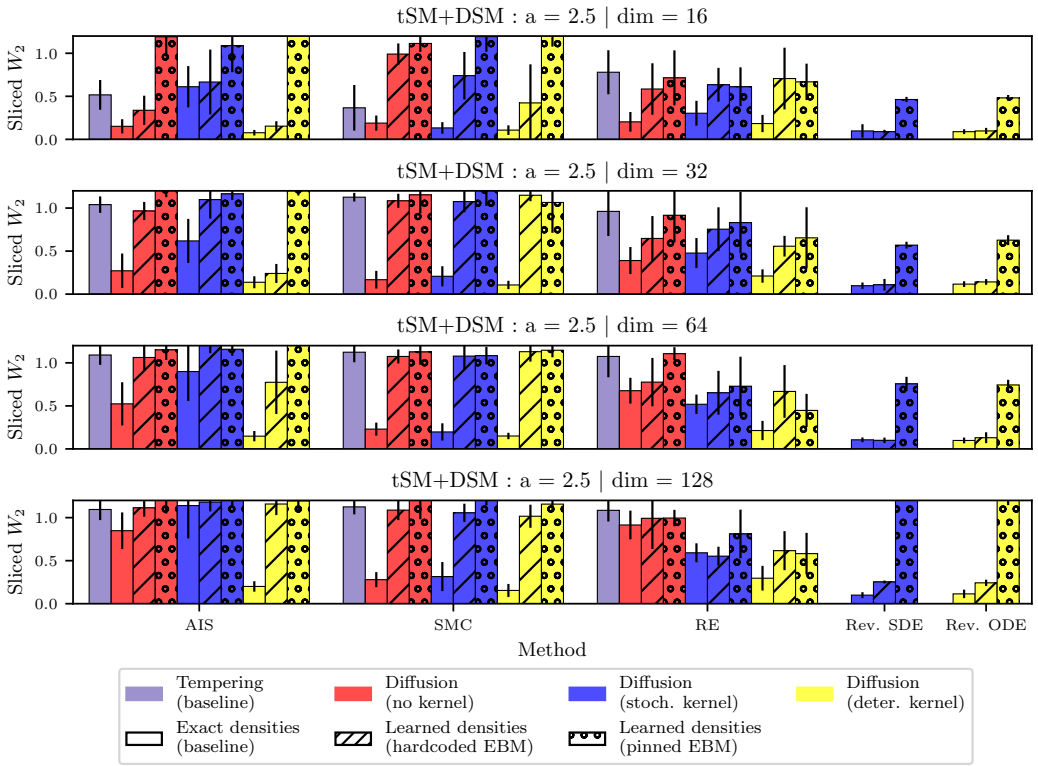


Figure E.20: **Diffusion-based aMC-BG results via sliced Wasserstein distance in realistic setting (B) with tSM+DSM objective**, when targeting *TwoModes* distribution with $a = 2.5$, for all dimensional settings. This is complementary to Figure 6.4.

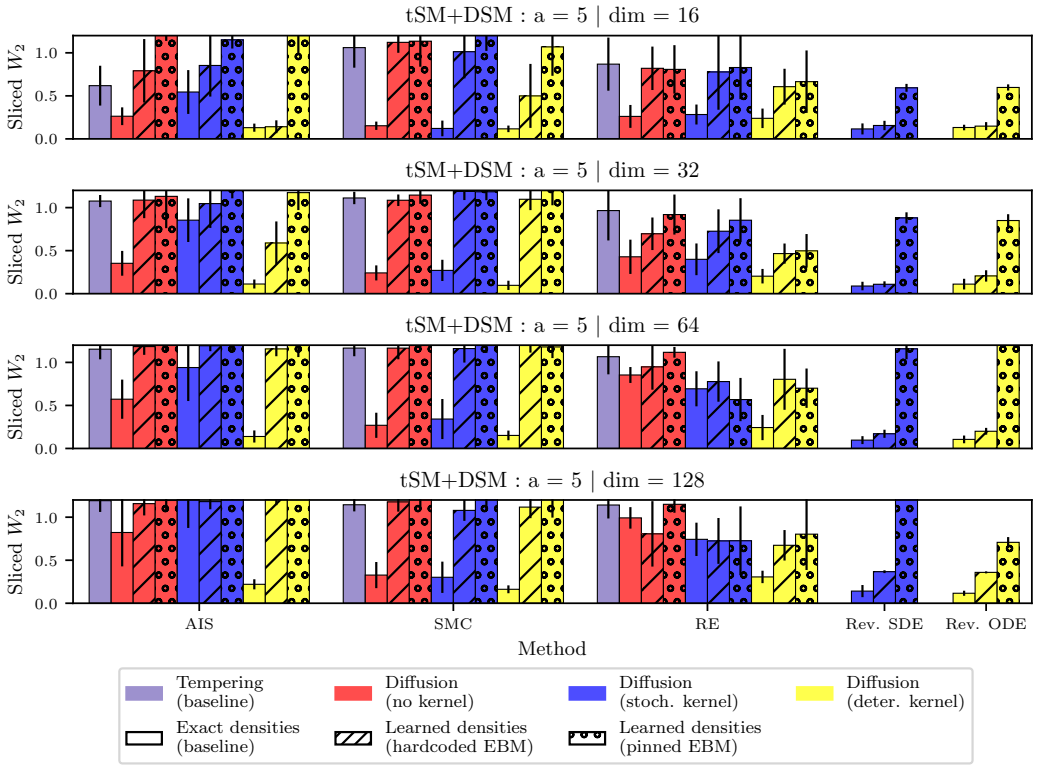


Figure E.21: **Diffusion-based aMC-BG results via sliced Wasserstein distance in realistic setting (B) with tSM+DSM objective**, when targeting *TwoModes* distribution with $a = 5$, for all dimensional settings. This is complementary to Figure 6.4.

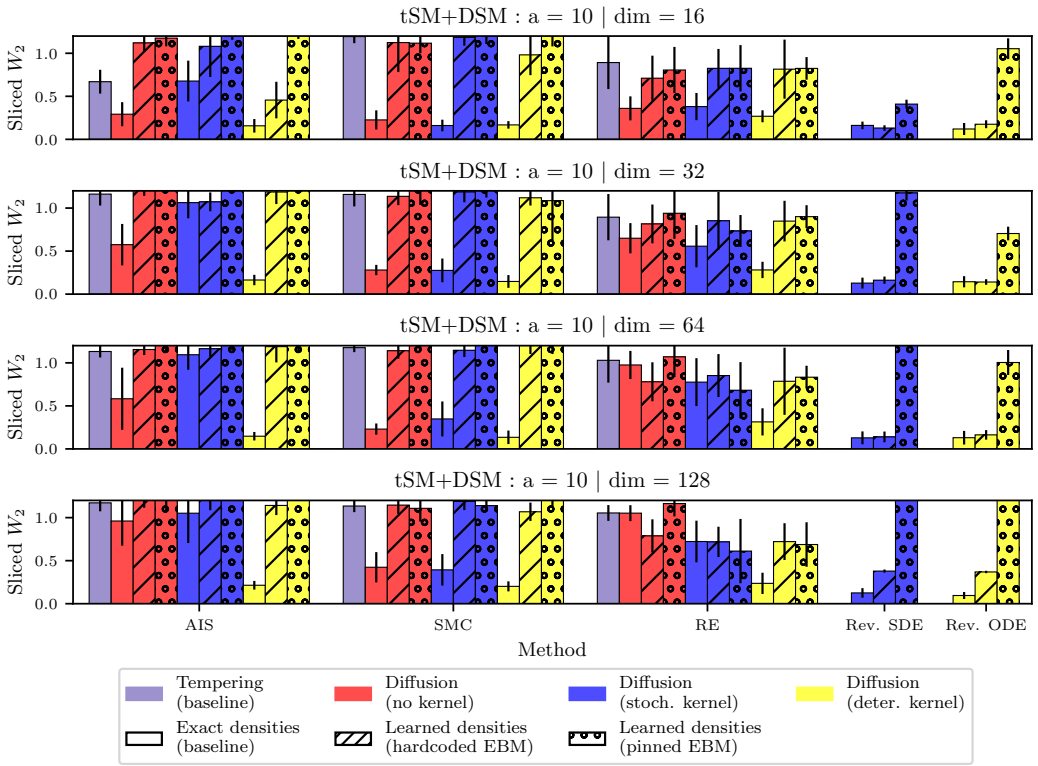


Figure E.22: **Diffusion-based aMC-BG results via sliced Wasserstein distance in realistic setting (B) with tSM+DSM objective**, when targeting *TwoModes* distribution with $a = 10$, for all dimensional settings. This is complementary to Figure 6.4.

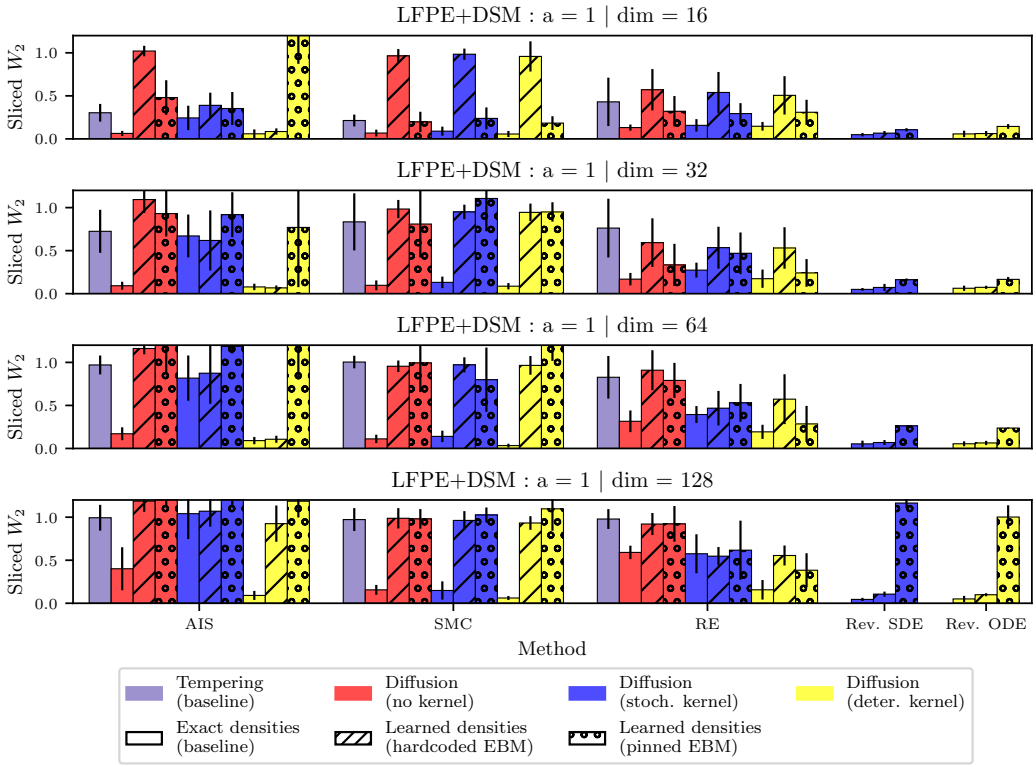


Figure E.23: **Diffusion-based aMC-BG results via sliced Wasserstein distance in realistic setting (B) with LFPE+DSM objective**, when targeting *TwoModes* distribution with $a = 1$, for all dimensional settings. This is complementary to Figure 6.4.

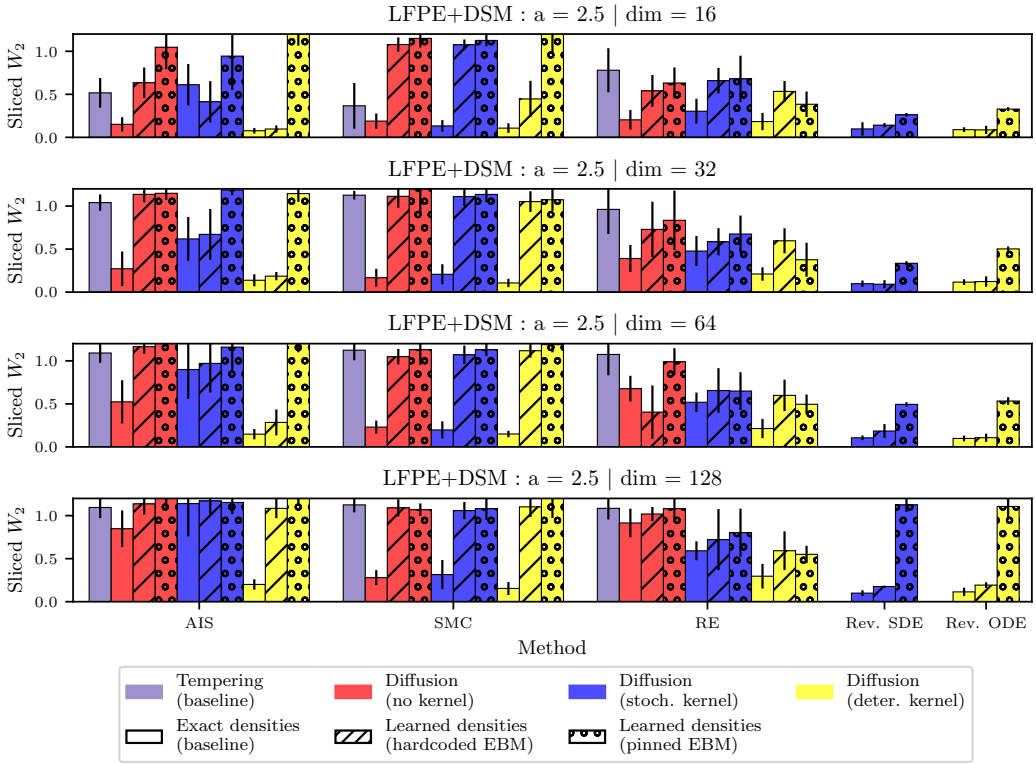


Figure E.24: **Diffusion-based aMC-BG results via sliced Wasserstein distance in realistic setting (B) with LFPE+DSM objective**, when targeting *TwoModes* distribution with $a = 2.5$, for all dimensional settings. This is complementary to Figure 6.4.

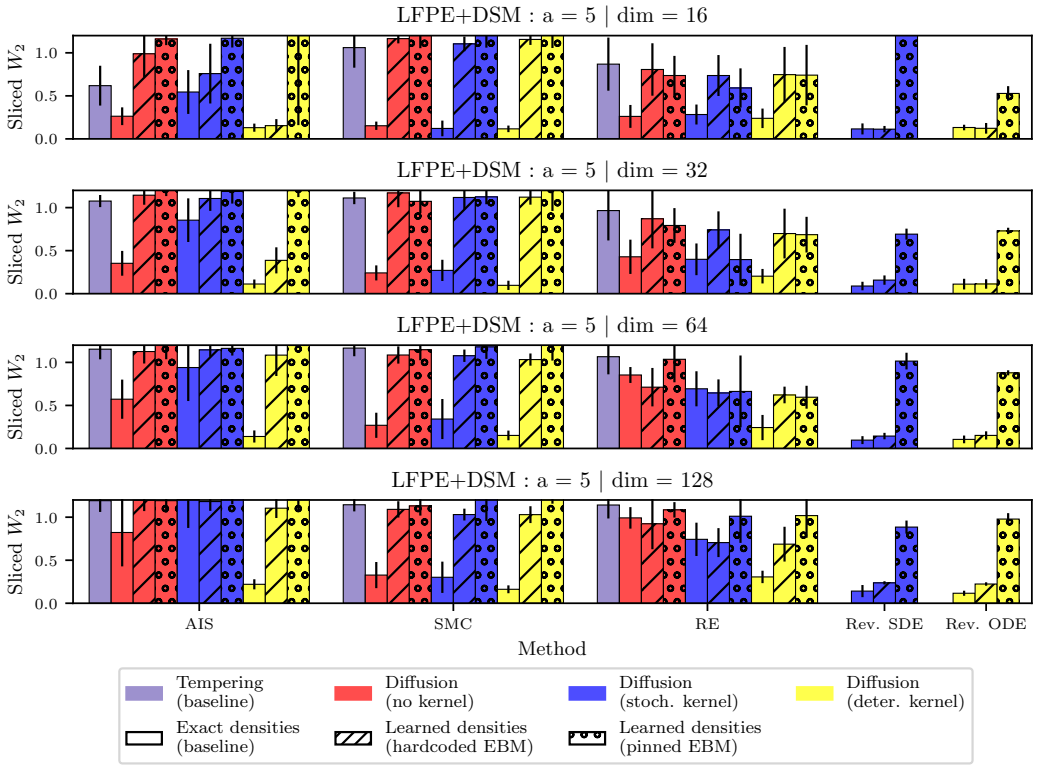


Figure E.25: **Diffusion-based aMC-BG results via sliced Wasserstein distance in realistic setting (B) with LFPE+DSM objective**, when targeting *TwoModes* distribution with $a = 5$, for all dimensional settings. This is complementary to Figure 6.4.

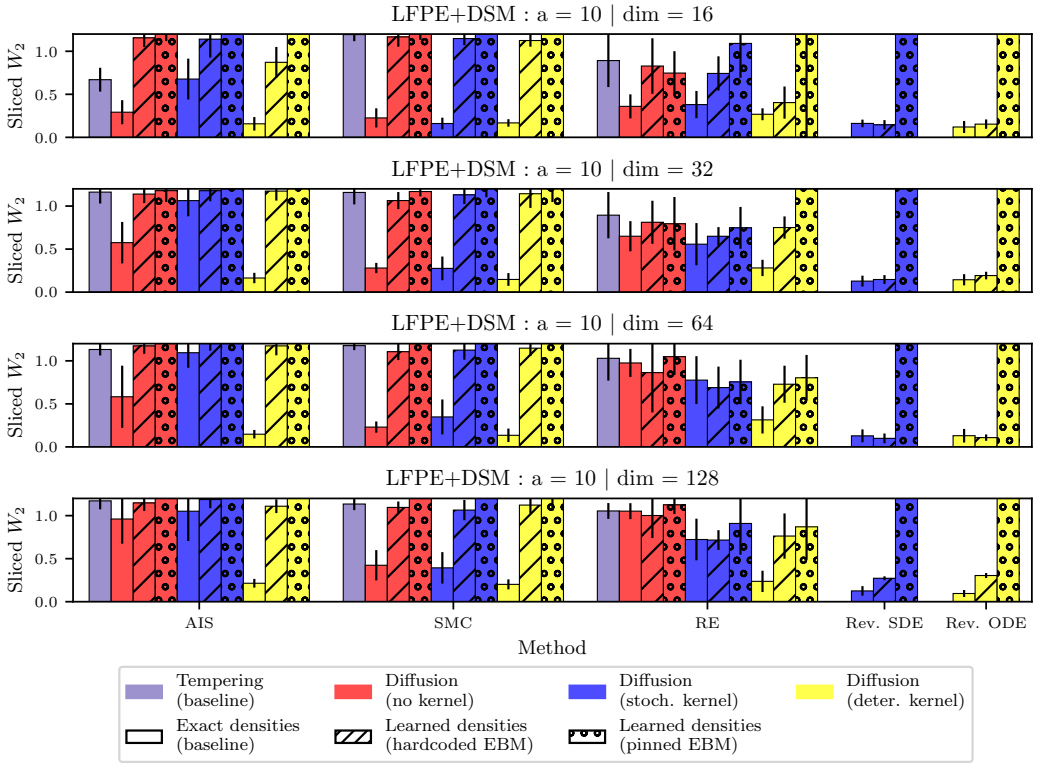


Figure E.26: **Diffusion-based aMC-BG results via sliced Wasserstein distance in realistic setting (B) with LFPE+DSM objective**, when targeting *TwoModes* distribution with $a = 10$, for all dimensional settings. This is complementary to Figure 6.4.

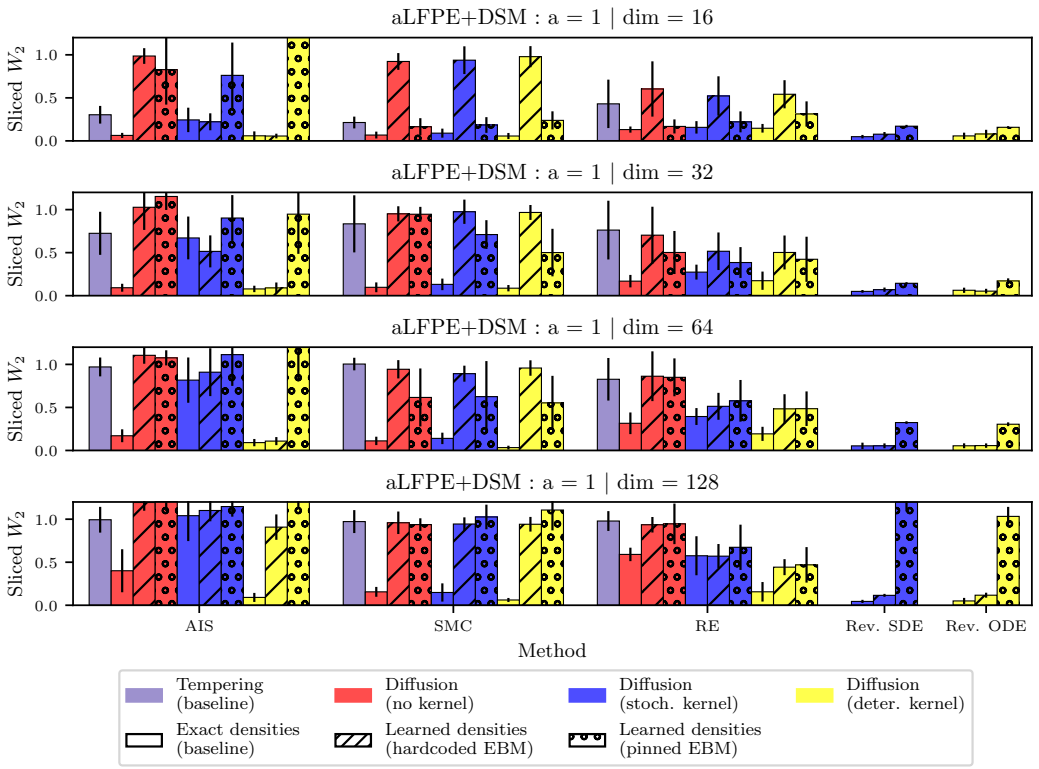


Figure E.27: **Diffusion-based aMC-BG results via sliced Wasserstein distance in realistic setting (B) with aLFPE+DSM objective**, when targeting *TwoModes* distribution with $a = 1$, for all dimensional settings. This is complementary to Figure 6.4.

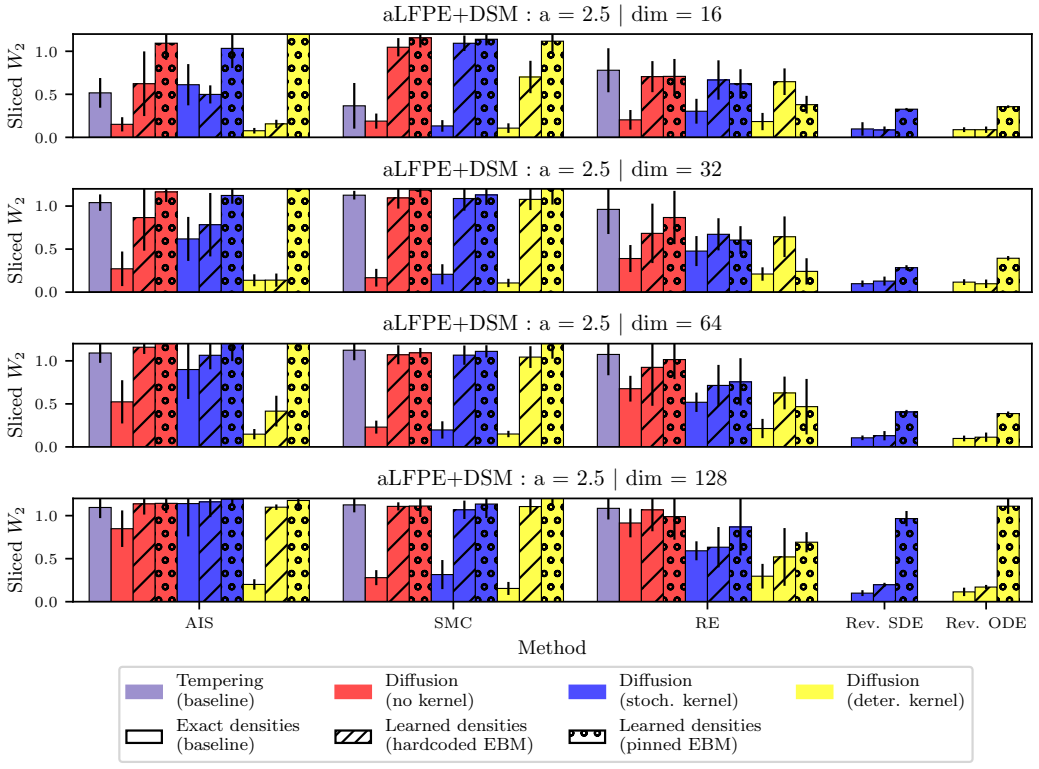


Figure E.28: **Diffusion-based aMC-BG results via sliced Wasserstein distance in realistic setting (B) with aLFPE+DSM objective**, when targeting *TwoModes* distribution with $a = 2.5$, for all dimensional settings. This is complementary to Figure 6.4.

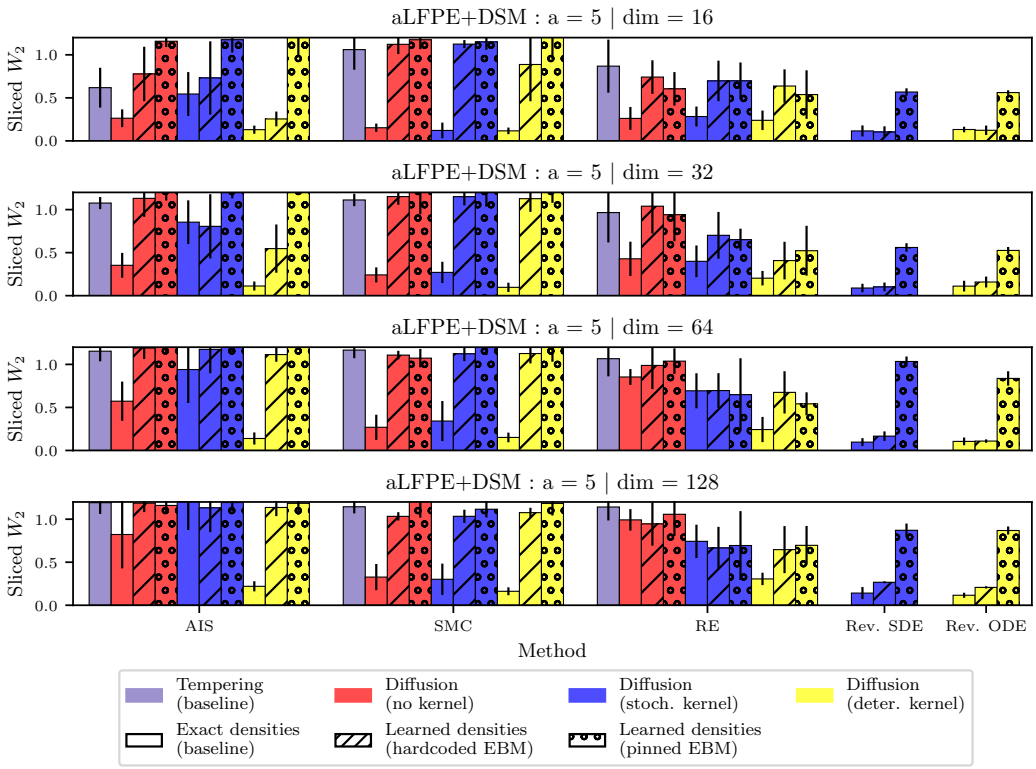


Figure E.29: **Diffusion-based aMC-BG results via sliced Wasserstein distance in realistic setting (B) with aLFPE+DSM objective**, when targeting *TwoModes* distribution with $a = 5$, for all dimensional settings. This is complementary to Figure 6.4.

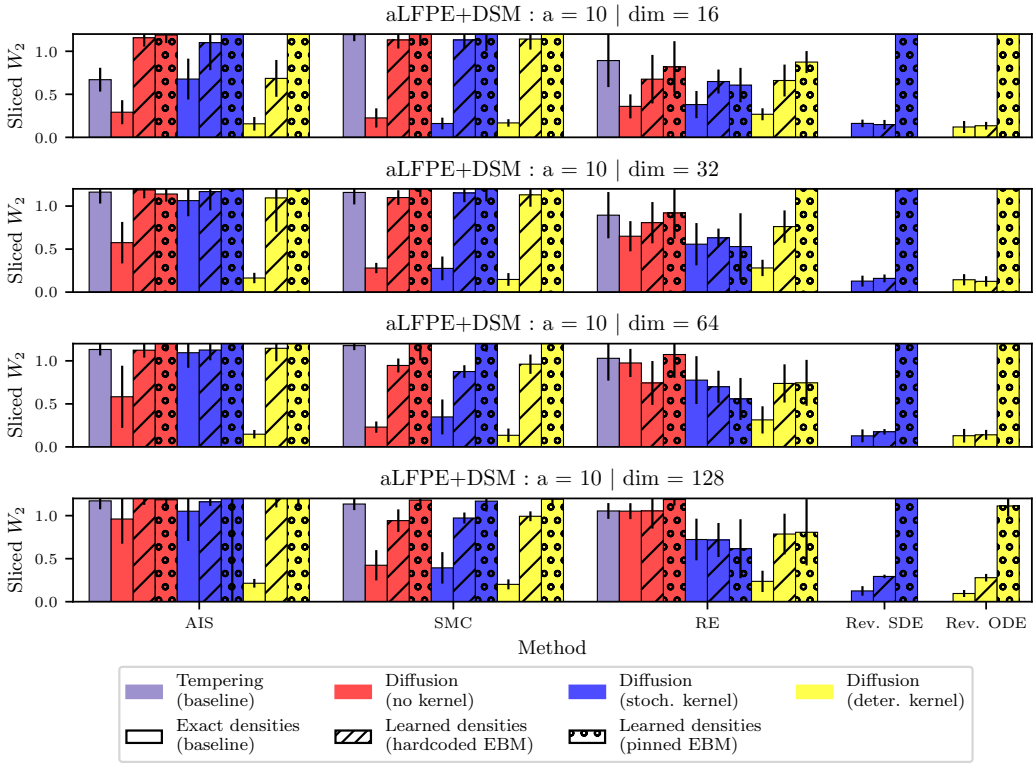


Figure E.30: **Diffusion-based aMC-BG results via sliced Wasserstein distance in realistic setting (B) with aLFPE+DSM objective**, when targeting *TwoModes* distribution with $a = 10$, for all dimensional settings. This is complementary to Figure 6.4.

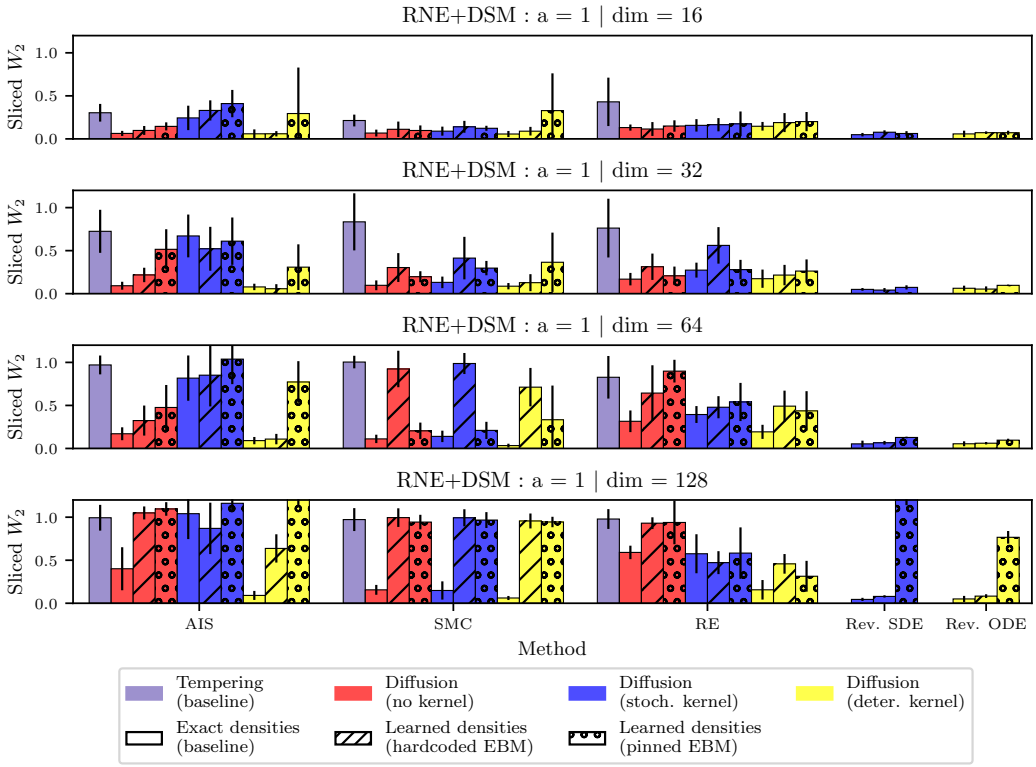


Figure E.31: **Diffusion-based aMC-BG results via sliced Wasserstein distance in realistic setting (B) with RNE+DSM objective**, when targeting *TwoModes* distribution with $a = 1$, for all dimensional settings. This is complementary to Figure 6.4.

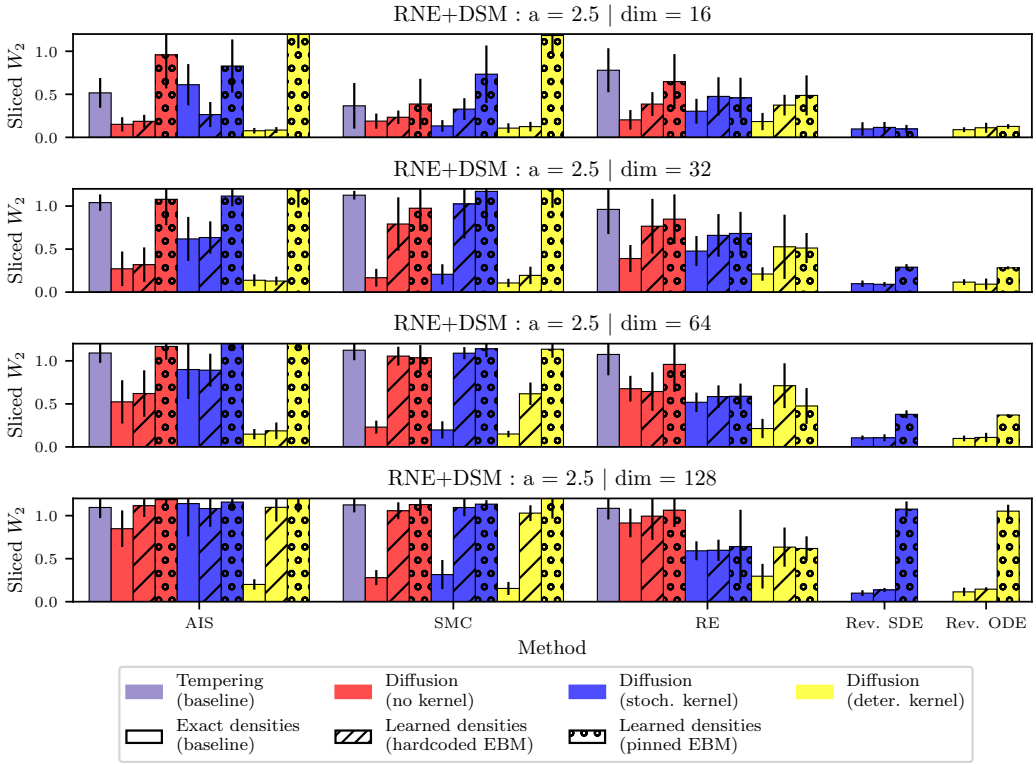


Figure E.32: **Diffusion-based aMC-BG results via sliced Wasserstein distance in realistic setting (B) with RNE+DSM objective**, when targeting *TwoModes* distribution with $a = 2.5$, for all dimensional settings. This is complementary to Figure 6.4.

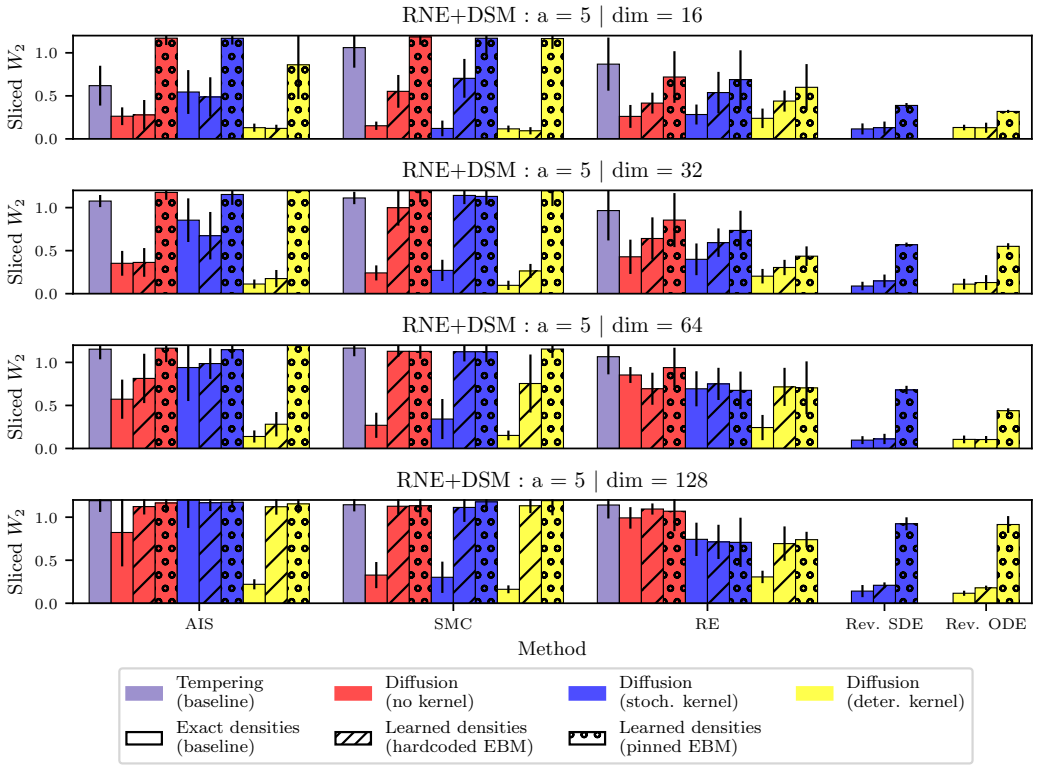


Figure E.33: **Diffusion-based aMC-BG results via sliced Wasserstein distance in realistic setting (B) with RNE+DSM objective**, when targeting *TwoModes* distribution with $a = 5$, for all dimensional settings. This is complementary to Figure 6.4.

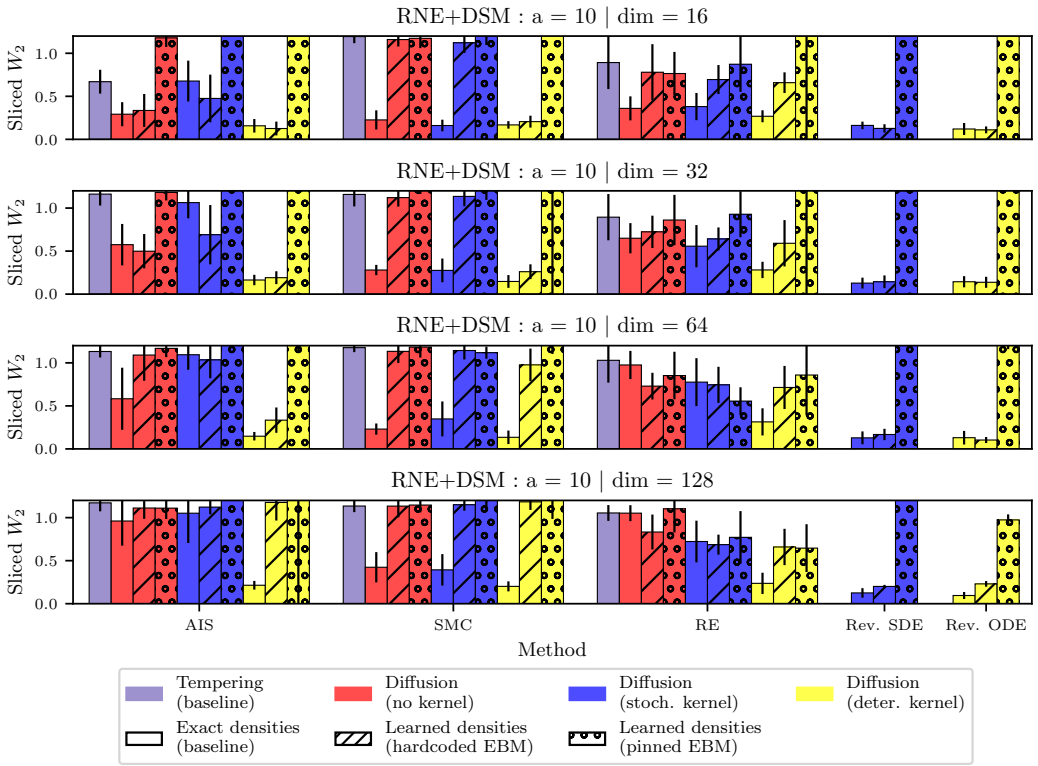


Figure E.34: **Diffusion-based aMC-BG results via sliced Wasserstein distance in realistic setting (B) with RNE+DSM objective**, when targeting *TwoModes* distribution with $a = 10$, for all dimensional settings. This is complementary to Figure 6.4.

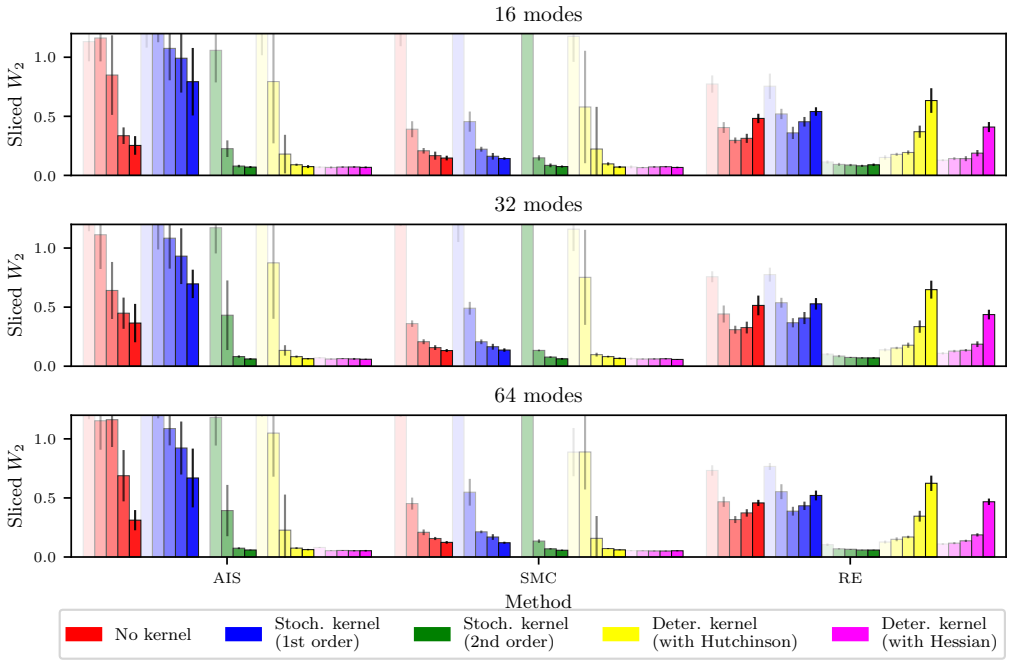


Figure E.35: **Diffusion-based aMC-BG results via sliced Wasserstein distance in idealized setting (A)**, when targeting *ManyModes* distribution in all possible settings. This is complementary to Figure 6.3.

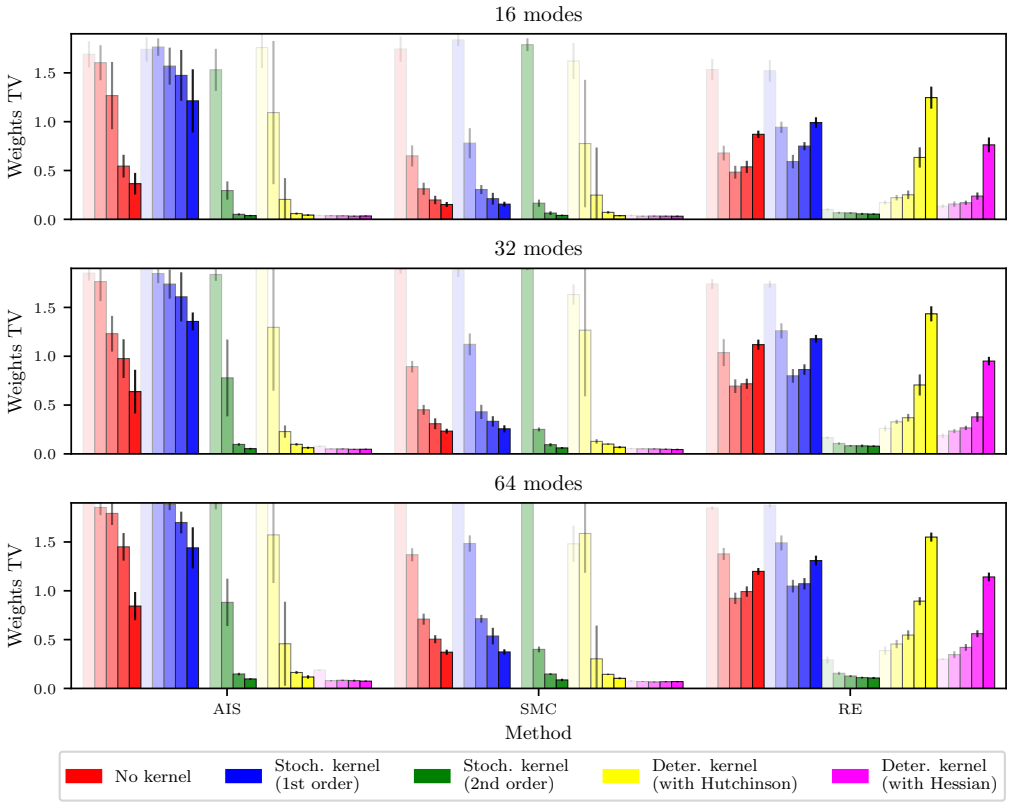


Figure E.36: **Diffusion-based aMC-BG results via weight histogram total variation distance in idealized setting (A)**, when targeting *ManyModes* distribution in all possible settings. This is complementary to Figure 6.3.

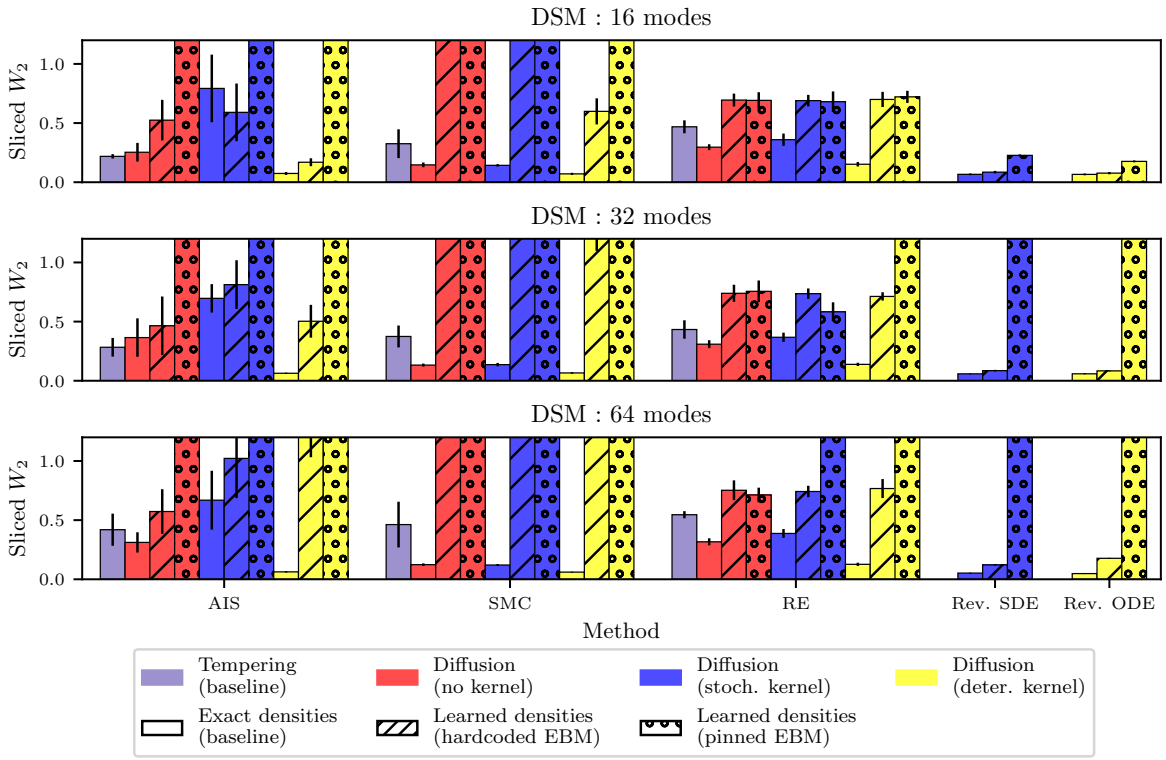


Figure E.37: **Diffusion-based aMC-BG results via sliced Wasserstein distance in realistic setting (B) with DSM objective**, when targeting *ManyModes* distribution in all possible settings. This is complementary to Figure 6.4.

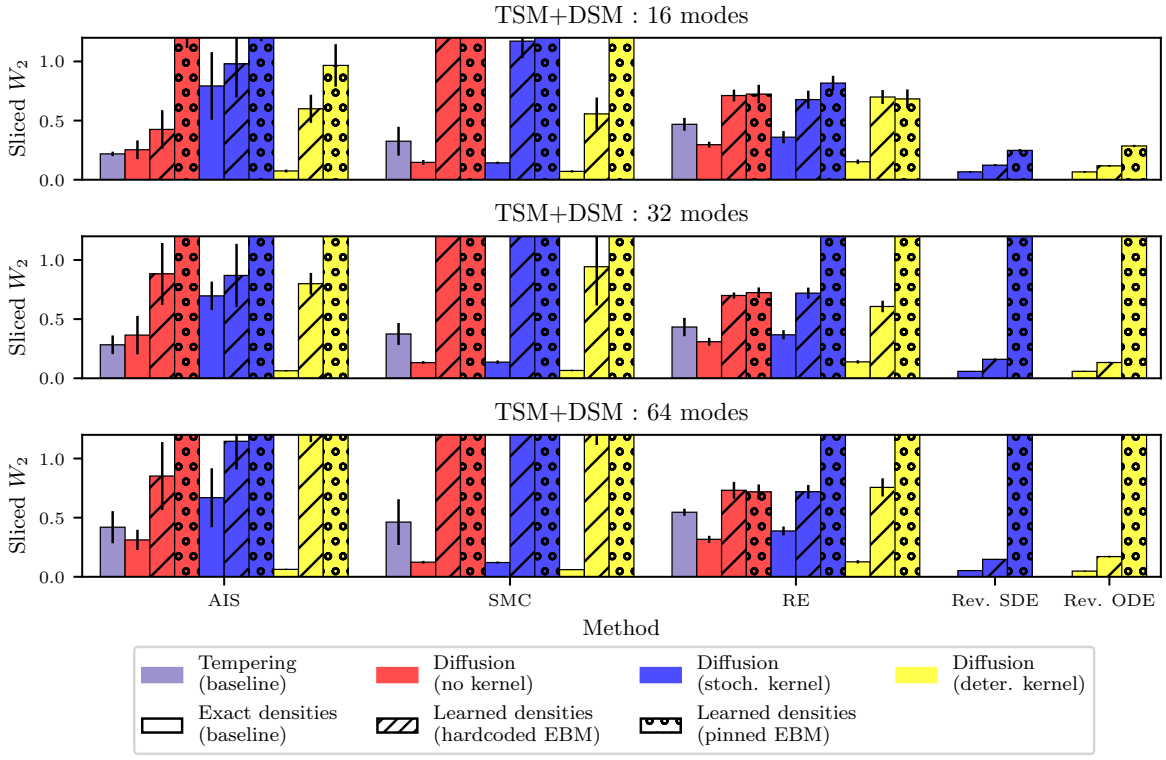


Figure E.38: **Diffusion-based aMC-BG results via sliced Wasserstein distance in realistic setting (B) with TSM+DSM objective**, when targeting *ManyModes* distribution in all possible settings. This is complementary to Figure 6.4.

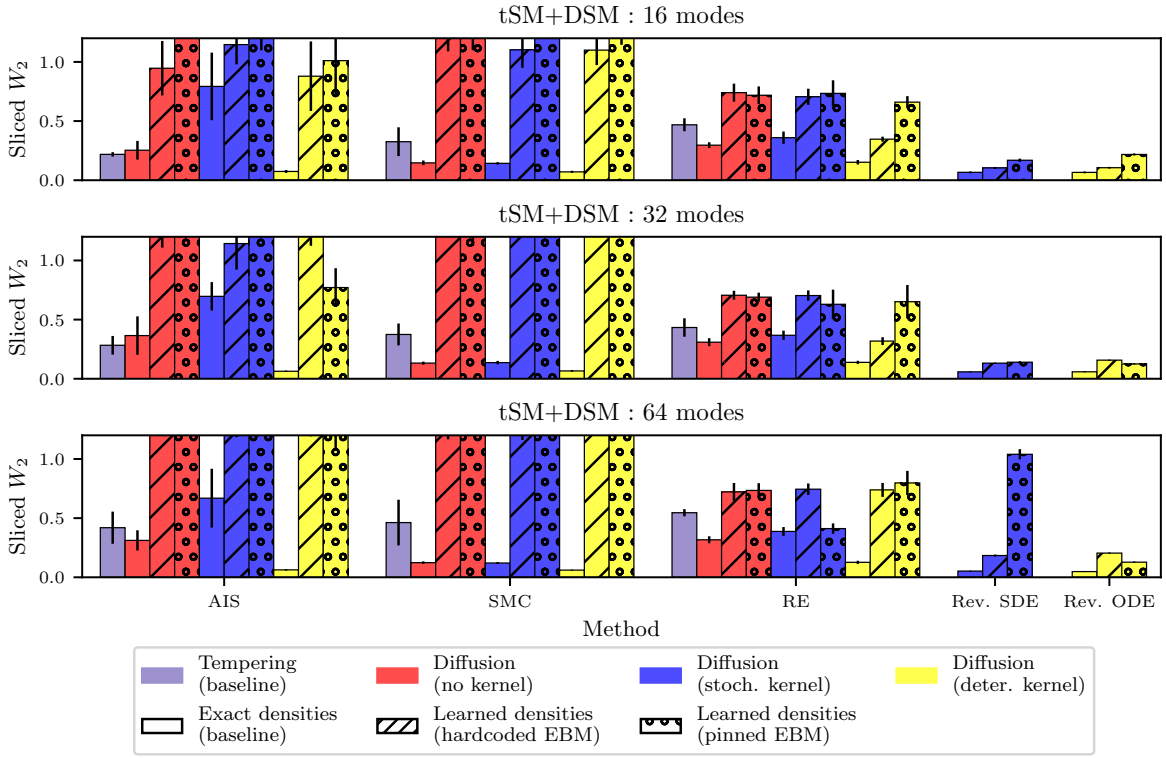


Figure E.39: **Diffusion-based aMC-BG results via sliced Wasserstein distance in realistic setting (B) with tSM+DSM objective**, when targeting *ManyModes* distribution in all possible settings. This is complementary to Figure 6.4.

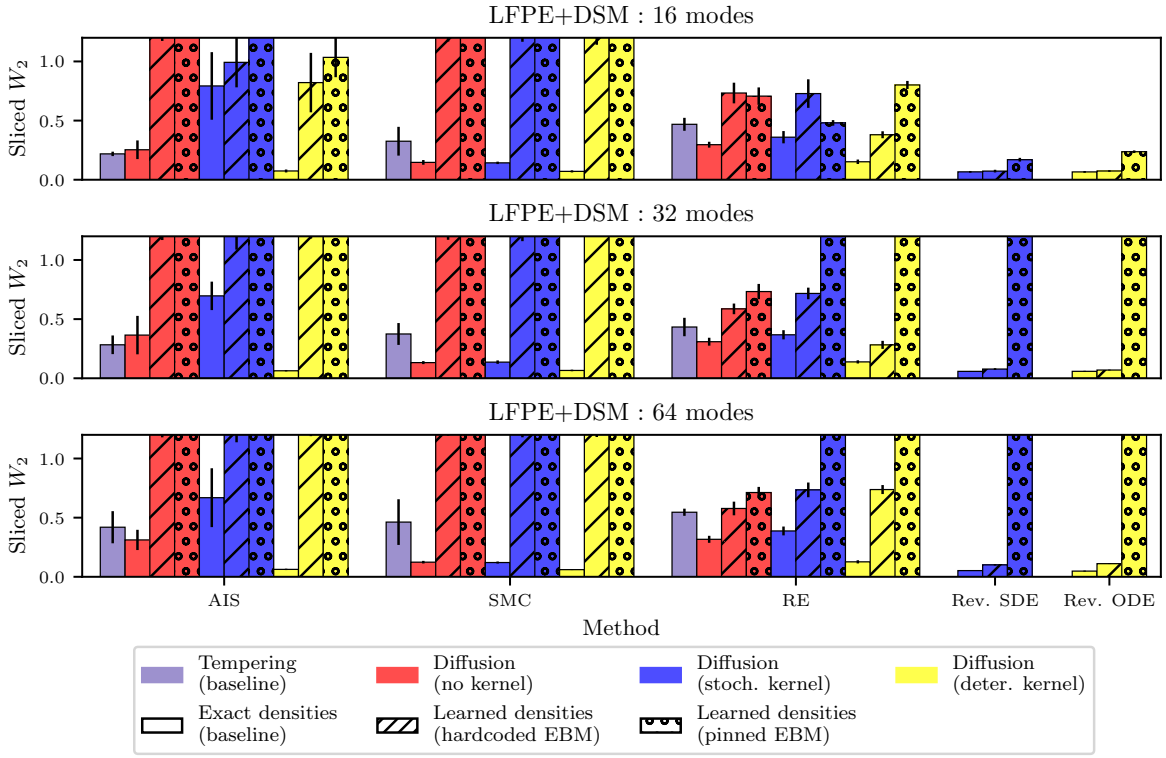


Figure E.40: **Diffusion-based aMC-BG results via sliced Wasserstein distance in realistic setting (B) with LFPE+DSM objective**, when targeting *ManyModes* distribution in all possible settings. This is complementary to Figure 6.4.

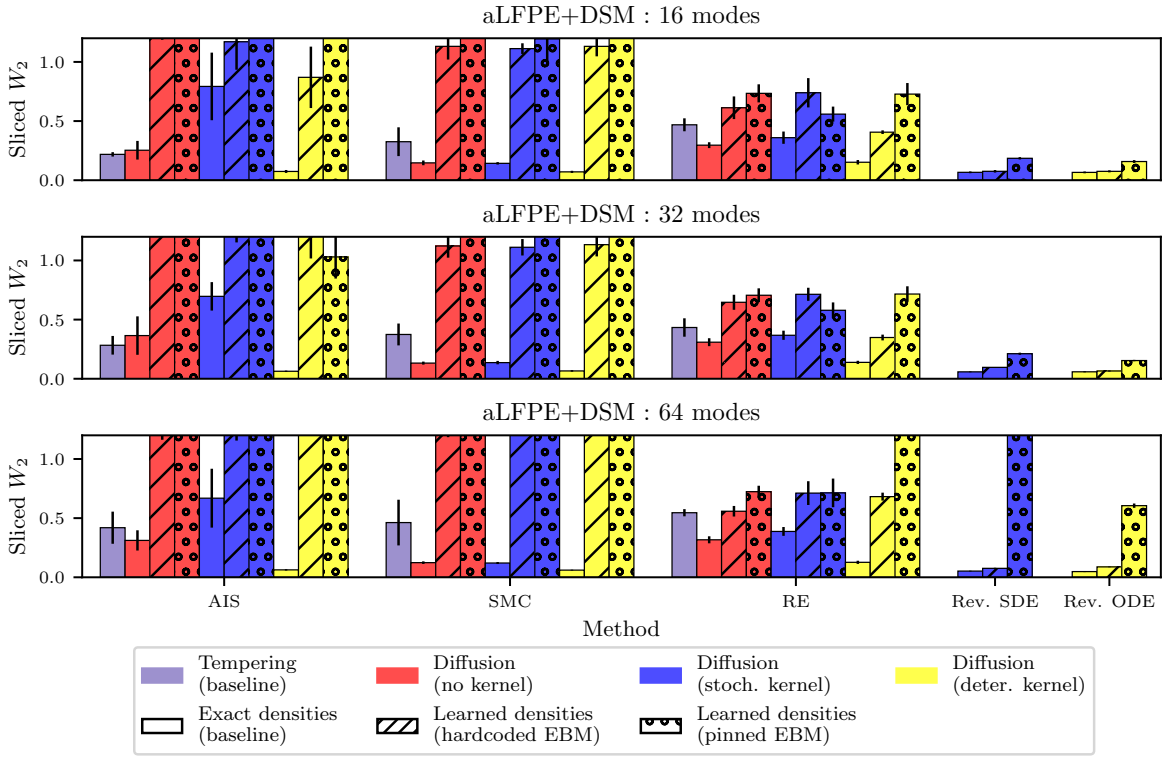


Figure E.41: **Diffusion-based aMC-BG results via sliced Wasserstein distance in realistic setting (B) with aLFPE+DSM objective**, when targeting *ManyModes* distribution in all possible settings. This is complementary to Figure 6.4.

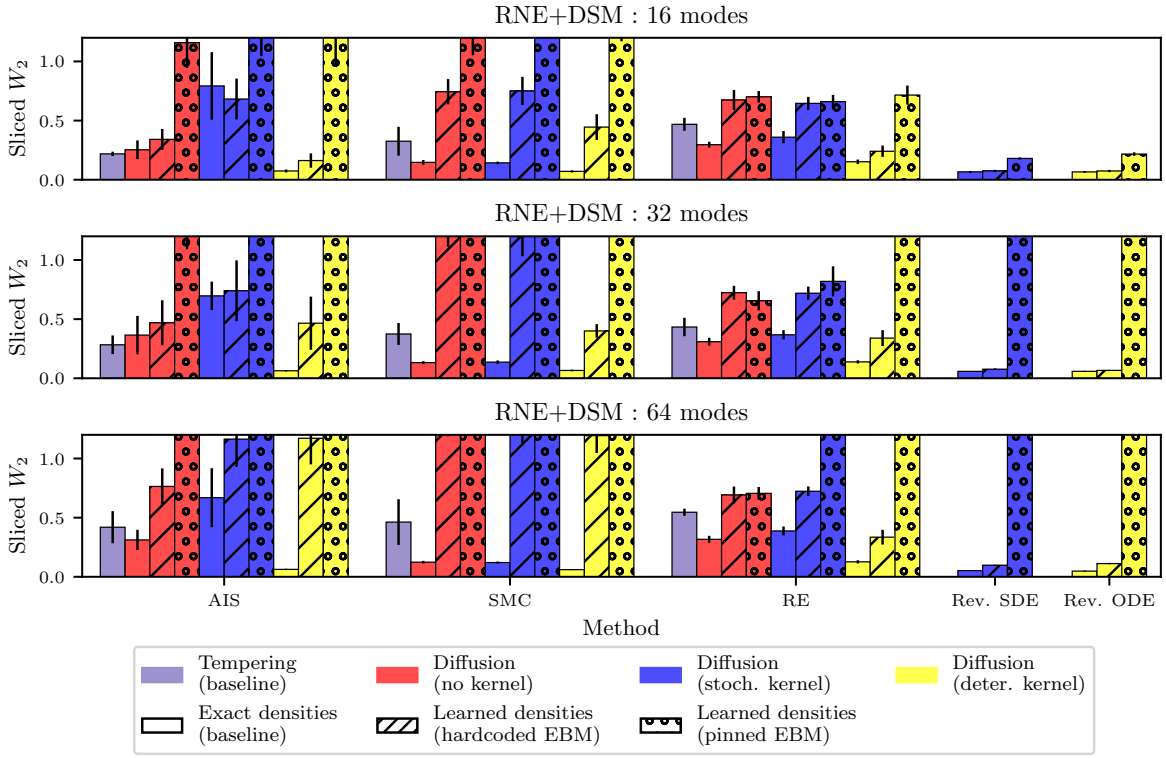


Figure E.42: **Diffusion-based aMC-BG results via sliced Wasserstein distance in realistic setting (B) with RNE+DSM objective**, when targeting *ManyModes* distribution in all possible settings. This is complementary to Figure 6.4.

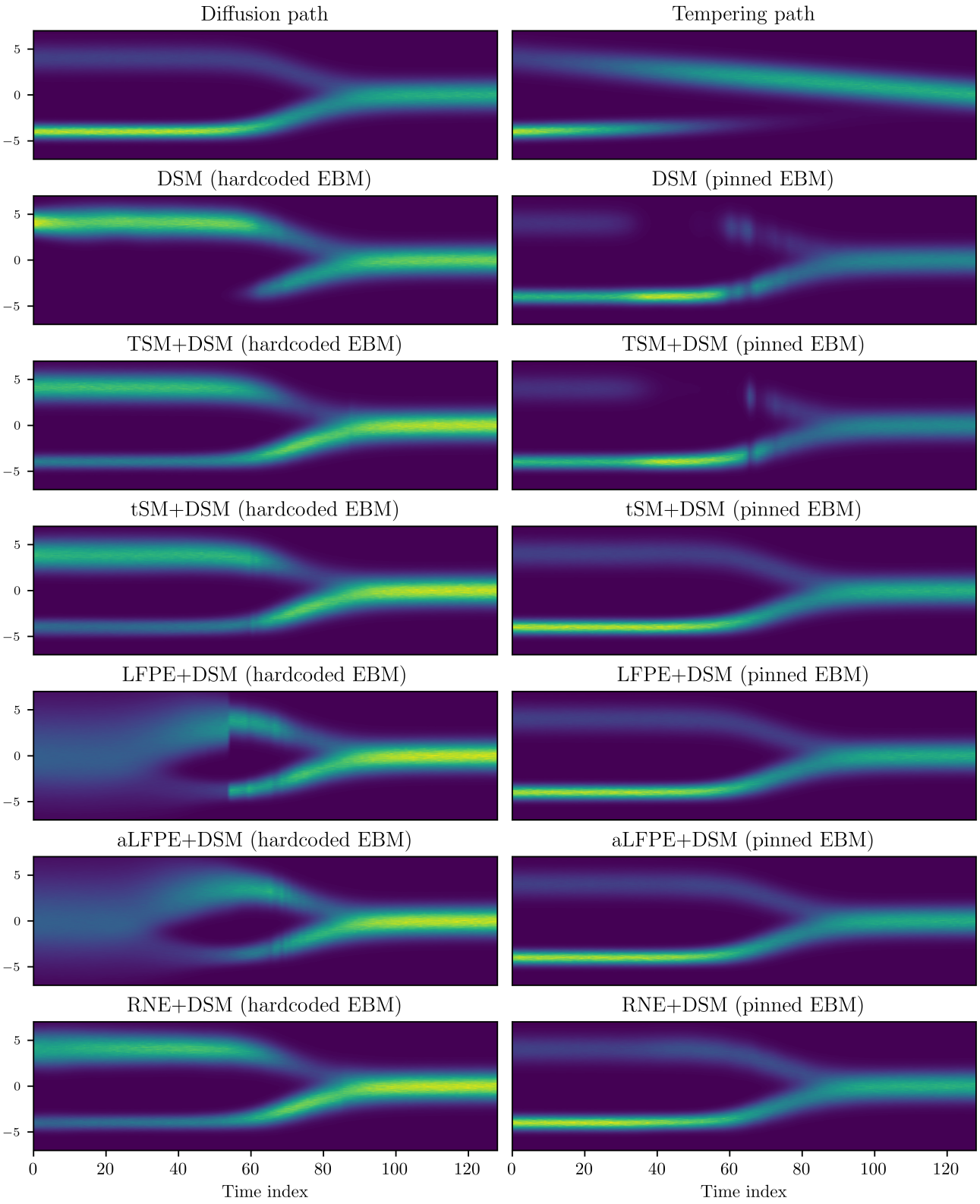


Figure E.43: **Diffusion density paths bridging π^{base} (last time index) to the same *TwoModes* target as in Figure 6.5 (first time index). (First row)** the exact diffusion path is displayed on the left, the exact tempering path on the right, **(From second to last row)** we display the learned density path when using the DSM, TSM+DSM, tSM+DSM, LFPE+DSM, aLFPE+DSM or RNE+DSM objective, with identical computational budget, **(Left)** use of Hardcoded EBM, **(Right)** use of Pinned EBM. This is complementary to Figure 6.6. Zoom in to get more details.

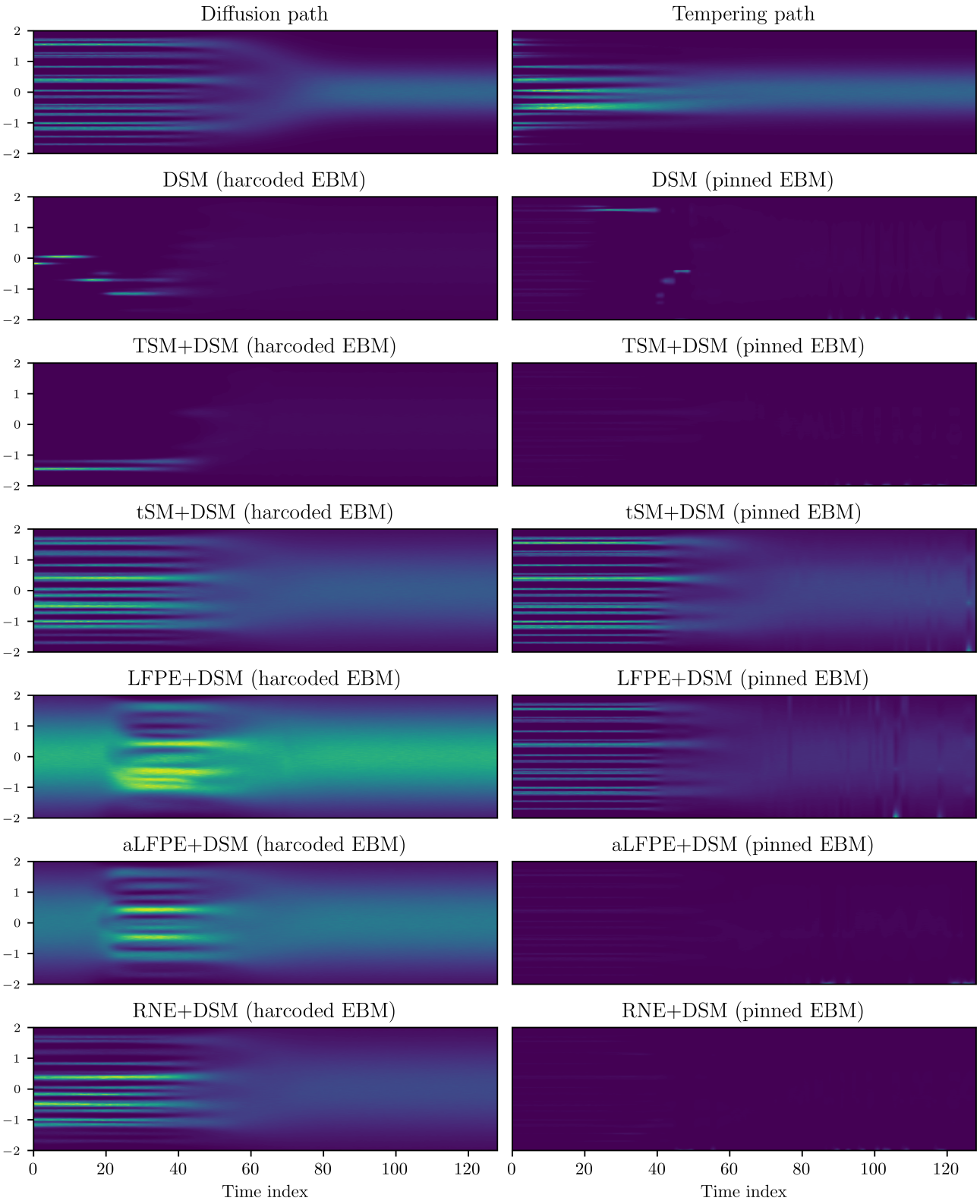


Figure E.44: **Diffusion density paths bridging π^{base} (last time index) to the same *ManyModes* target as in Figure 6.5 (first time index). (First row)** the exact diffusion path is displayed on the left, the exact tempering path on the right, **(From second to last row)** we display the learned density path when using the DSM, TSM+DSM, tSM+DSM, LFPE+DSM, aLFPE+DSM or RNE+DSM objective, with identical computational budget, **(Left)** use of Hardcoded EBM, **(Right)** use of Pinned EBM. This is complementary to Figure 6.6. Zoom in to get more details.

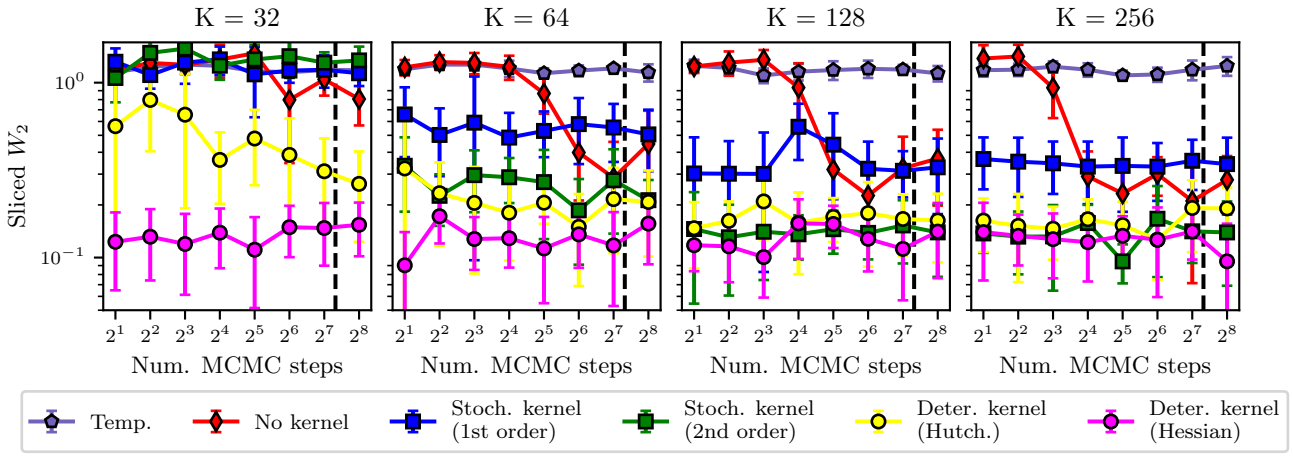


Figure E.45: **Sensitivity of SMC methods (including diffusion-based BGs) with respect to the number of local MCMC steps across different values of K.** We do not consider any MCMC warm-up procedure here. The default setting of the main experiments is represented by a dashed line.

E.4.4 Ablation studies

In this section, we investigate the sensitivity of the annealed samplers considered in this work with respect to their respective hyperparameters, with a particular focus on diffusion-based aMC-BGs featuring deterministic transitions, see Section 6.4. To keep the analysis focused and interpretable, all experiments are conducted in idealized setting **(A)** on a representative configuration, the so-called "medium case", of the *TwoModes* distribution with mode spacing $a = 5.0$ and dimension $d = 64$.

Dependence of SMC with respect to the number of local MCMC steps. Figure E.45 examines the sensitivity of all SMC-based methods to the number of local MCMC steps performed (with MALA sampler) at each intermediate global step $k \in \{0, \dots, K\}$, where the number of annealing levels K is selected in $\{32, 64, 128, 256\}$. Note that, unlike the main experiments, we do not use any MCMC warm-up here, thereby making the corresponding sampling runs much less computationally intensive on average; for comparison, we display with a dashed line the MCMC budget used in the main experiments (*i.e.*, 160 steps). We first observe that, in the case of the tempering path, increasing the number of local sampling steps does not bring better results for each value of K, thereby proving the unfavorable sampling conditions of this density path. For diffusion-based methods, we remark that increasing the number of MCMC steps is actually beneficial when no kernel is used, *i.e.*, specifically when the score information is not used at a global scale (between transitions) but a local scale (with MALA sampler); in particular, the no-kernel variant reaches the performance of the version using a first-order denoising kernel after a certain threshold. The other methods show little sensitivity to the number of MCMC steps in this range, suggesting robustness to this hyperparameter choice.

Dependence of RE with respect to the swap period. In Figure E.46, we investigate the sensitivity of all RE-based methods to the choice of the swap period, which corresponds to the number of local MCMC steps performed between swap operations (selected in $\{2, 4, 8, 16, 32, 64\}$), for the range of annealing levels $K \in \{32, 64, 128, 256\}$. Note that the overall MCMC budget is the same as in the main experiments in order to isolate the impact of the swap period on the sampling results; for comparison, we display with a dashed line the swap period used by default (set to 8). Based on these results, we may identify two groups of samplers : (i) methods with poor performance (tempering path, diffusion path without kernel or with first-order stochastic kernel), for which the results are globally independent to the setting of the swap period, and (ii) methods with better performance (diffusion path with second-order stochastic kernel or deterministic kernels), where we observe a sweet spot for the swap period. For the latter methods, this result clearly validates the intuitive trade-off between global exploration and local exploration in RE, induced by the number of local MCMC steps between swaps under a fixed global sampling budget. Indeed, while increasing the swap period tends to prioritize local exploration, decreasing the swap period rather favors global exploration, thus hindering the overall performance of RE in both cases.

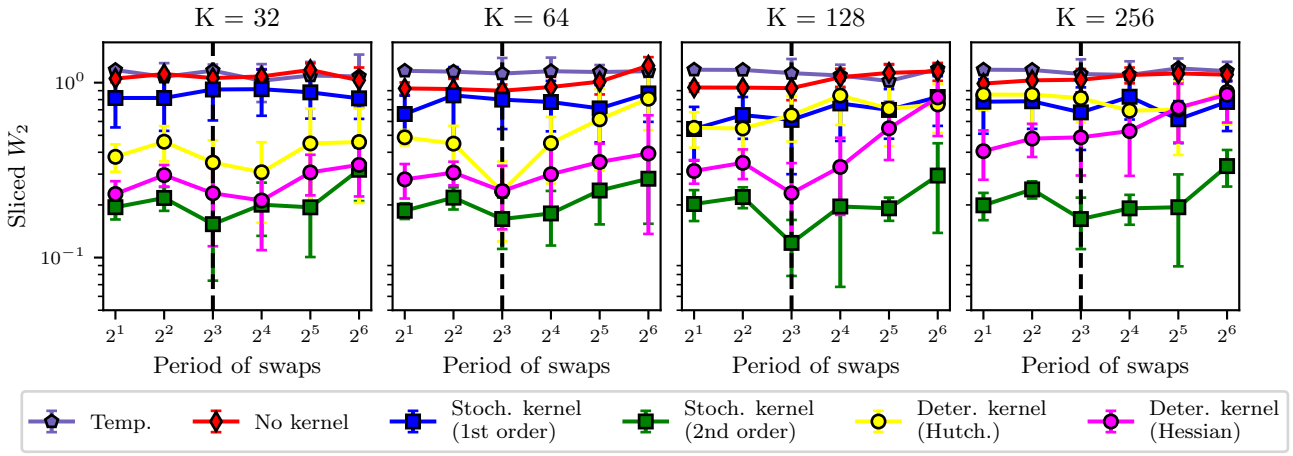


Figure E.46: **Sensitivity of RE methods (including diffusion-based BGs) with respect to the swap period across different values of K.** The global number of MCMC steps is the same as in the main experiments. The default setting of the main experiments is represented by a dashed line. For the three best performing methods, we observe a sweet spot when varying the swap period.

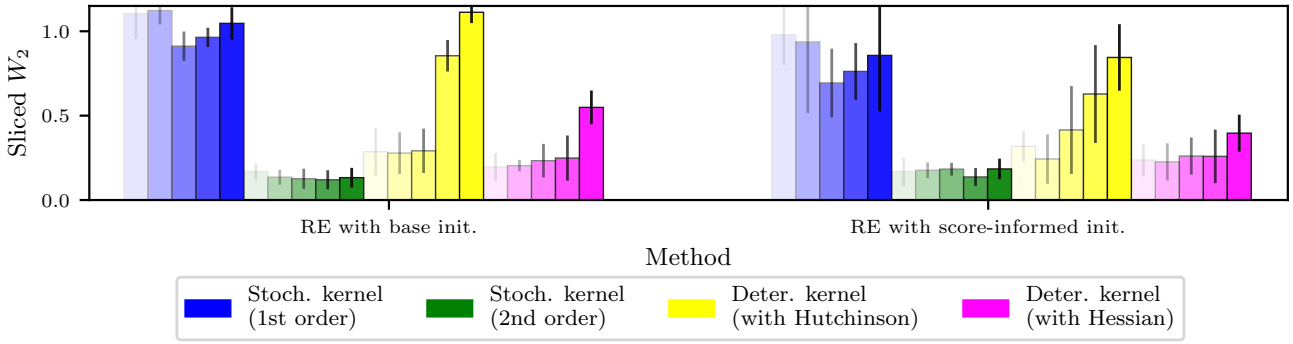


Figure E.47: **Sensitivity of diffusion-based RE-BGs methods with respect to the per-level initialization across different values of $K \in \{16, 32, 64, 128, 256\}$: the darker the bar, the higher K.** The group of bars displayed on the right of the figure ("RE with score-informed init") exactly corresponds to the RE bars from the third row of Figure 6.3. Compared to the naive initialization ("RE with base init"), populating the levels by simulating the reverse SDE performs on par or better for each setting.

Dependence of RE with respect to the per-level initialization. When targeting the diffusion path with RE samplers that exploit at least first-order information (*i.e.*, the scores of the intermediate marginal distributions), one may leverage this knowledge to wisely initialize the annealing levels. Indeed, by simulating the reverse SDE starting from π^{base}, we are able to populate each level with samples that are approximately distributed according to the related target marginal distribution, thereby offering a favorable setting for faster MCMC convergence. In our main experiments, we systematically adopt this strategy, and compare it to the naive initialization (that is performed in the standard version of RE) with samples from π^{base} at each level. The results displayed in Figure E.47 (which use the same color code as Figure 6.3) show that the score-informed warm-up initialization clearly benefits to all RE variants, especially when the number of levels is high.

Analysis of diffusion-based aMC-BGs with deterministic transitions. Finally, we propose three ablation studies focused on the diffusion-based variants of AIS, SMC and RE introduced in Section 6.4. For each annealed sampling method, we investigate for the range of annealing levels $K \in \{32, 64, 128\}$, the following :

• their inter-dependence to the number of fixed-point iterations $M \in \{2, 4, 8, 16\}$ and the truncation order index $I \in \{1, 2, 3, 4, 5\}$, see Figure E.48. For all samplers, we observe that the performance of the Hessian-based variant is globally independent to the setting of (M, I), thereby proving its robust accuracy under restricted computational budgets (especially lower than in the main experiments). For AIS/SMC methods, we observe that the performance gap between Hutchinson and Hessian-based variants diminishes when taking K larger with fixed (M, I), and is relatively constant with respect to (M, I) with large values of K; on the other hand, when K is low, we observe that a large number of fixed-point iterations is needed to close this gap. Inversely, this gap is very small at low K for RE sampler, and gets larger as K increases; interestingly, we do not note a significant dependence to (M, I) in RE variants.

• their inter-dependence to the truncation order index $I \in \{1, 2, 3, 4, 5\}$ and the number of Hutchinson samples $N_H \in \{2, 4, 8, 16, 32, 64, 128\}$ (here, the Hessian is used as the baseline), see Figure E.49. For AIS/SMC samplers, we observe that, for fixed values of N_H and K, taking a large truncation order often degrades the performance of the Hutchinson-based approach, especially when K is low. We attribute this behavior to the growing error accumulation of the expansion series terms, when increasing the value of I. For those samplers, it seems more advantageous to rather consider a large value of K, as well as intermediary values for I and N_H, in order to reach a good compromise in the estimation of the Jacobian log-determinants via expansion series. On the other hand, for RE sampler, the gap between the Hutchinson-based method and its baseline is remarkably constant with the choice of (I, N_H), but tends to increase when K is large.

• their inter-dependence to the number of fixed-point iterations $M \in \{2, 4, 8, 16\}$ and the number of Hutchinson samples $N_H \in \{2, 4, 8, 16, 32, 64, 128\}$ (here, the Hessian is used as the baseline), see Figure E.50. Based on the obtained results, we may draw consistent conclusions with respect to the two previous ablation studies: increasing the number of fixed-point iterations does not seem to specifically improve the accuracy of all annealed samplers with deterministic transitions (with or without access to Hessians), setting N_H with an intermediary value (for instance, 32 as we do in our main experiments) provides a good compromise between accuracy and computational effort, as long as K is large for AIS/SMC samplers or low for RE sampler.

Note that we display within each figure a dashed line to represent the setting used in our main experiments, given by $(M, I, N_H) = (4, 3, 32)$. In all presented ablation studies, we observe that it systematically embodies a positive balance between computational cost and empirical performance, thus motivating its use it as default setting.

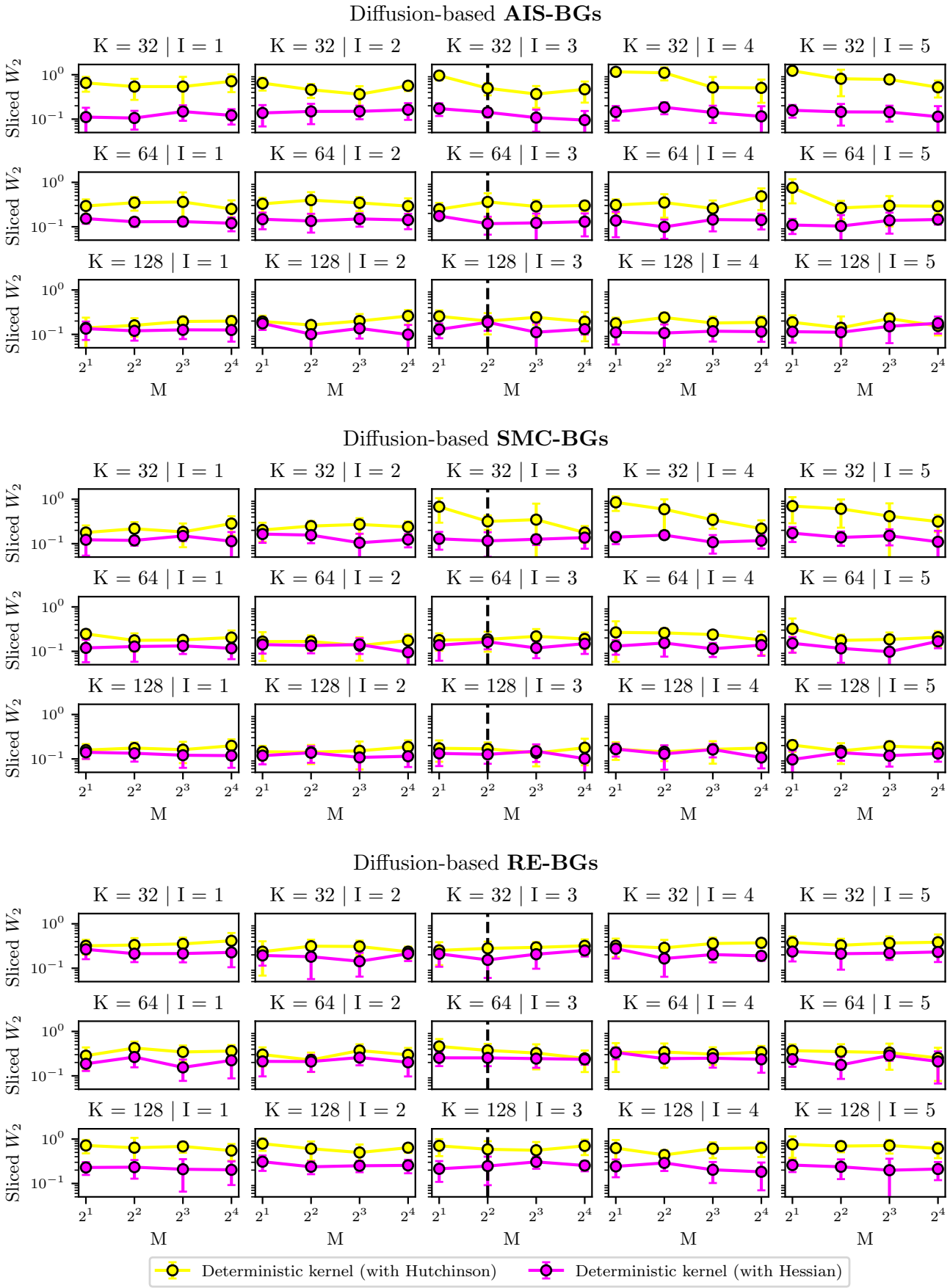


Figure E.48: **Sensitivity of all diffusion-based aMC-BGs with deterministic transitions with respect to the number of fixed-point iterations (x-axis) and the truncation order I, across different values of K. (Top)** AIS variant, **(Middle)** SMC variant, **(Bottom)** RE variant. We fix $N_H = 32$.

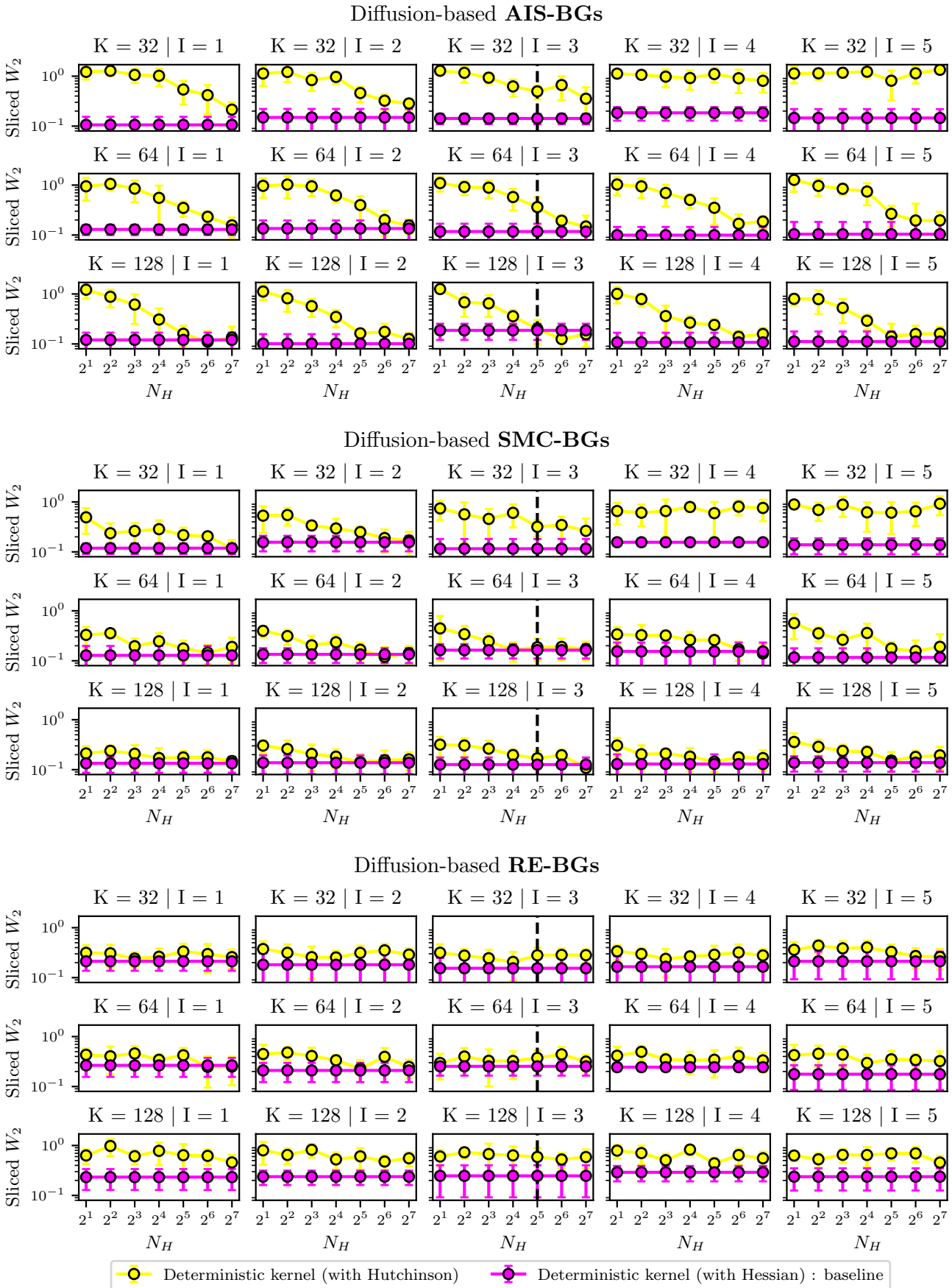


Figure E.49: **Sensitivity of all diffusion-based aMC-BGs with deterministic transitions with respect to the number of Hutchinson samples N_H (x-axis) and the truncation order I, across different values of K. (Top)** AIS variant, **(Middle)** SMC variant, **(Bottom)** RE variant. We fix $M = 4$.

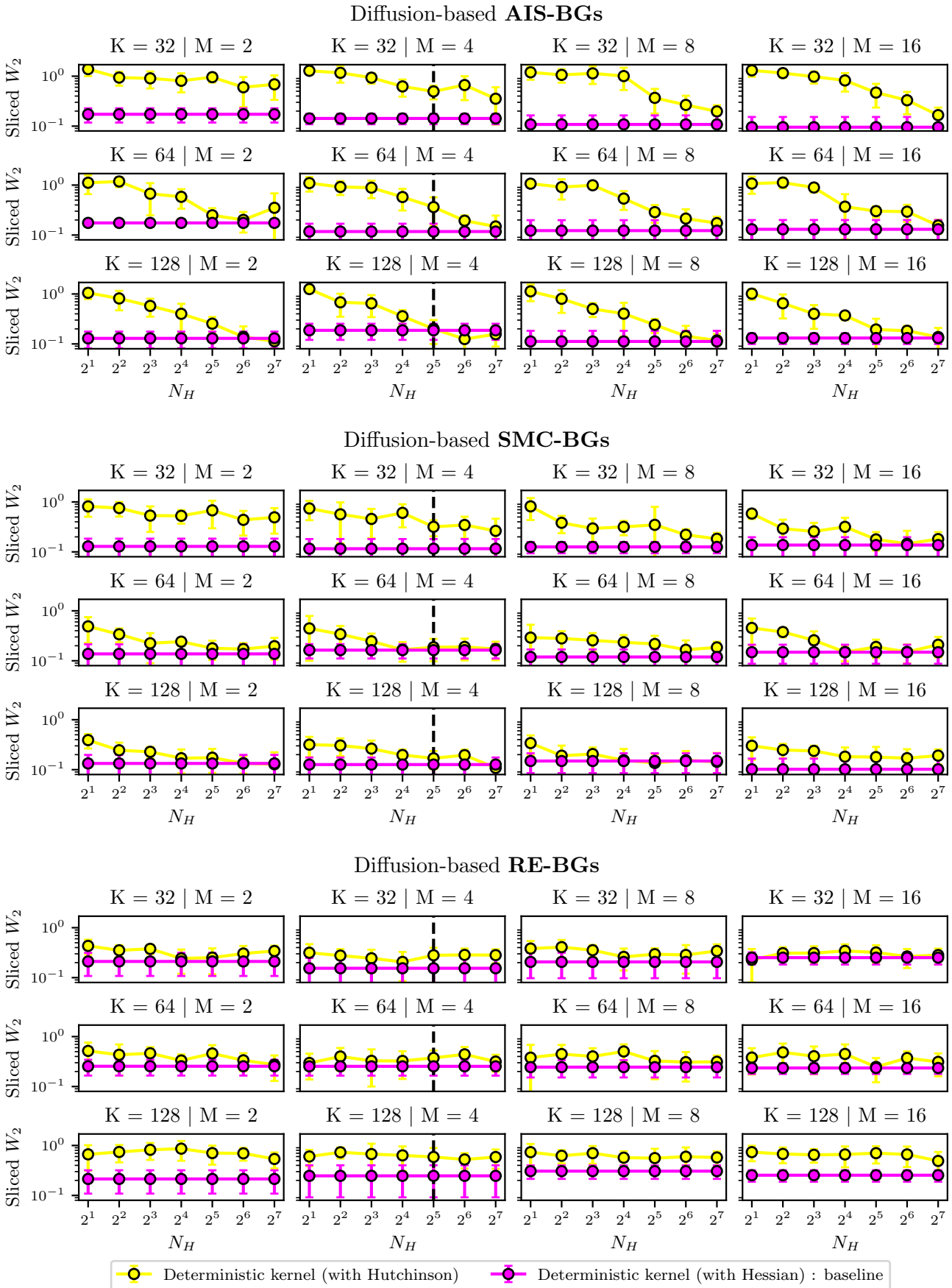


Figure E.50: **Sensitivity of all diffusion-based aMC-BGs with deterministic transitions with respect to the number of Hutchinson samples N_H (x-axis) and fixed-point iterations M, across different values of K. (Top)** AIS variant, **(Middle)** SMC variant, **(Bottom)** RE variant. We fix $I = 3$.

Appendix F

Appendix of Chapter 7

Appendix organization

First, additional notation is introduced in Appendix F.1. Then, we briefly recall some notions on undirected and directed trees in Appendix F.2. Similarly, martingale problems are introduced in Appendix F.3. The proofs of the main manuscript and additional theoretical results on Tree Schrödinger Bridges are given in Appendix F.4. Additional details on our consideration of the tree-based static SB problem are described in Appendix F.5. Details on the implementation of TreeDSB are given in Appendix F.6 and the experiments are investigated in Appendix F.7.

F.1 Additional notation

For any finite set E, we equivalently refer to the cardinal of E as $\mathrm{card}(\mathsf{E})$ or $|\mathsf{E}|$. Let $(\mathsf{X}, \mathcal{X})$ be a measurable space. For any $x \in (\mathbb{R}^d)^{\ell+1}$ and any $m \in \{0, \dots, \ell\}$, let $x_{-m} = (x_0, \dots, x_{m-1}, x_{m+1}, \dots, x_\ell)$. For any family of measures $\{\nu_j\}_{j\in\{0,\dots,\ell\}}$ defined on $(\mathsf{X}, \mathcal{X})$ and any $i \in \{0, \dots, \ell\}$, let $\nu_{-i} = \bigotimes_{j\in\{0,\dots,\ell\}\setminus\{i\}} \nu_j$. Let $I = \{i_1, \dots, i_q\} \subset \{1, \dots, \ell\}$ and $\mu \in \mathscr{P}^{(\ell)}$ such that $\mu \ll \mathrm{Leb}$. We define $I^{\mathrm{c}} = \{1, \dots, \ell\}\setminus I$ and denote it by $\{i_1^{\mathrm{c}}, \dots, i_{\bar{q}}^{\mathrm{c}}\}$ where $\bar{q} = \ell - q$. We denote the marginal of μ along I by μ_I, *i.e.*, $\mu_I \in \mathscr{P}^{(q)}$ and we have for any $\mathsf{A} \in \mathcal{B}((\mathbb{R}^d)^q)$, $\mu_I(\mathsf{A}) = \int_{\mathsf{X}} \mu(x) \prod_{j=1}^q \delta_{x_{i_j}}(\mathsf{A}_j)\mathrm{d}x$. In addition, note that $\mu_I \ll \mathrm{Leb}$. We denote the conditional distribution of μ given I by $\mu_{|I}(\cdot|\cdot)$, *i.e.*, $\mu_{|I}(\cdot|\cdot) \in \mathscr{P}^{(\bar{q})} \times (\mathbb{R}^d)^q$ and we have for any $y \in (\mathbb{R}^d)^q$ and any $\mathsf{A} \in \mathcal{B}((\mathbb{R}^d)^{\bar{q}})$, $\mu_{|I}(\mathsf{A}|y) = \int_{\mathsf{X}} \mu(x)/\mu_I(y) \prod_{j=1}^q \delta(x_{i_j} - y_j) \prod_{j'=1}^{\bar{q}} \delta_{x_{i_{j'}^{\mathrm{c}}}}(\mathsf{A}_{j'})\mathrm{d}x$. Remark that for any $y \in (\mathbb{R}^d)^q$, $\mu_{|I}(\cdot|y) \ll \mathrm{Leb}$. For any subset $\mathsf{J} \subset \mathsf{I}^{\mathrm{c}}$ with $\mathrm{card}(\mathsf{J}) = q_{\mathsf{J}}$, we also define $\mu_{\mathsf{J}|\mathsf{I}}(\cdot|\cdot) \in \mathscr{P}^{(q_{\mathsf{J}})} \times (\mathbb{R}^d)^q$ such that for any $y \in (\mathbb{R}^d)^q$, $\mu_{\mathsf{J}|\mathsf{I}}(\cdot|y) = \{\mu_{|\mathsf{I}}(\cdot|y)\}_{\mathsf{J}}$. For a collection of functions $\{f_i\}_{i\in\mathsf{I}}$, with $\mathsf{I} \subset \{1, \dots, n\}$ and $n \in \mathbb{N}$ such that $f_i : \mathbb{R}^d \to \mathbb{R}$, we define $\oplus_{i\in\mathsf{I}} f_i : (\mathbb{R}^d)^n \to \mathbb{R}$ such that for any $x = (x_1, \dots, x_n) \in (\mathbb{R}^d)^n$, $\oplus_{i\in\mathsf{I}} f(x) = \sum_{i\in\mathsf{I}} f_i(x_i)$.

F.2 Introduction to trees

Undirected tree. An undirected graph $\mathsf{T} = (\mathsf{V}, \mathsf{E})$, with vertices V and edges E, is said to be an *undirected tree* if it is *acyclic* and *connected* [Valiente, 2002, Definition 1.19.]. In particular, we have $\mathrm{card}(\mathsf{E}) = \mathrm{card}(\mathsf{V}) - 1$. The undirected edge between two nodes v_1 and v_2 is similarly denoted by $\{v_1, v_2\}$ or $\{v_2, v_1\}$. We say that $\mathsf{T}' = (\mathsf{V}', \mathsf{E}')$ is a *sub-tree* of T if T' is an undirected tree with vertices $\mathsf{V}' \subset \mathsf{V}$ and edges $\mathsf{E}' \subset \mathsf{E}$. For any vertex $v \in \mathsf{V}$, we define the set of its *neighbours* N_v as the set of vertices $v' \in \mathsf{V}$ such that $\{v, v'\} \in \mathsf{E}$. The integer $\mathrm{card}(\mathsf{N}_v)$ is referred to as the degree of v. The vertices with degree 1 are called *leaves*, and we denote the set of leaves by $\mathsf{V}_{\mathsf{L}} \subset \mathsf{V}$. The (unique) *path* in T between two vertices v and v' is the sequence of two-by-two distinct edges $\{\{v_i, v_{i+1}\}\}_{i=1}^n$ (with $n \geq 1$) such that $v_k = v_{k+1}$ for any $k \in \{1, \dots, n\}$ such that $k = 0 \mod(2)$, $v_1 = v$ and $v_{n+1} = v'$. This path can be seen as a linear sub-tree of T, and we define n as the *length* of this path. We say that T is *weighted* if there exists a map $w : \mathsf{E} \mapsto \mathbb{R}_+$; in this case, $w(\{v_1, v_2\})$, or equivalently $w(\{v_2, v_1\})$ (also denoted by w_{v_1,v_2} or w_{v_2,v_1}) is called the weight of the edge $\{v_1, v_2\}$. The tree T is said to be *rooted* in $r \in \mathsf{V}$ if r defines a partial ordering $\leq_{\mathsf{T},r} \subset \mathsf{V} \times \mathsf{V}$ such that for any $v_1, v_2 \in \mathsf{V}$, $v_1 \leq_{\mathsf{T},r} v_2$ if the node v_1 lies on the unique path between r and v_2.

Directed tree. Consider a directed graph $\mathsf{T}_r = (\mathsf{V}, \mathsf{E}_r)$ rooted in $r \in \mathsf{V}$. Any directed edge $e \in \mathsf{E}_r$ from $v_1 \in \mathsf{V}$ to $v_2 \in \mathsf{V}$ is denoted by (v_1, v_2). T_r is a said to be a *directed tree* rooted in r if (i) the underlying undirected graph $\mathsf{T} = (\mathsf{V}, \mathsf{E})$ is an undirected tree rooted in r and (ii) any $(v_1, v_2) \in \mathsf{E}_r$ is directed according to the partial ordering $\leq_{\mathsf{T},r}$, *i.e.*, $\{v_1, v_2\} \in \mathsf{E}$ and $v_1 \leq_{\mathsf{T},r} v_2$. For any vertices $(v, v') \in \mathsf{V} \times \mathsf{V}$ such that $v \leq_{\mathsf{T},r} v'$, the (unique) *path* in T_r from v to v', denoted by $\mathrm{path}_{\mathsf{T}_r}(v, v')$, is defined as the directed version of the path in T between v and v' (viewed as a sub-tree of T), which is rooted in v. We say that T_r is *weighted*, if T is weighted and the edges of T_r have the same weights as the corresponding undirected edges of T. For any $(v_1, v_2) \in \mathsf{E}_r$, we denote this weight by w_{v_1,v_2}. We say that T_r is the (unique) *directed version* of T rooted in r. It is endowed with a canonical vertex numbering $\zeta : \mathsf{V} \to \{0, \ldots, \mathrm{card}(\mathsf{V}) - 1\}$, corresponding to a depth-first traversal of its nodes, starting from the root r [Valiente, 2002, Definition 3.1.]. This numbering is consistent with the partial ordering on T, *i.e.*, if $v_1 \leq_{\mathsf{T},r} v_2$, $\zeta(v_1) \leq \zeta(v_2)$, and satisfies $\zeta(r) = 0$. In the rest of the paper, we will write in an equivalent manner v or $\zeta(v)$.

For any vertices $(v_1, v_2) \in \mathsf{E} \times \mathsf{E}$ such that $v_1 \leq_{\mathsf{T},r} v_2$, $\mathrm{path}_{\mathsf{T}_r}(v_1, v_2)$ corresponds to the ordered set of edges in E_r which define the ordered path between two vertices v_1 and v_2. For any vertex $v \in \mathsf{V}$, we define:

(a) the set of its *children* C_v as the set of vertices $v' \in \mathsf{V}$ such that $(v, v') \in \mathsf{E}_r$. In particular, for any $v \in \mathsf{V}_L$, the set of leaves, one has $\mathsf{C}_v = \emptyset$.

(b) its *parent* as the unique vertex $p(v)$ such that $(p(v), v) \in \mathsf{E}_r$, if $v \neq r$ (the parent of the root is not defined).

Note that $\mathsf{N}_r = \mathsf{C}_r$ and, for any vertex $v \in \mathsf{V} \backslash \{r\}$, $\mathsf{N}_v = \{p(v)\} \cup \mathsf{C}_v$.

Definition 113 (Tree-structured directed Probabilistic Graphical Model (PGM))**.** *Consider a directed tree $\mathsf{T}_r = (\mathsf{V}, \mathsf{E}_r)$. The directed PGM induced by T_r [Koller and Friedman, 2009, Definition 3.4.], denoted by $\mathscr{P}_{\mathsf{T}_r}$, is the family of distributions $\pi \in \mathscr{P}^{(|\mathsf{V}|)}$ which have a Markovian factorization along T_r, i.e.,*

$$\mathscr{P}_{\mathsf{T}_r} = \{\pi \in \mathscr{P}^{(|\mathsf{V}|)} : \pi = \pi_r \bigotimes_{(v,v') \in \mathsf{E}_r} \pi_{v'|v}\} \,. \tag{F.1}$$

Lemma 114. *Consider an undirected tree $\mathsf{T} = (\mathsf{V}, \mathsf{E})$. Let $(r, r') \in \mathsf{V} \times \mathsf{V}$. Let T' be a sub-tree of T with vertices V' such that $r' \in \mathsf{V}'$. Denote by $\mathsf{T}'_{r'}$ the directed version of T' rooted in r'. Then, for any $\pi \in \mathscr{P}_{\mathsf{T}_r}$, we have $\pi_{\mathsf{V}'} \in \mathscr{P}_{\mathsf{T}'_{r'}}$.*

Proof. Let $(r, r') \in \mathsf{V} \times \mathsf{V}$. We denote by $\mathsf{T}_r = (\mathsf{V}, \mathsf{E}_r)$, respectively $\mathsf{T}_{r'} = (\mathsf{V}, \mathsf{E}_{r'})$, the directed version of T rooted in r, respectively r'. We define the paths $\mathsf{P}_{r,r'} = \mathrm{path}_{\mathsf{T}_r}(r, r') \subset \mathsf{E}_r$ and $\mathsf{P}_{r',r} = \mathrm{path}_{\mathsf{T}_{r'}}(r', r) \subset \mathsf{E}_{r'}$. It is easy to see that

(a) $\mathsf{E}_r \backslash \mathsf{P}_{r,r'} = \mathsf{E}_{r'} \backslash \mathsf{P}_{r',r}$,

(b) $\mathsf{P}_{r,r'} = \{(v_2, v_1) : (v_1, v_2) \in \mathsf{P}_{r',r}\}$,

(c) $\mathsf{P}_{r',r} = \{(v_2, v_1) : (v_1, v_2) \in \mathsf{P}_{r,r'}\}$.

Let $\pi \in \mathscr{P}_{\mathsf{T}_r}$. First note that for any $(v_1, v_2) \in \mathsf{E}_r$, we have by Bayes decomposition $\pi_{v_1} \pi_{v_2|v_1} = \pi_{v_2} \pi_{v_1|v_2} = \pi_{v_1,v_2}$. Then it comes

$$\begin{aligned}
\pi &= \pi_r \bigotimes_{(v_1,v_2) \in \mathsf{E}_r} \pi_{v_2|v_1} \\
&= \pi_r \bigotimes_{(v_1,v_2) \in \mathsf{P}_{r,r'}} \pi_{v_2|v_1} \bigotimes_{(v_1,v_2) \in \mathsf{E}_r \backslash \mathsf{P}_{r,r'}} \pi_{v_2|v_1} \\
&= \pi_r \bigotimes_{(v_2,v_1) \in \mathsf{P}_{r',r}} \pi_{v_2|v_1} \bigotimes_{(v_1,v_2) \in \mathsf{E}_{r'} \backslash \mathsf{P}_{r',r}} \pi_{v_2|v_1} \\
&= \pi_{r'} \bigotimes_{(v_1,v_2) \in \mathsf{P}_{r',r}} \pi_{v_2|v_1} \bigotimes_{(v_1,v_2) \in \mathsf{E}_{r'} \backslash \mathsf{P}_{r',r}} \pi_{v_2|v_1} \\
&= \pi_{r'} \bigotimes_{(v_1,v_2) \in \mathsf{E}_{r'}} \pi_{v_2|v_1} \,,
\end{aligned}$$

and therefore, we have $\pi \in \mathscr{P}_{\mathsf{T}_{r'}}$.

Let T' be a sub-tree of T with vertices V' such that $r' \in \mathsf{V}'$. First note that $\mathsf{E}'_{r'} \subset \mathsf{E}_{r'}$. Using the previous computation, we have for any $\mathsf{A} \in \mathcal{B}((\mathbb{R}^d)^{|\mathsf{V}'|})$,

$$\begin{aligned}
\pi_{\mathsf{V}'}(\mathsf{A}) &= \textstyle\int_{(\mathbb{R}^d)^{|\mathsf{V}|}} \pi_{r'}(x_{r'}) \bigotimes_{(v_1,v_2) \in \mathsf{E}_{r'}} \pi_{v_2|v_1}(x_{v_2}|x_{v_1}) \prod_{v' \in \mathsf{V}'} \delta_{x_{v'}}(\mathsf{A}_{v'}) \mathrm{d}x \\
&= \textstyle\int_{(\mathbb{R}^d)^{|\mathsf{V}|-|\mathsf{V}'|}} \{\pi_{r'}(\mathsf{A}_{r'}) \bigotimes_{(v_1,v_2) \in \mathsf{E}'_{r'}} \pi_{v_2|v_1}(\mathsf{A}_{v_2}|x_{v_1})\} \bigotimes_{(v_1,v_2) \in \mathsf{E}_{r'} \backslash \mathsf{E}'_{r'}} \pi_{v_2|v_1}(x_{v_2}|x_{v_1}) \mathrm{d}x_{\mathsf{V} \backslash \mathsf{V}'} \\
&= \{\pi_{r'} \bigotimes_{(v_1,v_2) \in \mathsf{E}'_{r'}} \pi_{v_2|v_1}\}(\mathsf{A}) \,,
\end{aligned}$$

which proves that $\pi_{\mathsf{V}'} \in \mathscr{P}_{\mathsf{T}'_{r'}}$. □

Discretized undirected tree. Let $N \geq 1$. Consider an undirected tree $\mathsf{T} = (\mathsf{V}, \mathsf{E})$ with weights w. We say that $\mathsf{T}^{(N)} = (\mathsf{V}^{(N)}, \mathsf{E}^{(N)})$ is a N-discretized version of T if it is an undirected tree with weights $w^{(N)}$ such that

(a) $\mathsf{V}^{(N)} = \mathsf{V} \bigsqcup \cup_{\substack{e \in \mathsf{E}, \\ k \in \{1,\dots,N-1\}}} \{v_e^k\}$,

(b) $\mathsf{E}^{(N)} = \cup_{e \in \mathsf{E}} \cup_{k=0,\dots,N-1} \left\{\{v_e^k, v_e^{k+1}\}\right\}$ with the convention that the vertices v_e^N and v_e^N are defined such that $\{v_e^0, v_e^N\} = e$,

(c) $\sum_{e \in \mathrm{path}_{\mathsf{T}}(v,v')} 1/w_e^{(N)} = 1/w_{v,v'}$, if $\{v, v'\} \in \mathsf{E}$.

Remark that the leaves of $\mathsf{T}^{(N)}$ are exactly the original leaves of T and that $\mathsf{T}^{(1)} = \mathsf{T}$. The non-uniqueness of $\mathsf{T}^{(N)}$ comes from the freedom of choice on the weights of its edges.

Discretized directed tree. Let $N \geq 1$. Consider a directed tree $\mathsf{T}_r = (\mathsf{V}, \mathsf{E}_r)$ rooted in $r \in \mathsf{V}$ with weights w. We say that $\mathsf{T}_r^{(N)} = (\mathsf{V}^{(N)}, \mathsf{E}_r^{(N)})$ is a N-discretized version of T_r if it is the directed version of $\mathsf{T}^{(N)}$ rooted in r, where $\mathsf{T}^{(N)}$ is a N-discretized version of the underlying undirected tree of T_r.

F.3 Background on martingale problems

In this section, we introduce the background on Stochastic Differential Equations (SDEs) and weak solutions of SDEs following the framework of [Stroock and Varadhan, 1997, Section 10.1, page 249]. We recall that $\mathrm{C}_0^\infty(\mathbb{R}^d)$ is the space of infinitely differentiable real-valued functions which vanish at infinity. In addition, we have that $\mathcal{S}_+^d$ is the space of $d \times d$, symmetric, non-negative matrices.

Definition 115. *Let $T > 0$ or $T = +\infty$, $\sigma : [0,T) \times \mathbb{R}^d \to \mathcal{S}_+^d$ and $b : [0,T) \times \mathbb{R}^d \to \mathbb{R}^d$, locally bounded measurable functions. We define the* infinitesimal generator*, $\mathcal{A}$, given for any $f \in \mathrm{C}_0^\infty(\mathbb{R}^d)$, $t \in [0,T)$ and $x \in \mathbb{R}^d$ by*

$$\mathcal{A}_t(f)(x) = \langle b_t(x), \nabla f(x) \rangle + \tfrac{1}{2} \langle \sigma_t(x)\sigma_t(x)^\top, \nabla^2 f(x) \rangle. \tag{F.2}$$

We say that a probability measure $\mathbb{P}$ satisfies the martingale problem for $\mathcal{A}$ *if for any $t \in [0,T)$ and $f \in \mathrm{C}_0^\infty(\mathbb{R}^d)$, we have that $(f(\mathbf{X}_t) - \int_0^t \mathcal{A}_s(f)(\mathbf{X}_s)\mathrm{d}s)_{s \in [0,t]}$ is a $\mathbb{P}$-martingale.*

In the main document, see Section 7.2, we say that "a path measure $\mathbb{P}$ is associated with $\mathrm{d}\mathbf{X}_t = b(t, \mathbf{X}_t)\mathrm{d}t + \sigma(t, \mathbf{X}_t)\mathrm{d}\mathbf{B}_t$ with $(\mathbf{B}_t)_{t \geq 0}$ a d-dimensional Brownian motion" if $\mathbb{P}$ solves the martingale problem associated with $\mathcal{A}$ given by (F.2). Unless specified, we always assume that such a path measure exists and is unique. Below, we recall the following theorem, see [Stroock and Varadhan, 1997, Theorem 10.2.2], which gives sufficient conditions for the existence and uniqueness of solutions to the martingale problem.

Theorem 116. *Assume that for any $x \in \mathbb{R}^d$ we have*

$$\inf\{\langle \theta, \sigma\sigma^\top(s,x)\theta \rangle \ : \ \theta \in \mathbb{R}^d, \ \|\theta\| = 1, \ s \in [0,T]\} > 0, \tag{F.3}$$

$$\lim_{y \to x} \sup\{\|\sigma(s,x) - \sigma(s,y)\| \ : \ s \in [0,T]\} = 0. \tag{F.4}$$

In addition, assume that there exists $C > 0$ such that for any $x \in \mathbb{R}^d$

$$\sup\{\|\sigma\sigma^\top(t,x)\| \ : \ s \in [0,T]\} + \sup\{\langle x, b(t,x) \rangle \ : \ s \in [0,T]\} \leq C(1 + \|x\|^2). \tag{F.5}$$

Then, there exists a unique solution to the martingale problem with initialization $x_0 \in \mathbb{R}^d$.

F.4 Theoretical results on Tree Schrödinger Bridges

We respectively provide in Appendix F.4.1, Appendix F.4.2 and Appendix F.4.3 the proofs of the results of the main manuscript presented in Section 7.3, Section 7.4 and Section 7.5. Finally, we make a detailed comparison between our setting and the framework of Haasler et al. [2021] in Appendix F.4.4. In the rest of this section, we consider an undirected tree $\mathsf{T} = (\mathsf{V}, \mathsf{E})$, where $|\mathsf{V}| = \ell + 1$, and some subset $\mathsf{S} \subset \mathsf{V}$ which we denote by $\mathsf{S} = \{i_0, \dots, i_{K-1}\}$. We define $\mathsf{S}^{\mathrm{c}} = \mathsf{V} \backslash \mathsf{S}$.

F.4.1 Proofs of Section 7.3

Proposition 11 is straightforward to obtain by combining the definition of the Brownian motion with the definition of π^0 given in (7.4). The following lemma details the recursion relation between the (mIPF) iterates, which is key to prove Proposition 12.

Lemma 117. *Let $(\pi^n)_{n\in\mathbb{N}}$ be the sequence given by (mIPF). Let $n \in \mathbb{N}$, $k_n = (n-1) \bmod(K)$, $k_n + 1 = n \bmod(K)$. Denote by T_{k_n}, respectively T_{k_n+1} with edges E_{k_n+1}, the directed version of T rooted in i_{k_n}, respectively in i_{k_n+1}. We have:*

(i) $\pi^n \in \mathscr{P}_{\mathsf{T}_{k_n}}$,

(ii) $\pi^{n+1} = \mu_{i_{k_n+1}} \bigotimes_{(v,v')\in\mathsf{E}_{k_n+1}} \pi^n_{v'|v}$. *In particular, for any $(v,v') \in \mathsf{E}_{k_n+1}$, $\pi^{n+1}_{v'|v} = \pi^n_{v'|v}$.*

Proof. We show the result (i) by recursion on $n \in \mathbb{N}$, and will deduce (ii) from the proof. Using (7.4), we first have $\pi^0 \in \mathscr{P}_{\mathsf{T}_r}$, where r is chosen as i_{K-1}, see Section 7.3. Thus, we obtain the result (i) at step $n = 0$. Assume now that $\pi^n \in \mathscr{P}_{\mathsf{T}_{k_n}}$ for some $n \in \mathbb{N}$.

Consider the paths $\mathsf{P}_n = \mathrm{path}_{\mathsf{T}_{k_n}}(i_{k_n}, i_{k_n+1})$ and $\mathsf{P}_{n+1} = \mathrm{path}_{\mathsf{T}_{k_n+1}}(i_{k_n+1}, i_{k_n})$. Note that these two paths have the same length, denoted by J, and contain the same vertices, denoted by V_n. Let $\pi \in \mathscr{P}^{(\ell+1)}$ such that $\mathrm{KL}(\pi|\pi^n) < +\infty$. We have the following decomposition

$$\mathrm{KL}(\pi|\pi^n) = \mathrm{KL}(\pi_{\mathsf{V}_n}|\pi^n_{\mathsf{V}_n}) + \int_{(\mathbb{R}^d)^{J+1}} \mathrm{KL}(\pi_{|\mathsf{V}_n}|\pi^n_{|\mathsf{V}_n}) \mathrm{d}\pi_{\mathsf{V}_n}(x_{\mathsf{V}_n}) \,. \tag{F.6}$$

Hence, the $(n+1)$-th iterate of (mIPF) is given by $\pi^{n+1} = \pi^{n+1}_{\mathsf{V}_n} \otimes \pi^n_{|\mathsf{V}_n}$, with

$$\pi^{n+1}_{\mathsf{V}_n} = \arg\min\{\mathrm{KL}(\pi|\pi^n_{\mathsf{V}_n}) \,:\, \pi \in \mathscr{P}^{(J+1)},\ \pi_{i_{k_n+1}} = \mu_{i_{k_n+1}}\} \,. \tag{F.7}$$

Since $\pi^n \in \mathscr{P}_{\mathsf{T}_{k_n}}$, we have (i) $\pi^n_{|\mathsf{V}_n} = \bigotimes_{(v,v')\in\mathsf{E}_{k_n}\backslash\mathsf{P}_n} \pi^n_{v'|v}$ and (ii) $\pi^n_{\mathsf{V}_n} \in \mathscr{P}_{\mathsf{P}_{n+1}}$ by Lemma 114, where P_{n+1} is viewed as a directed tree rooted in i_{k_n+1}. Defining $\mathsf{V}_{n+1} = \mathsf{V}_n\backslash\{i_{k_n+1}\}$, we thus have $\pi^n_{\mathsf{V}_n} = \pi^n_{i_{k_n+1}} \otimes \pi^n_{\mathsf{V}_{n+1}|i_{k_n+1}}$ where $\pi^n_{\mathsf{V}_{n+1}|i_{k_n+1}} = \bigotimes_{(v,v')\in\mathsf{P}_{n+1}} \pi^n_{v'|v}$.

Let $\pi \in \mathscr{P}^{(J+1)}$ such that $\pi_{i_{k_n+1}} = \mu_{i_{k_n+1}}$ and $\mathrm{KL}(\pi|\pi^n_{\mathsf{V}_n}) < +\infty$. Similarly to the previous computation, we have the following decomposition

$$\begin{aligned}\mathrm{KL}(\pi|\pi^n_{\mathsf{V}_n}) &= \mathrm{KL}(\pi_{i_{k_n+1}}|\pi^n_{i_{k_n+1}}) + \int_{\mathbb{R}^d} \mathrm{KL}(\pi_{|i_{k_n+1}}|\pi^n_{\mathsf{V}_{n+1}|i_{k+1}}) \mathrm{d}\pi_{i_{k_n+1}}(x_{i_{k_n+1}}) \\ &= \mathrm{KL}(\mu_{i_{k_n+1}}|\pi^n_{i_{k_n+1}}) + \int_{\mathbb{R}^d} \mathrm{KL}(\pi_{|i_{k_n+1}}|\pi^n_{\mathsf{V}_{n+1}|i_{k_n+1}}) \mathrm{d}\mu_{i_{k_n+1}}(x_{i_{k_n+1}}) \,.\end{aligned}$$

Therefore, we obtain

$$\pi^{n+1}_{\mathsf{V}_n} = \mu_{i_{k_n+1}} \otimes \pi^n_{\mathsf{V}_{n+1}|i_{k+1}} = \mu_{i_{k+1}} \bigotimes_{(v,v')\in\mathsf{P}_{n+1}} \pi^n_{v'|v} \,.$$

Noting that $\mathsf{E}_{k_n}\backslash\mathsf{P}_n = \mathsf{E}_{k_n+1}\backslash\mathsf{P}_{n+1}$ and recalling that $\pi^{n+1} = \pi^{n+1}_{\mathsf{V}_n} \otimes \pi^n_{|\mathsf{V}_n}$, it finally comes

$$\pi^{n+1} = \mu_{i_{k_n+1}} \bigotimes_{(v,v')\in\mathsf{P}_{n+1}} \pi^n_{v'|v} \bigotimes_{(v,v')\in\mathsf{E}_{k_n+1}\backslash\mathsf{P}_{n+1}} \pi^n_{v'|v} = \mu_{i_{k_n+1}} \bigotimes_{(v,v')\in\mathsf{E}_{k_n+1}} \pi^n_{v'|v} \,. \tag{F.8}$$

Therefore, $\pi^{n+1} \in \mathscr{P}_{\mathsf{T}_{k_n+1}}$, which achieves the recursion for (i), and we obtain (ii) by (F.8). □

Hence, Lemma 117 shows that the (mIPF) iterates admit a Markovian factorization on T, and can be defined recursively using the edges of T. We now provide the proof of Proposition 12.

Proof of Proposition 12. We will prove this result by recursion on $n \in \mathbb{N}$. Observe that the initialisation is directly given by Proposition 11. Assume now that the result of Proposition 12 stands for some $n \in \mathbb{N}$. Let $k_n = (n-1) \bmod(K)$, $k_n + 1 = n \bmod(K)$. Denote by T_{k_n} with edges E_{k_n}, respectively T_{k_n+1} with edges E_{k_n+1}, the directed version of T rooted in i_{k_n}, respectively in i_{k_n+1}. For any vertex v of T_{k_n+1}, we define $p(v)$ as the (unique) parent of v and $c(v)$ as the unique child of v when it exists. Consider the $(n+1)$-th dynamic iterate defined by (a) and (b), *i.e.*, $(\mathbb{P}^{n+1}_{(v,v')})_{(v,v')\in\mathsf{E}_{k_n+1}}$. To prove that this iterate has the properties stated in Proposition 12, we proceed by recursion on the edges of T_{k_n+1}, following the bread-first order in T_{k_n+1}. In this order, the edge $(i_{k_n+1}, c(i_{k_n+1}))$ is the first to be considered. Remark that $c(i_{k_n+1})$ is well defined since i_{k_n+1} is a leaf of T.

Here, we denote $T_{c(i_{k_n+1}),i_{k_n+1}}$ by T. By construction, we have $\mathbb{P}^{n+1}_{(i_{k_n+1},c(i_{k_n+1}))} = \mu_{i_{k_n+1}} \otimes (\mathbb{P}^n_{(c(i_{k_n+1}),i_{k_n+1})})^R_{|0}$. By recursion assumption, $\mathbb{P}^n_{(c(i_{k_n+1}),i_{k_n+1})} \in \mathscr{P}(\mathrm{C}([0,T],\mathbb{R}^d))$ since $(c(i_{k_n+1}),i_{k_n+1}) \in \mathsf{E}_{k_n}$. Then, $\mathbb{P}^{n+1}_{(i_{k_n+1},c(i_{k_n+1}))}$ is a well defined path measure on $[0,T]$. By definition of the (mIPF) sequence, we have $\mu_{i_{k_n+1}} = \pi^{n+1}_{i_{k_n+1}}$. By recursion assumption, we also have that $\mathrm{Ext}(\mathbb{P}^n_{(c(i_{k_n+1}),i_{k_n+1})}) = \pi^n_{c(i_{k_n+1}),i_{k_n+1}}$. Hence, it comes that $(\mathbb{P}^n_{(c(i_{k_n+1}),i_{k_n+1})})^R_{T|0} = \pi^n_{c(i_{k_n+1})|i_{k_n+1}} = \pi^{n+1}_{c(i_{k_n+1})|i_{k_n+1}}$, where the last equality comes from Lemma 117. Finally, we obtain that $\mathrm{Ext}(\mathbb{P}^{n+1}_{(i_{k_n+1},c(i_{k_n+1}))}) = \pi^{n+1}_{i_{k_n+1},c(i_{k_n+1})}$, which proves the initialisation.

Assume now that $\mathbb{P}^{n+1}$ is well defined and has the right properties, up to some edge in T_{k_n+1}. Consider the following edge, denoted by $(v,v') \in \mathsf{E}_{k_n+1}$, in the breadth-first order. By edge recursion, we have that $\mathrm{Ext}(\mathbb{P}^{n+1}_{(p(v),v)}) = \pi^{n+1}_{p(v),v}$, and thus $\mathbb{P}^{n+1}_{(p(v),v),T_{p(v),v}} = \pi^{n+1}_v$. Define the path $\mathsf{P}_n = \mathrm{path}_{\mathsf{T}_{k_n}}(i_{k_n}, i_{k_n+1})$. Then, we face two cases.

(i) Either $(v,v') \in \mathsf{E}_{k_n} \backslash \mathsf{P}_n$. Then, we have by (a) that

$$\mathbb{P}^{n+1}_{(v,v')} = \mathbb{P}^{n+1}_{(p(v),v),T_{p(v),v}} \otimes \mathbb{P}^n_{(v,v')|0} = \pi^{n+1}_v \otimes \mathbb{P}^n_{(v,v')|0}$$

In particular, $\mathbb{P}^{n+1}_{(v,v')}$ is a well defined path measure on $[0,T_{v,v'}]$. Since $(v,v') \in \mathsf{E}_{k_n}$, $\mathrm{Ext}(\mathbb{P}^n_{(v,v')}) = \pi^n_{v,v'}$ by recursion assumption. In particular, $\mathbb{P}^n_{(v,v'),T_{v,v'}|0} = \pi^n_{v'|v} = \pi^{n+1}_{v'|v}$ where the last equality comes from Lemma 117. We thus have $\mathrm{Ext}(\mathbb{P}^{n+1}_{(v,v')}) = \pi^{n+1}_{v,v'}$.

(ii) Or $(v',v) \in \mathsf{P}_n$. Then, we have by (b) that

$$\mathbb{P}^{n+1}_{(v,v')} = \mathbb{P}^{n+1}_{(p(v),v),T_{v,v'}} \otimes (\mathbb{P}^n_{(v',v)})^R_{|0} = \pi^{n+1}_v \otimes (\mathbb{P}^n_{(v',v)})^R_{|0}$$

In particular, $\mathbb{P}^{n+1}_{(v,v')}$ is a well defined path measure on $[0,T_{v,v'}]$. Here, $(v',v) \in \mathsf{E}_{k_n}$ and thus, $\mathrm{Ext}(\mathbb{P}^n_{(v',v)}) = \pi^n_{v',v}$ by recursion assumption. In particular, $(\mathbb{P}^n_{(v',v)})^R_{T_{v',v}|0} = \pi^n_{v'|v} = \pi^{n+1}_{v'|v}$ where the last equality comes from Lemma 117. We thus have $\mathrm{Ext}(\mathbb{P}^{n+1}_{(v,v')}) = \pi^{n+1}_{v,v'}$.

This achieves the recursion. □

F.4.2 Proofs of Section 7.4

Remark on assumption A6. Although ***A6 is not needed to establish the result of Proposition 13, Corollary 14 and Proposition 15***, it is however crucial in the proof of convergence of (mIPF) stated in Proposition 16. Nevertheless, we choose to keep **A6** as an assumption in the statement of every theoretical result presented in Section 7.4 for sake of clarity.

Additional definitions. We define the set $\mathscr{P}_{\mathsf{S}} = \cap_{i\in\mathsf{S}} \mathscr{P}_i$, where $\mathscr{P}_i = \{\pi \in \mathscr{P}^{(\ell+1)} : \pi_i = \mu_i\}$, *i.e.*, $\mathscr{P}_{\mathsf{S}}$ is the set of all probability measures $\pi \in \mathscr{P}^{(\ell+1)}$ which verify

$$\int_{(\mathbb{R}^d)^{\ell+1}} f_i(x_i) \mathrm{d}\pi(x_{0:\ell}) = \int_{\mathbb{R}^d} f_i(x_i) \mathrm{d}\mu_i(x_i) \ ,$$

for any family of bounded measurable functions $\{f_i\}_{i\in\mathsf{S}} \in \mathrm{C}(\mathbb{R}^d,\mathbb{R})^K$. Since $\mathbb{R}^d$ is separable, there exists a dense family of functions $\{f_i^k\}_{k\in\mathbb{N}^*,i\in\mathsf{S}}$, with $f_i^k \in \mathrm{L}^\infty(\mu_i)$ for any $k \in \mathbb{N}^*$ and any $i \in \mathsf{S}$, such that $\pi \in \mathscr{P}_{\mathsf{S}}$ if and only if

$$\int_{(\mathbb{R}^d)^{\ell+1}} f_i^k(x_i) \mathrm{d}\pi(x_{0:\ell}) = \int_{\mathbb{R}^d} f_i^k(x_i) \mathrm{d}\mu_i(x_i)$$

or equivalently, upon centering f_i^k,

$$\int_{(\mathbb{R}^d)^{\ell+1}} f_i^k(x_i) \mathrm{d}\pi(x_{0:\ell}) = 0 \ .$$

In the rest of the section, we consider such family $\{f_i^k\}_{k\in\mathbb{N}^*,i\in\mathsf{S}}$.

For any $n \in \mathbb{N}^*$, we also define $\mathscr{P}^n_{\mathsf{S}} = \cap_{i\in\mathsf{S}} \mathscr{P}^n_i$, where $\mathscr{P}^n_i = \{\pi \in \mathscr{P}^{(\ell+1)} : \int_{(\mathbb{R}^d)^{\ell+1}} f_i^k(x_i) \mathrm{d}\pi(x_{0:\ell}) = 0, \ \forall k \in \{1,\dots,n\}\}$. In particular, we have

$$\mathscr{P}_{\mathsf{S}} = \cap_{n\in\mathbb{N}^*} \mathscr{P}^n_{\mathsf{S}} \ . \tag{F.9}$$

Finally, (static-mSB) can be rewritten as

$$\pi^\star = \arg\min\{\mathrm{KL}(\pi \mid \pi^0) : \pi \in \mathscr{P}_{\mathsf{S}}\} \ . \tag{F.10}$$

Proof of Proposition 13 and Corollary 14. In this part of the section, we present an extension of the theoretical results from Nutz [2021] to the multi-marginal setting. We first present two technical results, Lemma 118 and Lemma 119, which are respectively adapted from [Nutz, 2021, Lemma 2.10.] and [Nutz, 2021, Lemma 2.11.].

Lemma 118. *Let $\{\tilde{\mu}_j\}_{j\in\mathsf{S}^c}$ be a family of probability measures defined on $(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$. We define $\tilde{\pi}^0 = \bigotimes_{i\in\mathsf{S}} \mu_i \bigotimes_{j\in\mathsf{S}^c} \tilde{\mu}_j$. Let $\mathsf{A} \in \bigotimes_{m=0}^{\ell} \mathcal{B}(\mathbb{R}^d)$ such that $\tilde{\pi}^0(\mathsf{A}) = 1$. Then, for $\tilde{\pi}^0$-almost any $x^\star \in \mathsf{A}$, there exists a family of sets $\{\mathsf{X}_m^0\}_{m=0}^{\ell} \subset (\mathbb{R}^d)^{\ell+1}$ such that*

(a) $\mu_i(\mathsf{X}_i^0) = 1$ for any $i \in \mathsf{S}$, and $\tilde{\mu}_j(\mathsf{X}_j^0) = 1$ for any $j \in \mathsf{S}^c$,

(b) $\mathsf{A}^0 = \mathsf{A} \cap (\prod_{m=0}^{\ell} \mathsf{X}_m^0)$ satisfies $x^\star \in \mathsf{A}^0$ and

$$(x_0^\star, \dots, x_{m-1}^\star, x_m, x_{m+1}^\star, \dots, x_\ell^\star) \in \mathsf{A}^0, \forall x \in \mathsf{A}^0, \forall m \in \{0, \dots, \ell\} \, .$$

Proof. Consider such set A. We define for any $m \in \{0, \dots, \ell\}$ the set

$$\mathsf{X}_m = \{u \in \mathbb{R}^d : \tilde{\pi}^0_{-m}(\mathsf{A}_m^u) = 1\} \, , \tag{F.11}$$

where $\mathsf{A}_m^u = \{y \in (\mathbb{R}^d)^\ell : (y_0, \dots, y_{m-1}, u, y_m, \dots, y_{\ell-1}) \in \mathsf{A}\}$.

Take $i \in \mathsf{S}$. Assume that $\mu_i(\mathsf{X}_i) < 1$. We recall that $\tilde{\pi}^0 = \tilde{\pi}^0_{-i} \otimes \mu_i$. Using Fubini's theorem and that $\int_{\mathsf{A}_i^{x_i}} \mathrm{d}\tilde{\pi}^0_{-i}(x_{-i}) < 1$ for any $x_i \notin \mathsf{X}_i$, we have

$$\begin{aligned} 1 = \tilde{\pi}^0(\mathsf{A}) &= \textstyle\int_{\mathsf{A}} \mathrm{d}\tilde{\pi}^0_{-i}(x_{-i}) \otimes \mathrm{d}\mu_i(x_i) \\ &= \textstyle\int_{\mathbb{R}^d} \{\int_{\mathsf{A}_i^{x_i}} \mathrm{d}\tilde{\pi}^0_{-i}(x_{-i})\} \mathrm{d}\mu_i(x_i) \\ &= \textstyle\int_{\mathsf{X}_i} \{\int_{\mathsf{A}_i^{x_i}} \mathrm{d}\tilde{\pi}^0_{-i}(x_{-i})\} \mathrm{d}\mu_i(x_i) + \int_{\mathsf{X}_i^c} \{\int_{\mathsf{A}_i^{x_i}} \mathrm{d}\tilde{\pi}^0_{-i}(x_{-i})\} \mathrm{d}\mu_i(x_i) \\ &< \mu_i(\mathsf{X}_i) + \mu_i(\mathsf{X}_i^c) = 1 \, , \end{aligned}$$

which is absurd. Therefore, we obtain $\mu_i(\mathsf{X}_i) = 1$, and similarly, we have $\tilde{\mu}_j(\mathsf{X}_j) = 1$ for any $j \in \mathsf{S}^c$. For any $y \in (\mathbb{R}^d)^\ell$, any $m \in \{0, \dots, \ell\}$, we define the set

$$\bar{\mathsf{A}}_m^y = \{u \in \mathbb{R}^d : (y_0, \dots, y_{m-1}, u, y_m, \dots, y_{\ell-1}) \in \mathsf{A}\} \, .$$

Let $i \in \mathsf{S}$. We have by Fubini's theorem

$$\begin{aligned} 1 = \tilde{\pi}^0(\mathsf{A}) &= \textstyle\int_{\mathsf{A}} \mathrm{d}\mu_i(x_i) \otimes \mathrm{d}\tilde{\pi}^0_{-i}(x_{-i}) \\ &= \textstyle\int_{(\mathbb{R}^d)^\ell} \{\int_{\bar{\mathsf{A}}_i^{x_{-i}}} \mathrm{d}\mu_i(x_i)\} \mathrm{d}\tilde{\pi}^0_{-i}(x_{-i}) \\ &= \textstyle\int_{\prod_{\substack{m=0 \\ m\neq i}}^{\ell} \mathsf{X}_i} \{\int_{\bar{\mathsf{A}}_i^{x_{-i}}} \mathrm{d}\mu_i(x_i)\} \mathrm{d}\tilde{\pi}^0_{-i}(x_{-i}) \, , \end{aligned}$$

where the last equality comes from the fact that $\mu_i(\mathsf{X}_i) = 1$ for any $i \in \mathsf{S}$, $\tilde{\mu}_j(\mathsf{X}_j) = 1$ for any $j \in \mathsf{S}^c$ and that $\tilde{\pi}^0 = \bigotimes_{i\in\mathsf{S}} \mu_i \bigotimes_{j\in\mathsf{S}^c} \tilde{\mu}_j$. Consequently, there exists a measurable set $\mathsf{A}_{-i} \subset \prod_{\substack{m=0 \\ m\neq i}}^{\ell} \mathsf{X}_i$ such that the following properties hold: (a) $\mu_i(\bar{\mathsf{A}}_i^y) = 1$ for any $y \in \mathsf{A}_{-i}$, (b) $\tilde{\pi}^0_{-i}(\mathsf{A}_{-i}) = 1$. Similarly, this result holds for any $j \in \mathsf{S}^c$, *i.e.*, there exists a measurable set $\mathsf{A}_{-j} \subset \prod_{\substack{m=0 \\ m\neq j}}^{\ell} \mathsf{X}_i$ such that the following properties hold: (a) $\tilde{\mu}_j(\bar{\mathsf{A}}_j^y) = 1$ for any $y \in \mathsf{A}_{-j}$, (b) $\tilde{\pi}^0_{-j}(\mathsf{A}_{-j}) = 1$. We consider such sets $\{\mathsf{A}_{-m}\}_{m=0}^{\ell}$ for the rest of the proof and finally define the set

$$\tilde{\mathsf{A}} = \cap_{m=0}^{\ell} \tilde{\mathsf{A}}_m \, ,$$

where $\tilde{\mathsf{A}}_m = \mathsf{A}_{-m} \times \{u \in \bar{\mathsf{A}}_m^y : y \in \mathsf{A}_{-m}\}$. By definition, we have $\tilde{\mathsf{A}} \subset \mathsf{A} \cap \prod_{m=0}^{\ell} \mathsf{X}_m$, using the fact that $\tilde{\mathsf{A}}_m \subset \mathsf{A}$ for any $m \in \{0, \dots, \ell\}$. In addition, for any $i \in \mathsf{S}$, we get by Fubini's theorem

$$\tilde{\pi}^0(\tilde{\mathsf{A}}_i) = \textstyle\int_{\tilde{\mathsf{A}}_i} \mathrm{d}\mu_i(x_i) \otimes \mathrm{d}\tilde{\pi}^0_{-i}(x_{-i}) = \int_{\mathsf{A}_{-i}} \{\int_{\bar{\mathsf{A}}_i^{x_{-i}}} \mathrm{d}\mu_i(x_i)\} \mathrm{d}\tilde{\pi}^0_{-i}(x_{-i}) = \tilde{\pi}^0_{-i}(\mathsf{A}_{-i}) = 1 \, ,$$

and similarly, we get $\tilde{\pi}^0(\tilde{\mathsf{A}}_j) = 1$ for any $j \in \mathsf{S}^c$. We can deduce that $\tilde{\pi}^0(\tilde{\mathsf{A}}) = 1$ since $\tilde{\pi}^0(\tilde{\mathsf{A}}^c) \leq \sum_{m=0}^{\ell} \tilde{\pi}^0(\tilde{\mathsf{A}}_m^c) = 0$.

Let $x^\star \in \tilde{\mathsf{A}}$. In particular, $x^\star \in \mathsf{A}$. We define the set $\mathsf{A}^0 = \mathsf{A} \cap (\prod_{m=0}^{\ell} \mathsf{X}_m^0)$, where $\mathsf{X}_m^0 = \mathsf{X}_m \cap \bar{\mathsf{A}}_m^{x^\star_{-m}}$ for any $m \in \{0, \dots, \ell\}$. We now establish the result of Lemma 118.

We first prove (a). Let $i \in \mathsf{S}$. Since $x^\star \in \tilde{\mathsf{A}}$, we have $x^\star \in \tilde{\mathsf{A}}_i$ and therefore $x^\star_{-i} \in \mathsf{A}_{-i}$. By definition of A_{-i}, we obtain that $\mu_i(\bar{\mathsf{A}}_i^{x^\star_{-i}}) = 1$ and thus,

$$\mu_i(\{\mathsf{X}_i^0\}^{\mathrm{c}}) \leq \mu_i(\mathsf{X}_i^{\mathrm{c}}) + \mu_i(\{\bar{\mathsf{A}}_i^{x^\star_{-i}}\}^{\mathrm{c}}) = 0,$$

which gives $\mu_i(\mathsf{X}_i^0) = 1$, and similarly, we have $\tilde{\mu}_j(\mathsf{X}_j^0) = 1$ for any $j \in \mathsf{S}^{\mathrm{c}}$.

We now prove (b). Let $m \in \{0, \dots, \ell\}$. Since $x^\star \in \tilde{\mathsf{A}} \subset \mathsf{A}$, we get $x^\star_m \in \bar{\mathsf{A}}_m^{x^\star_{-m}}$. Using that $\tilde{\mathsf{A}} \subset \mathsf{A} \cap_{m=0}^{\ell} \mathsf{X}_m$, we get $x^\star \in \mathsf{A}^0$. Let $x \in \mathsf{A}^0$. We denote $x^m = (x_0^\star, \dots, x_{m-1}^\star, x_m, x_{m+1}^\star, \dots, x_\ell^\star)$. We need to show that $x^m \in \mathsf{A}$ and $x^m \in \prod_{j=1}^{\ell} \mathsf{X}_j^0 = \prod_{j=1}^{\ell} (\mathsf{X}_j \cap \bar{\mathsf{A}}_j^{x^\star_{-m}})$. First, since $x_j^m = x_j$ or $x_j^\star$ for any $j \in \{0, \dots, \ell\}$, and $x \in \mathsf{A}^0$ and $x^\star \in \mathsf{A}^0$, we get that for any $j \in \{0, \dots, \ell\}$, $x_j^m \in \mathsf{X}_j$. Similarly, for any $j \in \{0, \dots, \ell - 1\}$, $x_j^m \in \bar{\mathsf{A}}_j^{x^\star_{-m}}$. Therefore, we get that $x^m \in \prod_{j=1}^{\ell} (\mathsf{X}_j \cap \bar{\mathsf{A}}_j^{x^\star_{-m}})$. Since $x_m \in \mathsf{A}_m^{x^\star_{-m}}$ (because $x \in \prod_{j=1}^{\ell} (\mathsf{X}_j \cap \bar{\mathsf{A}}_j^{x^\star_{-m}})$), we get that $x \in \mathsf{A}$, which concludes the proof. □

Lemma 119. *Let $\mathsf{A}^0 \subset (\mathbb{R}^d)^{\ell+1}$. For any $m \in \{0, \dots, \ell\}$, we denote $\mathsf{X}_m^0 = \mathrm{proj}_m(\mathsf{A}^0)$. We make the following assumptions.*

(a) Assume there exists $x^\star \in \mathsf{A}^0$ such that for any $x \in \mathsf{A}^0$, for any $m \in \{0, \dots, \ell\}$, we have $(x_0^\star, \dots, x_{m-1}^\star, x_m, x_{m+1}^\star, \dots, x_\ell^\star) \in \mathsf{A}^0$.

(b) Assume there exists a family of functions $\{\varphi_{i_k}^n\}_{n \in \mathbb{N}^*, k \in \{0, \dots, K-1\}}$ with $\varphi_{i_k}^n : \mathsf{X}_{i_k}^0 \to [-\infty, +\infty)$ such that for any $n \in \mathbb{N}^*$ and any $k \in \{0, \dots, K-2\}$, we have $\varphi_{i_k}^n(x_{i_k}^\star) = 0$.

(c) Denote $F^n(x) = \sum_{k=0}^{K-1} \varphi_{i_k}^n(x_{i_k})$ for any $x \in \mathsf{A}^0$. Assume that for any $x \in \mathsf{A}^0$, $F(x) = \lim_{n \to \infty} F^n(x)$ exists and is such that $F(x) \in [-\infty, +\infty)$ with $F(x^\star) \in \mathbb{R}$.

Then, for any $i \in \mathsf{S}$, for any $x_i \in \mathsf{X}_i^0$, $\varphi_i(x_i) = \lim_{n \to \infty} \varphi_i^n(x_i)$ exists and is such that $\varphi_i(x_i) \in [-\infty, +\infty)$.

Proof. Consider $\mathsf{A}^0 \subset (\mathbb{R}^d)^{\ell+1}$ such that assumptions (a), (b) and (c) hold. Remark that we have $F^n(x^\star) = \varphi_{i_{K-1}}^n(x_{i_{K-1}}^\star)$.

Let $x \in \mathsf{A}^0$. We denote $x^m = (x_1^\star, \dots, x_{m-1}^\star, x_m, x_{m+1}^\star, \dots, x_l^\star)$ for any $m \in \{0, \dots, \ell\}$. In particular, we have $x^m \in \mathsf{A}^0$ by assumption (a). Let us define

$$\begin{aligned} \varphi_{i_k}(x_{i_k}) &= F(x^{i_k}) - F(x^\star), \quad \forall k \in \{0, \dots, K-2\}, \\ \varphi_{i_{K-1}}(x_{i_{K-1}}) &= F(x^{i_{K-1}}). \end{aligned}$$

Using assumption (c), we have $\varphi_i(x_i) \in [-\infty, +\infty)$ for any $i \in \mathsf{S}$. Let $k \in \{0, \dots, K-2\}$. We have by definition of F^n,

$$\varphi_{i_k}^n(x_{i_k}) = F^n(x^{i_k}) - \sum_{\substack{m=0 \\ m \neq k}}^{K-1} \varphi_{i_m}^n(x_{i_m}^\star) = F^n(x^{i_k}) - F^n(x^\star),$$

where we used assumption (b) in the last equality. Since $x^{i_k} \in \mathsf{A}^0$ and $x^\star \in \mathsf{A}^0$, we have by assumption (c),

$$\lim_{n \to \infty} \varphi_{i_k}^n(x_{i_k}) = F(x^{i_k}) - F(x^\star) = \varphi_{i_k}(x_{i_k}).$$

Furthermore, by combining the definition of F^n with assumption (b), we have

$$\lim_{n \to \infty} \varphi_{i_{K-1}}^n(x_{i_{K-1}}) = F(x^{i_{K-1}}) = \varphi_{i_{K-1}}(x_{i_{K-1}}),$$

which concludes the proof. □

In what follows, before proving Proposition 13, we respectively show in Proposition 120 and Proposition 121 how **A**7 and **A**8 can be satisfied in the case where $\pi^0 \in \mathscr{P}_{\mathsf{T}_r}$, as in (7.4), that is

$$\pi^0 = \pi_r^0 \bigotimes_{(v,v') \in \mathsf{E}_r} \pi_{v'|v}^0. \tag{F.12}$$

Proposition 120. *Let $\pi^0 \in \mathscr{P}_{\mathsf{T}_r}$. Assume that $\pi^0_r = \mu_r$ if $r \in \mathsf{S}$ or $\pi^0_r = \mathrm{N}(m_r, \sigma_r \mathrm{Id})$, with $m_r \in \mathbb{R}^d$ and $\sigma_r > 0$ if $r \in \mathsf{S}^{\mathrm{c}}$. In addition, assume that for any $(v, v') \in \mathsf{E}_r$, $\pi^0_{v'|v}(\cdot|x_v) = \mathrm{N}(x_v, \sigma_{v,v'}\mathrm{Id})$ with $\sigma_{v,v'} > 0$. Finally, assume that for any $i \in \mathsf{S}$, $\int_{\mathbb{R}^d} \|x\|^2 \mathrm{d}\mu_i(x) < +\infty$ and $\mathrm{H}(\mu_i) < +\infty$. Then **A**7 is satisfied.*

Proof. Let $\pi = \otimes_{i\in\mathsf{S}}\mu_i \otimes_{i\in\mathsf{S}^{\mathrm{c}}} \nu_i$ with ν_i any Gaussian measure with positive definite covariance matrix. First, we have that

$$\mathrm{KL}(\pi \mid \pi^0) = \mathrm{KL}(\pi_r \mid \pi^0_r) + \textstyle\sum_{(v,v')\in\mathsf{E}_r} \int_{\mathbb{R}^d} \mathrm{KL}(\pi_{v'|v}|\pi^0_{v'|v})\mathrm{d}\pi_v \ . \tag{F.13}$$

For any $(v, v') \in \mathsf{E}_r$, there exists $C_{v,v'} \geq 0$ such that

$$\begin{aligned}\textstyle\int_{\mathbb{R}^d} \mathrm{KL}(\pi_{v'|v}|\pi^0_{v'|v})\mathrm{d}\pi_v &\leq C_{v,v'} - \mathrm{H}(\pi_{v'}) + \textstyle\int_{\mathbb{R}^d\times\mathbb{R}^d} \|x_v - x_{v'}\|^2/(2\sigma^2_{v,v'})\mathrm{d}\pi_v \otimes \pi_{v'}(x_v, x_{v'}) \\ &\leq C_{v,v'} - \mathrm{H}(\pi_{v'}) + (1/\sigma^2_{v,v'}) \textstyle\int_{\mathbb{R}^d} \|x_v\|^2 \mathrm{d}\pi_v(x_v) + (1/\sigma^2_{v,v'}) \int_{\mathbb{R}^d} \|x_{v'}\|^2 \mathrm{d}\pi_{v'}(x_{v'}) < +\infty.\end{aligned}$$

We conclude the proof upon remarking that $\mathrm{KL}(\pi_r \mid \pi^0_r) < +\infty$. □

Proposition 121. *Let $\pi^0 \in \mathscr{P}_{\mathsf{T}_r}$. Assume that $\pi^0_r = \mu_r$ if $r \in \mathsf{S}$ or $\pi^0_r = \mathrm{N}(m_r, \sigma_r \mathrm{Id})$, with $m_r \in \mathbb{R}^d$ and $\sigma_r > 0$ if $r \in \mathsf{S}^{\mathrm{c}}$. In addition, assume that for any $(v, v') \in \mathsf{E}_r$, $\pi^0_{v'|v}(\cdot|x_v) = \mathrm{N}(x_v, \sigma_{v,v'}\mathrm{Id})$ with $\sigma_{v,v'} > 0$. Finally, assume that for any $i \in \mathsf{S}$, μ_i admits a positive density w.r.t. the Lebesgue measure. Then **A**8 is satisfied.*

Proof. We have that π^0 admits a positive density w.r.t the Lebesgue measure. Letting $\tilde{\pi}^0 = \bigotimes_{i\in\mathsf{S}} \mu_i \bigotimes_{j\in\mathsf{S}^{\mathrm{c}}} \tilde{\mu}_j$ where $\tilde{\mu}_j$ which admits a positive density w.r.t. the Lebesgue measure for any $j \in \mathsf{S}^{\mathrm{c}}$, we get that $\tilde{\pi}^0$ admits a positive density w.r.t. the Lebesgue measure and therefore $\pi^0 \sim \tilde{\pi}^0$, which concludes the proof. □

Using the preliminary results presented above, we are now ready to prove Proposition 13.

Proof of Proposition 13. Assume **A**6 and **A**7. Since $\mathscr{P}_{\mathsf{S}}$ is convex and closed in total-variation norm, there exists a probability distribution $\pi^\star$ solution to (F.10), or equivalently to (static-mSB), by using **A**7 with [Csiszár, 1975, Theorem 2.1.]. Moreover, this solution is unique by strict convexity of $\mathrm{KL}(\cdot \mid \pi^0)$.

We now turn to the proof of existence of potentials defining $(\mathrm{d}\pi^\star/\mathrm{d}\pi^0)$, by adapting the arguments of [Nutz, 2021, Section 2.3.]. Define $\nu^n = \arg\min\{\mathrm{KL}(\pi \mid \pi^0) : \pi \in \mathscr{P}^n_{\mathsf{S}}\}$ for any $n \in \mathbb{N}^*$. Since $\{\mathscr{P}^n_{\mathsf{S}}\}_{n\in\mathbb{N}^*} \subset \mathscr{P}^{(\ell+1)}$ is a decreasing sequence of sets that are convex and closed in total-variation norm such that (F.9) holds, we get from [Nutz, 2021, Proposition 1.17.] with **A**7 that

$$\lim_{n\to\infty} \|\nu^n - \pi^\star\|_{\mathrm{TV}} = 0 \ ,$$

or equivalently

$$\lim_{n\to\infty} \|(\mathrm{d}\nu^n/\mathrm{d}\pi^0) - (\mathrm{d}\pi^\star/\mathrm{d}\pi^0)\|_{\mathrm{L}^1(\pi^0)} = 0 \ . \tag{F.14}$$

Following [Nutz, 2021, Example 1.18], there exists a family of bounded measurable functions $\{\varphi^n_i\}_{n\in\mathbb{N}^*, i\in\mathsf{S}}$ with $\varphi^n_i : \mathbb{R}^d \to \mathbb{R}$ such that for any $n \in \mathbb{N}^*$

$$(\mathrm{d}\nu^n/\mathrm{d}\pi^0) = \exp[\textstyle\bigoplus_{i\in\mathsf{S}} \varphi^n_i] \ . \tag{F.15}$$

We consider such family $\{\varphi^n_i\}_{n\in\mathbb{N}^*, i\in\mathsf{S}}$ for the rest of the proof. By combining (F.14) and (F.15), we obtain, up to extraction,

$$(\mathrm{d}\pi^\star/\mathrm{d}\pi^0) = \lim_{n\to\infty} \exp[\textstyle\bigoplus_{i\in\mathsf{S}} \varphi^n_i] \quad \pi^0\text{-a.s.} \ . \tag{F.16}$$

We now define the following sets

$$\begin{aligned}\mathsf{A}^\star &= \{x \in (\mathbb{R}^d)^{\ell+1} : \lim_{n\to\infty} \textstyle\bigoplus_{i\in\mathsf{S}} \varphi^n_i(x_i) \in [-\infty, +\infty)\} \ , \\ \mathsf{B}^\star &= \{x \in (\mathbb{R}^d)^{\ell+1} : \lim_{n\to\infty} \textstyle\bigoplus_{i\in\mathsf{S}} \varphi^n_i(x_i) > -\infty\} \subset \mathsf{A}^\star\end{aligned}$$

Using (F.16), we have $\pi^0(\mathsf{A}^\star) = 1$. Using **A**8, it comes $\tilde{\pi}^0(\mathsf{A}^\star) = 1$. Moreover, we also get that $\pi^\star(\mathsf{B}^\star) = 1$ by (F.16). Thus, it comes $\pi^0(\mathsf{B}^\star) > 0$, and $\tilde{\pi}^0(\mathsf{B}^\star) > 0$ using **A**8.

We then apply Lemma 118 to $\tilde{\pi}^0$ and $\mathsf{A} = \mathsf{A}^\star$. Since $\tilde{\pi}^0(\mathsf{B}^\star) > 0$, it implies that there exists $x^\star \in \mathsf{B}^\star$ and a measurable set $\mathsf{A}^0 \subset \mathsf{B}^\star$ verifying the properties (a) and (b). Following [Nutz, 2021, Corollary 2.12], we may assume without loss of generality in the statement of Lemma 118 that the sets X^0_m are measurable with $\prod_{m=0}^{\ell} \mathsf{X}^0_m \subset \mathsf{A}$. In this case, we obtain that $\mu_i(\mathrm{proj}_i(\mathsf{A}^0)) = 1$ for any $i \in \mathsf{S}$.

We now aim at applying Lemma 119 to the set A^0. Remark that A^0 directly satisfies assumption (a). For any $n \in \mathbb{N}^*$, consider the following transformation of the functions $\{\varphi_i^n\}_{i\in\mathsf{S}}$

$$\begin{aligned} \varphi^n_{i_k} &\leftarrow \varphi^n_{i_k} - \varphi^n_{i_k}(x^\star_{i_k}), \quad \forall k \in \{0, \dots, K-2\} , \\ \varphi^n_{i_{K-1}} &\leftarrow \varphi^n_{i_{K-1}} + \textstyle\sum_{k=0}^{K-2} \varphi^n_{i_k}(x^\star_{i_k}) . \end{aligned}$$

For any $i \in \mathsf{S}$, we restrict $\varphi^n_{i_k}$ to $\mathsf{X}^0_{i_k}$, so that the family $\{\varphi^n_i\}_{n\in\mathbb{N}^*, i\in\mathsf{S}}$ now verifies assumption (b). Finally, since $\mathsf{A}^0 \subset \mathsf{A}^\star$ and $x^\star \in \mathsf{B}^\star$, we directly obtain assumption (c).

Therefore, Lemma 119 may be applied. It provides us with the family of functions $\{\varphi_i\}_{i\in\mathsf{S}}$ defined by $\varphi_i : \mathsf{X}^0_i \to [-\infty, +\infty)$ with $\varphi_i = \lim_{n\to\infty} \varphi^n_i$ μ_i-a.s. for any $i \in \mathsf{S}$. Since $\mu_i(\mathrm{proj}_i(\mathsf{A}^0)) = 1$ for any $i \in \mathsf{S}$, we may extend the functions φ_i to $\mathbb{R}^d$. In particular, we can find a family of functions $\{\psi^\star_i\}_{i\in\mathsf{S}}$ with $\psi^\star_i : \mathbb{R}^d \to [-\infty, +\infty)$ such that $\psi^\star_i = \varphi_i$ μ_i-a.s. Note that these functions are measurable as limits of measurable functions.

Since $\pi^0 \sim \tilde{\pi}^0$ by **A**8, (F.16) turns into

$$(\mathrm{d}\pi^\star/\mathrm{d}\pi^0) = \exp[\textstyle\bigoplus_{i\in\mathsf{S}} \psi^\star_i] \quad \pi^0\text{-a.s.} . \tag{F.17}$$

Finally, we show that the functions $\psi^\star_i$ are μ_i-a.s. finite. Let $i \in \mathsf{S}$. Let us define $\mathsf{A}_i = \{x_i \in \mathbb{R}^d : \psi^\star_i(x_i) = -\infty\}$. Using (F.17), we obtain $(\mathrm{d}\pi^\star/\mathrm{d}\pi^0)(\mathsf{A}_i \times (\mathbb{R}^d)^\ell) = 0$. Since $\pi^\star_i = \mu_i$, we have

$$\mu_i(\mathsf{A}_i) = \pi^\star(\mathsf{A}_i \times (\mathbb{R}^d)^\ell) = \textstyle\int_{\mathsf{A}_i\times(\mathbb{R}^d)^\ell} (\mathrm{d}\pi^\star/\mathrm{d}\pi^0)\mathrm{d}\pi^0 = 0 ,$$

which gives the result. □

We now turn to the proof of Corollary 14, which states that the iterates of (mIPF) can be expressed via potentials, in the same manner as the solution $\pi^\star$ to (static-mSB).

Proof of Corollary 14. Assume **A**6, **A**7 and **A**8. We prove the result of this corollary by recursion on $n \in \mathbb{N}^*$. First take $n = 1$. In this case, the first iteration of (mIPF) is a multi-marginal SB problem of the form (static-mSB) where $S = \{i_0\}$ with reference measure π^0. Therefore, using **A**7 and **A**8, we can apply Proposition 13 and obtain existence of $\psi^1_{i_0} : \mathbb{R}^d \to \mathbb{R}$ such that

$$(\mathrm{d}\pi^1/\mathrm{d}\pi^0) = \exp[\psi^1_{i_0}] \quad \pi^0\text{-a.s} .$$

By taking $\psi^0_{i_k} = 0$ for $k \in \{1, \dots, K-1\}$, we thus obtain the result at step $n = 1$.

Now assume that the result is verified for some $n \in \mathbb{N}^*$, with $k_n = (n-1) \bmod(K)$. We define $k_n + 1 = n \bmod(K)$ and $q_n \in \mathbb{N}$ as the quotient of the Euclidean division of n by K. In this case, the $(n+1)$-th iteration of (mIPF) is a multi-marginal SB problem of the form (static-mSB) where $\mathsf{S} = \{i_{k_n+1}\}$ with reference measure π^n. Using (F.19), we have that **A**7 is satisfied for this new (static-mSB) problem. **A**6 and **A**8 are satisfied for this problem, given the form of π^n. Therefore, we can apply Proposition 13 and obtain existence of $\psi^{q_n+1}_{i_{k_n+1}} : \mathbb{R}^d \to \mathbb{R}$ such that

$$(\mathrm{d}\pi^{n+1}/\mathrm{d}\pi^n) = \exp[\psi^{q_n+1}_{i_{k_n+1}}] \quad \pi^n\text{-a.s} . \tag{F.18}$$

By assumption, we have that $\pi^n \ll \pi^0$. Hence, we obtain $\pi^{n+1} \ll \pi^0$ and thus,

$$(\mathrm{d}\pi^{n+1}/\mathrm{d}\pi^0) = (\mathrm{d}\pi^{n+1}/\mathrm{d}\pi^n)(\mathrm{d}\pi^n/\mathrm{d}\pi^0) \quad \pi^0\text{-a.s} .$$

By combining (F.18) with the result of the recursion at step n, we directly obtain the result at step $n+1$, which achieves the proof. □

Proofs of Proposition 15 and Proposition 16. In this part of the section, we establish the proofs of results related to the convergence of (mIPF), respectively Proposition 15 and Proposition 16, which can be seen as a natural extension of [Ruschendorf, 1995, Proposition 2.1.] and [Ruschendorf, 1995, Theorem 3.1.].

Proof of Proposition 15. Under **A**6 and **A**7, we obtain by Proposition 13 existence and uniqueness of a solution to (static-mSB), which we denote by $\pi^\star$. Since $\pi^\star \in \mathscr{P}_{\mathsf{S}}$, using recursively [Csiszár, 1975, Theorem 3.12.], the fact that $\{\pi_{i_k} = \mu_{i_k} \,:\, \pi \in \mathscr{P}^{(|\mathsf{V}|)}\}$ is convex for any $k \in \{0, \dots, K-1\}$ and (mIPF), we obtain

$$\mathrm{KL}(\pi^\star \mid \pi^0) = \mathrm{KL}(\pi^\star \mid \pi^n) + \textstyle\sum_{i=1}^n \mathrm{KL}(\pi^i \mid \pi^{i-1}) \,. \tag{F.19}$$

Therefore, we have $\sum_{i=1}^\infty \mathrm{KL}(\pi^i \mid \pi^{i-1}) \leq \mathrm{KL}(\pi^\star \mid \pi^0) < \infty$ and thus,

$$\lim_{i \to +\infty} \mathrm{KL}(\pi^i \mid \pi^{i-1}) = 0 \,. \tag{F.20}$$

Let $n \in \mathbb{N}^*$ with $n > 2K$, $k \in \{0, \dots, K-1\}$ and let $q_n \in \mathbb{N}$ be the quotient of the Euclidean division of $n-1$ by K. We define $n_k = q_n K + k + 1$ with $(n_k - 1) = k \mod(K)$ if $n_k \leq n$. Otherwise, we set $n_k = (q_n - 1)K + k + 1$ with $(n_k - 1) = k \mod(K)$. Note that we always have $|n - n_k| \leq 2K$. In particular, we have $\pi^{n_k}_{i_k} = \mu_{i_k}$ by definition of (mIPF). Therefore, we obtain

$$\begin{aligned}
\|\pi^n_{i_k} - \mu_{i_k}\|_{\mathrm{TV}} &\leq \|\pi^n - \pi^{n_k}\|_{\mathrm{TV}} \\
&\leq \|\pi^n - \pi^{n-1}\|_{\mathrm{TV}} + \ldots + \|\pi^{n_k+1} - \pi^{n_k}\|_{\mathrm{TV}} && \text{(triangle inequality)} \\
&\leq (2\mathrm{KL}(\pi^n \mid \pi^{n-1}))^{1/2} + \ldots + (2\mathrm{KL}(\pi^{n_k+1} \mid \pi^{n_k}))^{1/2} \,, && \text{(Pinsker's inequality)}
\end{aligned}$$

where each term goes to 0 as $n \to +\infty$ in the last inequality by (F.20), which achieves the proof. □

We now turn to the proof Proposition 16, which requires several preliminary technical results. For the rest of this section, we define, for any $n \in \mathbb{N}$, q_n as the quotient of the Euclidean division of $n-1$ by K (in particular, $q_0 = -1$).

Schrödinger equations. Under **A**6, **A**7 and **A**8, we know from Proposition 13 that the unique solution $\pi^\star$ to (static-mSB) can be π^0-a.s. written as $(\mathrm{d}\pi^\star/\mathrm{d}\pi^0) = \exp[\bigoplus_{i \in \mathsf{S}} \psi^\star_i]$, where $\{\psi^\star_i\}_{i \in \mathsf{S}}$ are measurable potentials, referred to as *Schrödinger potentials*. These functions are determined by the fixed-point *Schrödinger equations*

$$\psi^\star_i(x_i) = \log[r_i(x_i) / \textstyle\int_{(\mathbb{R}^d)^\ell} \exp[\sum_{j \in \mathsf{S} \setminus \{i\}} \psi^\star_j(x_j)] h(x_{0:\ell}) \mathrm{d}\nu_{-i}(x_{-i})] \quad \mu_i\text{-a.s.}, \quad \forall i \in \mathsf{S} \,,$$

which are obtained by marginalising $\pi^\star$ along its constrained marginals. This family of potentials is not unique. Indeed, for any family of real numbers $\{\lambda_{i_k}\}_{k \in \{0, \dots, K-2\}}$, we have

$$(\mathrm{d}\pi^\star/\mathrm{d}\pi^0) = \exp[\textstyle\bigoplus_{i \in \mathsf{S}} \tilde{\psi}_i] \,,$$

where $\tilde{\psi}_{i_k} = \psi^\star_{i_k} + \tilde{\lambda}_{i_k}$ for any $k \in \{0, \dots, K-1\}$ with $\tilde{\lambda}_{i_k} = \lambda_{i_k}$ if $k \in \{0, \dots, K-2\}$ and $\tilde{\lambda}_{i_{K-1}} = -\sum_{i=0}^{K-2} \lambda_{i_k}$.

Remark on the initialisation of (mIPF). Consider a probability measure $\bar{\pi}^0 \in \mathscr{P}^{(\ell+1)}$ of the form

$$(\mathrm{d}\bar{\pi}^0/\mathrm{d}\pi^0) = \exp[\textstyle\bigoplus_{i \in \mathsf{S}} \psi^0_i] \,, \tag{F.21}$$

where $\{\psi^0_i\}_{i \in \mathsf{S}}$ is a family of measurable potentials with $\psi^0_i : \mathbb{R}^d \to \mathbb{R}$ such that $\left|\int_{\mathbb{R}^d} \psi^0_i \mathrm{d}\mu_i\right| < \infty$ for any $i \in \mathsf{S}$. Then, for any $\pi \in \mathscr{P}_{\mathsf{S}}$, we have

$$\mathrm{KL}(\pi \mid \pi^0) = \mathrm{KL}(\pi \mid \bar{\pi}^0) + \textstyle\int_{(\mathbb{R}^d)^K} \bigoplus_{i \in \mathsf{S}} \psi^0_i \mathrm{d}\pi = \mathrm{KL}(\pi \mid \bar{\pi}^0) + \sum_{i \in \mathsf{S}} \int_{\mathbb{R}^d} \psi^0_i \mathrm{d}\mu_i \,. \tag{F.22}$$

Hence, (static-mSB) is equivalent to the multi-marginal SB problem

$$\arg\min\{\mathrm{KL}(\pi|\bar{\pi}^0) \,:\, \pi \in \mathscr{P}^{(\ell+1)}, \; \pi_i = \mu_i \,, \forall i \in \mathsf{S}\} \,.$$

We refer to [Peyré et al., 2019, Proposition 4.2] for the EOT counterpart of this result. This means that the solutions of the multi-marginal Schrödinger Bridge problem are invariant by multiplication of the reference measure by potentials on the *fixed* marginals. Consequently, the initialisation of the (mIPF) sequence may be chosen as $\bar{\pi}^0$ instead of π^0.

For sake of clarity, we now refer to the reference probability measure of (static-mSB) as $\bar{\pi}$ or π^{-1} and to the initialisation of the (mIPF) iterates as π^0.

Solving (mIPF) with potentials. To prove the convergence of the (mIPF) iterates to the solution $\pi^\star$ given by Proposition 13, we first rewrite these iterates with potentials, following the form of $\pi^\star$.

To do so, we recursively define the sequence of potentials $\{\psi_i^n\}_{n\in\mathbb{N},i\in\mathsf{S}}$ by

$$\psi_{i_0}^0 = \ldots = \psi_{i_{K-2}}^0 = 0\,, \tag{F.23}$$
$$\psi_{i_{K-1}}^0(x_{i_{K-1}}) = \log(r_{i_{K-1}}(x_{i_{K-1}})/\textstyle\int_{(\mathbb{R}^d)^\ell} h(x_{0:\ell})\mathrm{d}\nu_{-i_{K-1}}(x_{-i_{K-1}}))\,,$$

and for any $n\in\mathbb{N}^*$ and $k\in\{0,\ldots,K-1\}$

$$\psi_{i_k}^{q_n+1}(x_{i_k}) = \log[r_{i_k}(x_{i_k})/\textstyle\int_{(\mathbb{R}^d)^\ell}\exp[\bigoplus_{\ell=0}^k \psi_{i_\ell}^{q_n+1}(x_{i_\ell})\bigoplus_{m=k+1}^{K-1}\psi_{i_m}^{q_n}(x_{i_m})] \tag{F.24}$$
$$\times\, h(x_{0:\ell})\mathrm{d}\nu_{-i_k}(x_{-i_k})]\,, \tag{F.25}$$

recalling that q_n is the quotient of the Euclidean division of $n-1$ by K.

We now define the sequence of probability measures $\{\pi^n\}_{n\in\mathbb{N}}$ by

$$\mathrm{d}\pi^n/\mathrm{d}\bar\pi = \exp[\textstyle\bigoplus_{\ell=0}^{k_n}\psi_{i_\ell}^{q_n+1}\bigoplus_{m=k_n+1}^{K-1}\psi_{i_m}^{q_n}],\ k_n = (n-1)\bmod(K),\ n = q_nK+k_n+1. \tag{F.26}$$

In particular, we have $(\mathrm{d}\pi^0/\mathrm{d}\bar\pi) = \exp[\oplus_{\ell=0}^{K-1}\psi_{i_\ell}^0] = \exp[\psi_{i_{K-1}}^0]$, and thus $\int_{\mathbb{R}^d}\psi_{i_{K-1}}^0\mathrm{d}\mu_{i_{K-1}} = \mathrm{KL}(\mu_{i_{K-1}}\mid\bar\pi_{i_{K-1}})$. Consequently, π^0 can be chosen as the initialisation of (mIPF), following the previous remark, if we assume that $\mathrm{KL}(\mu_{i_{K-1}}\mid\bar\pi_{i_{K-1}})<\infty$. In (TreeSB) with $r=i_{K-1}$, the latter assumption is directly verified since we choose $\bar\pi_{i_{K-1}} = \mu_{i_{K-1}}$.

Let $n\in\mathbb{N}$, with $k_n = (n-1)\bmod(K)$, $k_n+1 = n\bmod(K)$. Using (F.23) and (F.25), we get that $\pi_{i_{k_n}}^n = \mu_{i_{k_n}}$. Moreover, we have

$$\mathrm{d}\pi^n/\mathrm{d}\pi^{n-1} = \exp[\psi_{i_{k_n}}^{q_n+1} - \psi_{i_{k_n}}^{q_n}]\,, \tag{F.27}$$

with the convention that $\psi_{i_{K-1}}^{-1} = 0$. In particular, we obtain that $\pi_{|i_{k_n+1}}^{n+1} = \pi_{|i_{k_n+1}}^n$.

In conclusion, the sequence $\{\pi^n\}_{n\in\mathbb{N}}$ defined in (F.26) verifies $\pi^{n+1} = \mu_{i_{k_n+1}}\pi_{|i_{k_n+1}}^n$ for any $n\in\mathbb{N}$. By decomposition property of the Kullback-Leibler divergence, this sequence solves (mIPF) with initialisation π^0. *We consider such iterates in the following.*

Since $\pi_{i_{k_n}}^n = \mu_{i_{k_n}}$, we have that

$$\mathrm{KL}(\pi^n\mid\pi^{n-1}) = \textstyle\int_{\mathbb{R}^d}(\psi_{i_{k_n}}^{q_n+1} - \psi_{i_{k_n}}^{q_n})\mathrm{d}\mu_{i_{k_n}}\,. \tag{F.28}$$

Before proving a multi-marginal counterpart to [Ruschendorf, 1995, Lemma 4.1], we state and prove the following result.

Proposition 122. *Let π_0,π_1 two probability measures on $\mathbb{R}^d$ such that $\pi_0\ll\pi_1$. Then, denoting $f = \mathrm{d}\pi_0/\mathrm{d}\pi_1$, the following assertions are equivalent:*

(a) $\mathrm{KL}(\pi_0\mid\pi_1)<+\infty$

(b) $\int_{\mathbb{R}^d}|\log(f)(x)|\mathrm{d}\pi_0(x)<+\infty$

(c) $\int_{\mathbb{R}^d}\log(f)(x)\mathbb{1}_{f(x)>1}\mathrm{d}\pi_0(x)<+\infty$

If one of these conditions is satisfied then $\int_{\mathbb{R}^d}|\log(f)(x)|\,\mathrm{d}\pi_0\le\mathrm{KL}(\pi_0\mid\pi_1)+2/\mathrm{e}$.

Proof. First, note that

$$\textstyle\int_{\mathbb{R}^d}|\log(f)(x)|\mathbb{1}_{f<1}\mathrm{d}\pi_0(x)\le\int_{\mathbb{R}^d}|\log(f)(x)f(x)|\mathbb{1}_{f<1}\mathrm{d}\pi_1(x)\le 1/\mathrm{e}\,, \tag{F.29}$$

where we have used that for any $u\in[0,1]$, $|u\log(u)|\le 1/\mathrm{e}$. We have that (b) implies (c). Using the previous result we have that (c) implies (b). Hence (c) and (b) are equivalent. In addition, it is clear that (b) implies (a). Finally (this is more of a convention), we have that $\mathrm{KL}(\pi_0\mid\pi_1) = \int_{\mathbb{R}^d}\log(f)(x)\mathbb{1}_{f(x)>1}\mathrm{d}\pi_0(x) + \int_{\mathbb{R}^d}\log(f)(x)\mathbb{1}_{f(x)<1}\mathrm{d}\pi_0(x)<+\infty$. Using (F.29) this implies (c). Finally, we have

$$\begin{aligned}\textstyle\int_{\mathbb{R}^d}|\log(f)(x)|\,\mathrm{d}\pi_0(x) &= \textstyle\int_{\mathbb{R}^d}\log(f)(x)\mathrm{d}\pi_0(x) - 2\int_{\mathbb{R}^d}\log(f)(x)\mathbb{1}_{f(x)<1}\mathrm{d}\pi_0(x)\\ &\le \mathrm{KL}(\pi_0\mid\pi_1)+2/\mathrm{e},\end{aligned}$$

which concludes the proof. □

We begin with the following lemma which controls the integral of the potentials uniformly w.r.t. $n \in \mathbb{N}$. It can be seen as the *multi-marginal* counterpart of [Ruschendorf, 1995, Lemma 4.1].

Lemma 123. *Assume* **A**9. *There exist* $\{c_i\}_{i\in\mathsf{S}} \in (0,+\infty)^K$ *such that for any function* $f : (\mathbb{R}^d)^{\ell+1} \to \mathbb{R}$ *of the form* $f = \bigoplus_{i\in\mathsf{S}} f_i$, *we have*

$$c_i \|f\|_{\mathrm{L}^1(\pi^\star)} \geq \|f_i\|_{\mathrm{L}^1(\mu_i)}, \quad \forall i \in \mathsf{S} \,. \tag{F.30}$$

For any $n \in \mathbb{N}^\star$, *we have*

(a) $\sum_{i\in\mathsf{S}} \int_{\mathbb{R}^d} \psi_i^n \mathrm{d}\mu_i \leq \mathrm{KL}(\pi^\star \mid \bar{\pi}) < \infty$,

(b) $\int_{(\mathbb{R}^d)^{\ell+1}} (\bigoplus_{i\in\mathsf{S}} \psi_i^\star - \bigoplus_{i\in\mathsf{S}} \psi_i^n) \mathrm{d}\pi^\star \leq \mathrm{KL}(\pi^\star \mid \bar{\pi}) < \infty$,

(c) $\sup_{n\in\mathbb{N}} \int_{\mathbb{R}^d} |\psi_i^n| \, \mathrm{d}\mu_i < \infty$, $\forall i \in \mathsf{S}$.

Proof. First, we have that (F.30) is a direct consequence of [Kober, 1940, Theorem 1] and **A**9. Let us now prove (a). Using (F.28), we have

$$\begin{aligned}
\sum_{m=0}^{Kn} \mathrm{KL}(\pi^m \mid \pi^{m-1}) &= \sum_{\ell=0}^{n-1} \sum_{k=0}^{K-1} \mathrm{KL}(\pi^{\ell K+k+1} \mid \pi^{\ell K+k}) + \mathrm{KL}(\pi^0 \mid \pi^{-1}) \\
&= \sum_{\ell=0}^{n-1} \sum_{i\in\mathsf{S}} \int_{\mathbb{R}^d} (\psi_i^{\ell+1} - \psi_i^\ell) \mathrm{d}\mu_i + \int_{\mathbb{R}^d} (\psi_{i_{K-1}}^0 - \psi_{i_{K-1}}^{-1}) \mathrm{d}\mu_{i_{K-1}} \\
&= \sum_{i\in\mathsf{S}} \sum_{\ell=0}^{n-1} \int_{\mathbb{R}^d} (\psi_i^{\ell+1} - \psi_i^\ell) \mathrm{d}\mu_i + \int_{\mathbb{R}^d} (\psi_{i_{K-1}}^0 - \psi_{i_{K-1}}^{-1}) \mathrm{d}\mu_{i_{K-1}} \\
&= \sum_{i\in\mathsf{S}} \int_{\mathbb{R}^d} (\psi_i^n - \psi_i^0) \mathrm{d}\mu_i + \int_{\mathbb{R}^d} (\psi_{i_{K-1}}^0 - \psi_{i_{K-1}}^{-1}) \mathrm{d}\mu_{i_{K-1}} \\
&= \sum_{i\in\mathsf{S}} \int_{\mathbb{R}^d} \psi_i^n \mathrm{d}\mu_i \leq \mathrm{KL}(\pi^\star \mid \bar{\pi}). \,,
\end{aligned}$$

where the last inequality follows the proof of Proposition 15.

Since the first term in the inequality of (b) is equal to $\mathrm{KL}(\pi^\star \mid \pi^{nK})$, we obtain (b) using that $\mathrm{KL}(\pi^\star \mid \pi^{nK}) \leq \mathrm{KL}(\pi^\star \mid \bar{\pi})$ following the proof of Proposition 15.

Let us now prove (c). Since $\mathrm{KL}(\pi^\star \mid \bar{\pi}) < \infty$, using Proposition 122, we have that $\bigoplus_{i\in\mathsf{S}} \psi_i^\star \in \mathrm{L}^1(\pi^\star)$. From (b) and Proposition 122, we also get that $\bigoplus_{i\in\mathsf{S}} (\psi_i^\star - \psi_i^n) \in \mathrm{L}^1(\pi^\star)$, and thus $\int_{(\mathbb{R}^d)^{\ell+1}} \left|\bigoplus_{i\in\mathsf{S}} (\psi_i^\star - \psi_i^n)\right| \mathrm{d}\pi^\star \leq C_0$ with $C_0 > 0$. Therefore, we have

$$\int_{(\mathbb{R}^d)^{\ell+1}} \left|\bigoplus_{i\in\mathsf{S}} \psi_i^n\right| \mathrm{d}\pi^\star \leq \int_{(\mathbb{R}^d)^{\ell+1}} \left|\bigoplus_{i\in\mathsf{S}} \psi_i^\star\right| \mathrm{d}\pi^\star + \int_{(\mathbb{R}^d)^{\ell+1}} \left|\bigoplus_{i\in\mathsf{S}} (\psi_i^\star - \psi_i^n)\right| \mathrm{d}\pi^\star \leq 2C_0 \,.$$

Using (F.30), we conclude with **A**9 that for any $i \in \mathsf{S}$, we have

$$\int_{\mathbb{R}^d} |\psi_i^n| \, \mathrm{d}\mu_i \leq 2c_i C_0 \,,$$

which concludes the proof of (c). □

The next lemma gives an explicit expression for $\mathrm{KL}(\pi^n \mid \bar{\pi})$. It can be seen as the *multi-marginal* counterpart of [Ruschendorf, 1995, Lemma 4.2].

Lemma 124. *For any* $n \in \mathbb{N}$, *with* $k_n = (n-1) \bmod (K)$, *we have*

$$\begin{aligned}
\mathrm{KL}(\pi^n \mid \bar{\pi}) = \int_{\mathbb{R}^d} \psi_{i_{k_n}}^{q_n+1} \mathrm{d}\mu_{i_{k_n}} &+ \sum_{\ell=0}^{k_n-1} \int_{\mathbb{R}^d} \psi_{i_\ell}^{q_n+1} \exp[\psi_{i_\ell}^{q_n+1} - \psi_{i_\ell}^{q_n+2}] \mathrm{d}\mu_{i_\ell} \\
&+ \sum_{m=k_n+1}^{K-1} \int_{\mathbb{R}^d} \psi_{i_m}^{q_n} \exp[\psi_{i_m}^{q_n} - \psi_{i_m}^{q_n+1}] \mathrm{d}\mu_{i_m} \,.
\end{aligned}$$

Proof. Let $n \in \mathbb{N}$, with $k_n = (n-1) \bmod (K)$. Using (F.26), we have

$$\mathrm{KL}(\pi^n \mid \bar{\pi}) = \int_{\mathbb{R}^d} \psi_{i_{k_n}}^{q_n+1} \mathrm{d}\mu_{i_{k_n}} + \sum_{\ell=0}^{k_n-1} \int_{\mathbb{R}^d} \psi_{i_\ell}^{q_n+1} \mathrm{d}\pi_{i_\ell}^n + \sum_{m=k_n+1}^{K-1} \int_{\mathbb{R}^d} \psi_{i_m}^{q_n} \mathrm{d}\pi_{i_m}^n \,. \tag{F.31}$$

Consider $m \in \{k_n+1, \dots, K-1\}$. Let m_n be the closest integer to n such that $m_n > n$ and $m = (m_n - 1) \bmod (K)$. By (F.27), we have

$$\mathrm{d}\pi^n = \exp[\textstyle\bigoplus_{j=k_n+1}^{m} \psi_{i_j}^{q_n} - \psi_{i_j}^{q_n+1}] \mathrm{d}\pi^{m_n}.$$

Using (F.27) recursively, we obtain

$$\mathrm{d}\pi^n_{i_m} = \exp[\psi^{q_n}_{i_m} - \psi^{q_n+1}_{i_m}]\mathrm{d}\pi^{m_n}_{i_m}, \tag{F.32}$$

where we recall that $\pi^{m_n}_{i_m} = \mu_{i_m}$.

Consider now $\ell \in \{0, \ldots, k_n - 1\}$. Let ℓ_n be the closest integer to n such that $\ell_n > n$ and $\ell = (\ell_n - 1) \bmod(K)$. By (F.27), we have

$$\mathrm{d}\pi^n = \exp[\bigoplus_{j=k_n+1}^{K-1}\{\psi^{q_n}_{i_j} - \psi^{q_n+1}_{i_j}\}\bigoplus_{j'=0}^{\ell}\{\psi^{q_n+1}_{i_{j'}} - \psi^{q_n+2}_{i_{j'}}\}]\mathrm{d}\pi^{\ell_n},$$

and using (F.27) recursively, we obtain

$$\mathrm{d}\pi^n_{i_\ell} = \exp[\psi^{q_n+1}_{i_\ell} - \psi^{q_n+2}_{i_\ell}]\mathrm{d}\pi^{\ell_n}_{i_\ell}, \tag{F.33}$$

where we recall that $\pi^{\ell_n}_{i_\ell} = \mu_{i_\ell}$. We conclude the proof upon combining (F.31), (F.32) and (F.33). □

We are now ready to prove a *uniform integrability* result which is the multi-marginal counterpart of [Ruschendorf, 1995, Lemma 4.4]. Before stating Lemma 126, we prove the following well-known lemma. We recall that a sequence $(\Psi_n)_{n\in\mathbb{N}}$ such that for any $n \in \mathbb{N}$, $\Psi_n \in \mathrm{L}^1(\mu)$, is *uniformly integrable* w.r.t. μ if (i) $\sup_{n\in\mathbb{N}} \int_{\mathbb{R}^d} |\Psi_n| \,\mathrm{d}\mu < +\infty$ and (ii) for any $\varepsilon > 0$, there exists $K > 0$ such that for any $n \in \mathbb{N}$, $\int_{\overline{\mathrm{B}}(0,K)^c} |\Psi_n| \,\mathrm{d}\mu \leq \varepsilon$.

Lemma 125. *Let $f : \mathbb{R} \to \mathbb{R}$, convex and non-decreasing on $[A, +\infty)$ with $A > 0$ and $\lim_{x\to+\infty} f(x)/x = +\infty$. Assume that $\sup_{n\in\mathbb{N}} \int_{\mathbb{R}^d} f(|\Psi_n|)\mathrm{d}\mu < +\infty$. Then, $(\Psi_n)_{n\in\mathbb{N}}$ is uniformly integrable w.r.t. μ.*

Proof. Since f is convex, using Jensen's inequality, we get that $\sup_{n\in\mathbb{N}} f(\int_{\mathbb{R}^d} |\Psi_n| \,\mathrm{d}\mu) < +\infty$ and since $\lim_{x\to+\infty} f(x)/x = +\infty$ we have $\sup_{n\in\mathbb{N}} \int_{\mathbb{R}^d} |\Psi_n| \,\mathrm{d}\mu < +\infty$. Let $\varepsilon > 0$, there exists $K > 0$ such that for any $x > K$, $x \leq \varepsilon f(x)/B$ with $B = \sup_{n\in\mathbb{N}} \int_{\mathbb{R}^d} f(|\Psi_n|)\mathrm{d}\mu < +\infty$. Therefore, we have for any $n \in \mathbb{N}$

$$\int_{\overline{\mathrm{B}}(0,K)^c} |\Psi_n| \,\mathrm{d}\mu \leq (\varepsilon/B) \int_{\overline{\mathrm{B}}(0,K)^c} f(|\Psi_n|)\mathrm{d}\mu \leq \varepsilon, \tag{F.34}$$

which concludes the proof. □

Lemma 126. *Assume* **A**9 *and* **A**10*. Then, $\{\exp[\bigoplus_{i\in\mathsf{S}} \psi^n_i]\}_{n\in\mathbb{N}}$ is uniformly integrable w.r.t. $\bar{\pi}$.*

Proof. It is enough to show that the sequence $\{f(\exp[\bigoplus_{i\in\mathsf{S}} \psi^n_i])\}_{n\in\mathbb{N}}$ is bounded in $\mathrm{L}^1(\bar{\pi})$, where $f : u \mapsto u\log(u)$ is continuous, convex and such that $\lim_{u\to\infty} f(u)/u = +\infty$, see Lemma 125. Let $n \in \mathbb{N}$. We have

$$\begin{aligned}
&\int_{(\mathbb{R}^d)^{\ell+1}} f(\exp[\textstyle\bigoplus_{i\in\mathsf{S}} \psi^n_i])\mathrm{d}\bar{\pi} = \mathrm{KL}(\pi^{nK} \mid \bar{\pi}) \\
&= \textstyle\int_{\mathbb{R}^d} \psi^n_{i_{K-1}} \mathrm{d}\mu_{i_{K-1}} + \sum_{k=0}^{K-2} \int_{\mathbb{R}^d} \psi^n_{i_k} \exp[\psi^n_{i_k} - \psi^{n+1}_{i_k}]\mathrm{d}\mu_{i_k} && \text{(Lemma 124)} \\
&= \textstyle\sum_{k=0}^{K-1} \int_{\mathbb{R}^d} \psi^n_{i_k} \mathrm{d}\mu_{i_k} + \sum_{k=0}^{K-2} \int_{\mathbb{R}^d} \psi^n_{i_k} \{\exp[\psi^n_{i_k} - \psi^{n+1}_{i_k}] - 1\}\mathrm{d}\mu_{i_k} \\
&\leq \mathrm{KL}(\pi^\star \mid \bar{\pi}) + (\bar{c}+1) \textstyle\sum_{k=0}^{K-2} \int_{\mathbb{R}^d} \psi^n_{i_k} \mathrm{d}\mu_{i_k} && \text{(Lemma 123-(a), \textbf{A}10)} \\
&\leq \mathrm{KL}(\pi^\star \mid \bar{\pi}) + (\bar{c}+1) \textstyle\sum_{k=0}^{K-2} \sup_{n\in\mathbb{N}} \int_{\mathbb{R}^d} \left|\psi^n_{i_k}\right| \mathrm{d}\mu_{i_k} < \infty\,. && \text{(Lemma 123-(c))}
\end{aligned}$$

□

With the preliminary results stated above, we are now ready to prove Proposition 16.

Proof of Proposition 16. Using **A**9 and **A**10, we have, by Lemma 126, uniform integrability of $\{\exp[\bigoplus_{i\in\mathsf{S}} \psi^n_i]\}_{n\in\mathbb{N}}$ in $\mathrm{L}^1(\bar{\pi})$. Therefore, the sequence $\{\pi^{nK}\}_{n\in\mathbb{N}}$ is relatively compact with respect to the weak topology of $\sigma(\mathrm{L}^1(\bar{\pi}), \mathrm{L}^\infty(\bar{\pi}))$, denoted as the τ-topology. We recall that $\lim_{n\to\infty} \mathrm{KL}(\pi^{nK+1} \mid \pi^{nK}) = 0$. This implies that $\{\pi^{nK+1}\}_{n\in\mathbb{N}}$ is also relatively τ-compact. By trivial recursion, we obtain that the sequences $\{\pi^{nK+k}\}_{n\in\mathbb{N}}$, where $k \in \{2, \ldots, K-1\}$ are also relatively τ-compact. Therefore, $\{\pi^n\}_{n\in\mathbb{N}}$ is relatively τ-compact and τ-sequentially compact.

We consider an increasing function $\Phi : \mathbb{N} \to \mathbb{N}$ such that $\{\pi^m\}_{m\in\Phi(\mathbb{N})}$ is a τ-convergent subsequence, and we denote by $\tilde{\pi}$ its limit for this topology. In particular, $\tilde{\pi} \in \mathscr{P}_\mathsf{S}$ by Proposition 15. We assume without loss of generality that $\Phi(\mathbb{N}) \subset K\mathbb{N}$.

Using the lower semi-continuity of the Kullback-Leibler divergence [Dupuis and Ellis, 2011, Lemma 1.4.3], we get

$$\mathrm{KL}(\tilde{\pi} \mid \bar{\pi}) \leq \liminf \mathrm{KL}(\pi^m \mid \bar{\pi}) \leq \limsup \mathrm{KL}(\pi^m \mid \bar{\pi}) \ .$$

Consider $k \in \{0, \dots, K-2\}$. By (F.27), we have

$$\frac{\mathrm{d}\mu_{i_k}}{\mathrm{d}\pi_{i_k}^{nK+k}} = \frac{\mathrm{d}\pi_{i_k}^{nK+k+1}}{\mathrm{d}\pi_{i_k}^{nK+k}} = \frac{\mathrm{d}\pi^{nK+k+1}}{\mathrm{d}\pi^{nK+k}} = \exp[\psi_{i_k}^{n+1} - \psi_{i_k}^n] \ ,$$

and thus,

$$\|\mu_{i_k} - \pi_{i_k}^{nK+k}\|_{\mathrm{TV}} = (1/2) \int_{\mathbb{R}^d} \left|\mathrm{d}\pi_{i_k}^{nK+k}/\mathrm{d}\mu_{i_k} - 1\right| \mathrm{d}\mu_{i_k} = (1/2) \int_{\mathbb{R}^d} \left|\exp[\psi_{i_k}^n - \psi_{i_k}^{n+1}] - 1\right| \mathrm{d}\mu_{i_k} \ .$$

With Proposition 15, we obtain that $\{\exp[\psi_{i_k}^n - \psi_{i_k}^{n+1}]\}_{n\in\mathbb{N}}$ converges to 1 in $\mathrm{L}^1(\mu_{i_k})$. In addition using the uniform integrability of $\{\psi_{i_k}^n\}_{n\in\mathbb{N}}$ and **A**10, we get

$$\limsup_{n\to+\infty} \int_{\mathbb{R}^d} \psi_{i_k}^n \exp[\psi_{i_k}^n - \psi_{i_k}^{n+1}] \mathrm{d}\mu_{i_k} = \limsup_{n\to+\infty} \int_{\mathbb{R}^d} \psi_{i_k}^n \mathrm{d}\mu_{i_k} \ .$$

We denote $m = K\ell$. Since $\mathrm{KL}(\pi^m \mid \bar{\pi}) = \int_{\mathbb{R}^d} \psi_{i_{K-1}}^\ell \mathrm{d}\mu_{i_{K-1}} + \sum_{k=0}^{K-2} \int_{\mathbb{R}^d} \psi_{i_k}^\ell \exp[\psi_{i_k}^\ell - \psi_{i_k}^{\ell+1}] \mathrm{d}\mu_{i_k}$ by Lemma 124, we finally have

$$\mathrm{KL}(\tilde{\pi} \mid \bar{\pi}) \leq \limsup\{\textstyle\sum_{k=0}^{K-1} \int_{\mathbb{R}^d} \psi_{i_k}^\ell \mathrm{d}\mu_{i_k}\} \leq \mathrm{KL}(\pi^\star \mid \bar{\pi})$$

where the last inequality comes from Lemma 123.

Since $\tilde{\pi}_i = \mu_i$ for any $i \in \mathsf{S}$, using Proposition 15, we have $\tilde{\pi} = \pi^\star$ by uniqueness of $\pi^\star$. Hence, $\pi^\star$ is the only limit point of $\{\pi^n\}_{n\in\mathbb{N}}$ in the τ-topology. In particular, $\mathrm{KL}(\pi^n \mid \bar{\pi}) \to \mathrm{KL}(\pi^\star \mid \bar{\pi})$. Since $\mathscr{P}_\mathsf{S}$ is convex, this last result implies $\|\pi^\star - \pi^n\|_{\mathrm{TV}} \to 0$, see the proof of Theorem 2.1 in Csiszár [1975]. □

We finish this section by highlighting that **A**10 is stronger than [Ruschendorf, 1995, B1]. A natural extension of the latter assumption would consist of having a guarantee on the $(K-1)$ first potentials given by (F.25), as presented below.

A 14. *There exist $0 < \underline{c} < \bar{c}$ such that for any $k \in \{0, \dots, K-2\}$, we have $\underline{c} \leq \exp(-\psi_{i_k}^1) \leq \bar{c}$.*

Under **A**14, [Ruschendorf, 1995, Lemma 4.3] can be adapted as written below.

Lemma 127. *Assume **A**14. Then, for any $n \in \mathbb{N}^*$*

(a) for any $k \in \{0, \dots, K-2\}$, there exists $\alpha_{n,k} \in \mathbb{N}$ such that

$$\underline{c} \cdot (\underline{c}/\bar{c})^{\alpha_{n,k}(K-2)} \leq \exp[\psi_{i_k}^{n-1} - \psi_{i_k}^n] \leq \bar{c} \cdot (\bar{c}/\underline{c})^{\alpha_{n,k}(K-2)}$$

(b) there exists $\alpha_{n,K-1} \in \mathbb{N}$ such that

$$1/\bar{c}^{K-1} \cdot (\underline{c}/\bar{c})^{\alpha_{n,K-1}(K-2)} \leq \exp[\psi_{i_{K-1}}^{n-1} - \psi_{i_{K-1}}^n] \leq 1/\underline{c}^{K-1} \cdot (\bar{c}/\underline{c})^{\alpha_{n,K-1}(K-2)}$$

where $\{\alpha_{n,k}\}_{n\in\mathbb{N}^*, k\in\{0,\dots,K-1\}}$ is a strictly increasing sequence that can be explicitly defined.

Proof. We prove the result by recursion on $n \in \mathbb{N}^*$.

Take $n = 1$. Let $k \in \{0, \dots, K-2\}$. We define $\alpha_{1,k} = 0$ and directly obtain (a) by **A**10 since $\psi_{i_k}^0 = 0$. Let us prove (b). We have by (F.25)

$$\begin{aligned}\exp[\psi_{i_{K-1}}^0 - \psi_{i_{K-1}}^1] &= \frac{\int_{(\mathbb{R}^d)^\ell} \exp[\bigoplus_{k=0}^{K-2} \psi_{i_k}^1] h \mathrm{d}\nu_{-i_{K-1}}}{\int_{(\mathbb{R}^d)^\ell} \exp[\bigoplus_{k=0}^{K-2} \psi_{i_k}^0] h \mathrm{d}\nu_{-i_{K-1}}} \\ &= \frac{\int_{(\mathbb{R}^d)^\ell} \exp[\bigoplus_{k=0}^{K-2} \{\psi_{i_k}^1 - \psi_{i_k}^0\} + \bigoplus_{k=0}^{K-2} \psi_{i_k}^0] h \mathrm{d}\nu_{-i_{K-1}}}{\int_{(\mathbb{R}^d)^\ell} \exp[\bigoplus_{k=0}^{K-2} \psi_{i_k}^0] h \mathrm{d}\nu_{-i_{K-1}}} \ .\end{aligned}$$

Using (a) at rank $n = 1$, we have

$$1/\bar{c}^{K-1} \leq \exp[\textstyle\bigoplus_{k=0}^{K-2} \{\psi_{i_k}^1 - \psi_{i_k}^0\}] \leq 1/\underline{c}^{K-1} \ ,$$

and therefore, we obtain (b) by taking $\alpha_{1,K-1} = 0$. Let us assume that the result is verified for some $n \in \mathbb{N}^*$. We have

$$\begin{aligned}\exp[\psi^n_{i_0} - \psi^{n+1}_{i_0}] &= \frac{\int \exp[\bigoplus_{k=1}^{K-1} \psi^n_{i_k}] h\mathrm{d}\nu_{-i_0}}{\int \exp[\bigoplus_{k=1}^{K-1} \psi^{n-1}_{i_k}] h\mathrm{d}\nu_{-i_0}} \\ &= \frac{\int \exp[\bigoplus_{k=1}^{K-2} \{\psi^n_{i_k} - \psi^{n-1}_{i_k}\} \oplus \{\psi^n_{i_{K-1}} - \psi^{n-1}_{i_{K-1}}\} + \bigoplus_{k=1}^{K-1} \psi^{n-1}_{i_k}] h\mathrm{d}\nu_{-i_0}}{\int \exp[\bigoplus_{k=1}^{K-1} \psi^{n-1}_{i_k}] h\mathrm{d}\nu_{-i_0}}\end{aligned}$$

Using (a) and (b) at rank n, we have

$$\begin{aligned}1/\bar{c}^{K-2} \cdot (\underline{c}/\bar{c})^{(K-2)\sum_{k=1}^{K-2}\alpha_{n,k}} &\leq \exp[\bigoplus_{k=1}^{K-2} \{\psi^n_{i_k} - \psi^{n-1}_{i_k}\}] \\ &\leq 1/\underline{c}^{K-2} \cdot (\bar{c}/\underline{c})^{(K-2)\sum_{k=1}^{K-2}\alpha_{n,k}} , \\ \underline{c}^{K-1} \cdot (\underline{c}/\bar{c})^{\alpha_{n,K-1}(K-2)} \leq \exp[\psi^n_{i_{K-1}} - \psi^{n-1}_{i_{K-1}}] &\leq \bar{c}^{K-1} \cdot (\bar{c}/\underline{c})^{\alpha_{n,K-1}(K-2)} .\end{aligned}$$

Therefore, we obtain

$$\begin{aligned}\underline{c} \cdot (\underline{c}/\bar{c})^{(K-2)\sum_{k=1}^{K-1}\alpha_{n,k}} &\leq \exp[\bigoplus_{k=1}^{K-2} \{\psi^n_{i_k} - \psi^{n-1}_{i_k}\} \oplus \{\psi^n_{i_{K-1}} - \psi^{n-1}_{i_{K-1}}\}] \\ &\leq \bar{c} \cdot (\bar{c}/\underline{c})^{(K-2)\sum_{k=1}^{K-1}\alpha_{n,k}} , \\ \underline{c} \cdot (\underline{c}/\bar{c})^{(K-2)\sum_{k=1}^{K-1}\alpha_{n,k}} \leq \exp[\psi^n_{i_0} - \psi^{n+1}_{i_0}] &\leq \bar{c} \cdot (\bar{c}/\underline{c})^{(K-2)\sum_{k=1}^{K-1}\alpha_{n,k}} .\end{aligned}$$

Now, we define $\alpha_{n+1,0} = \sum_{k=1}^{K-1} \alpha_{n,k}$ to obtain (a) for $k = 0$. Consider now $k \in \{1, \dots, K-2\}$. Following the same steps as above, we recursively define

$$\alpha_{n+1,k} = \textstyle\sum_{j=0}^{k-1} \alpha_{n+1,j} + \sum_{j'=k+1}^{K-1} \alpha_{n,j'} ,$$

which gives (a) at rank $n+1$. Let us now prove (b) at rank $n+1$. We have

$$\begin{aligned}\exp[\psi^n_{i_{K-1}} - \psi^{n+1}_{i_{K-1}}] &= \frac{\int \exp[\bigoplus_{k=0}^{K-2} \psi^{n+1}_{i_k}] h\mathrm{d}\nu_{-i_{K-1}}}{\int \exp[\bigoplus_{k=0}^{K-2} \psi^n_{i_k}] h\mathrm{d}\nu_{-i_{K-1}}} \\ &= \frac{\int \exp[\bigoplus_{k=0}^{K-2} \{\psi^{n+1}_{i_k} - \psi^n_{i_k}\} + \bigoplus_{k=0}^{K-2} \psi^n_{i_k}] h\mathrm{d}\nu_{-i_k}}{\int \exp[\bigoplus_{k=0}^{K-2} \psi^n_{i_k}] h\mathrm{d}\nu_{-i_{K-1}}} .\end{aligned}$$

Using (a) at rank $n+1$, we obtain

$$\begin{aligned}1/\bar{c}^{K-1} \cdot (\underline{c}/\bar{c})^{(K-2)\sum_{k=0}^{K-2}\alpha_{n+1,k}} &\leq \exp[\bigoplus_{k=0}^{K-2} \{\psi^{n+1}_{i_k} - \psi^n_{i_k}\}] \\ &\leq 1/\underline{c}^{K-1} \cdot (\bar{c}/\underline{c})^{(K-2)\sum_{k=0}^{K-2}\alpha_{n+1,k}} .\end{aligned}$$

Therefore, by taking $\alpha_{n+1,K-1} = \sum_{k=0}^{K-2} \alpha_{n+1,k}$, we obtain (b), which concludes the proof. □

Unfortunately, Lemma 127 only yields non-vacuous bounds in the case $K = 2$. Indeed, when $K > 2$, the sequence $\{\alpha_{n,k}\}_{n\in\mathbb{N}^*,k\in\{0,\dots,K-1\}}$ leads to increase the bounds on the quantities $\exp[\psi^{n-1}_{i_k} - \psi^n_{i_k}]$, which motivates the use of **A**10.

F.4.3 Proof of Section 7.5

For the rest of this section, we consider the multi-marginal Schrödinger bridge problem given by (TreeSB) and establish in Proposition 129 the correspondence with the regularized Wasserstein propagation problem presented in Solomon et al. [2014, 2015]. We first state a technical result.

Lemma 128. *Let $\varepsilon > 0$. Assume that π^0 is given by (7.4), where $r \in \mathsf{V}$ is chosen arbitrarily. Then, for any $\pi \in \mathscr{P}_{\mathsf{T}_r}$, we have*

$$\begin{aligned}\varepsilon \mathrm{KL}(\pi \mid \pi^0) = &\textstyle\sum_{(v,v')\in\mathsf{E}_r} \{w_{v,v'} \mathbb{E}_{\pi_{v,v'}}[\|X_v - X_{v'}\|^2] - \varepsilon \mathrm{H}(\pi_{v,v'})\} \\ &+ \varepsilon \textstyle\sum_{v\in\mathsf{V}} \mathrm{card}(\mathrm{C}_v) \mathrm{H}(\pi_v) + \varepsilon \mathrm{KL}(\pi_r \mid \pi^0_r) ,\end{aligned}$$

where we recall that $\mathrm{C}_v = \{v' \in \mathsf{V} : (v, v') \in \mathsf{E}_r\}$.

Proof. Since $\pi, \pi^0 \in \mathscr{P}_{\mathsf{T}_r}$, we obtain the following decomposition

$$\begin{aligned}
&\mathrm{KL}(\pi \mid \pi^0) \\
&= \mathrm{KL}(\pi_r \textstyle\prod_{(v,v')\in \mathsf{E}_r} \pi_{v'|v} \mid \pi^0_r \prod_{(v,v')\in \mathsf{E}_r} \pi^0_{v'|v}) \\
&= \mathrm{KL}(\pi_r \mid \pi^0_r) + \textstyle\sum_{(v,v')\in \mathsf{E}_r} \int_{\mathbb{R}^d} \mathrm{KL}(\pi_{v'|v}(\cdot|x_v) \mid \pi^0_{v'|v}(\cdot|x_v))\mathrm{d}\pi_v(x_v) \\
&= \mathrm{KL}(\pi_r \mid \pi^0_r) - \textstyle\sum_{(v,v')\in \mathsf{E}_r} \int_{\mathbb{R}^d\times\mathbb{R}^d} \log \pi^0_{v'|v} \mathrm{d}\pi_{v,v'} - \sum_{(v,v')\in \mathsf{E}_r} \int_{\mathbb{R}^d} \mathrm{H}(\pi_{v'|v}(\cdot|x_v))\mathrm{d}\pi_v(x_v) \, .
\end{aligned} \tag{F.35}$$

We finally obtain the result by using the definition of π^0 and noticing that $\int_{\mathbb{R}^d} \mathrm{H}(\pi_{v'|v}(\cdot|x_v))\mathrm{d}\pi_v(x_v) = \mathrm{H}(\pi_{v,v'}) - \mathrm{H}(\pi_v)$ for any $(v, v') \in \mathsf{E}_r$. □

Proposition 129. *Let $\varepsilon > 0$ and $\mu_0 \in \mathscr{P}$ such that $\mu_0 \ll \mathrm{Leb}$. Assume that π^0 is given by (7.4), where $r \in \mathsf{V}$ is chosen arbitrarily, and that $\varphi_r = \mathrm{d}\mu_0/\mathrm{dLeb}$. Also assume **A**7. Then, the set of marginals of the solution to (TreeSB) is exactly the solution to the entropic-regularized Wasserstein Propagation problem [Solomon et al., 2014, 2015] defined by*

$$\begin{aligned}
\arg\min\{\textstyle\sum_{(v,v')\in \mathsf{E}_r} w_{v,v'} W^2_{2,\varepsilon/w_{v,v'}}(\nu_v, \nu_{v'}) + \varepsilon \sum_{v\in\mathsf{V}} \mathrm{card}(\mathrm{C}_v)\mathrm{H}(\nu_v) + \varepsilon \mathrm{KL}(\nu_r \mid \mu_0) : \\
\{\nu_v\}_{v\in\mathsf{V}} \in \mathscr{P}^{\ell+1}, \ \nu_i = \mu_i, \forall i \in \mathsf{S}\} \, ,
\end{aligned} \tag{WP}$$

where we recall that $\mathrm{C}_v = \{v' \in \mathsf{V} : (v, v') \in \mathsf{E}_r\}$.

Proof. Assume that π^0 is given by (7.4), where $r \in \mathsf{V}$ is chosen arbitrarily, and that $\varphi_r = \mathrm{d}\mu_0/\mathrm{dLeb}$. In particular, we have $\pi^0_r = \mu_0$. Moreover, it is clear that π^0 verifies **A**6, and **A**8 by Proposition 121.

Let $\{\nu_v\}_{v\in\mathsf{V}} \in \mathscr{P}^{\ell+1}$ and $\{\nu^{(v,v')}\}_{(v,v')\in\mathsf{E}_r} \in (\mathscr{P}^{(2)})^{|\mathsf{E}_r|}$. We define

$$\begin{aligned}
F(\{\nu_v\}) &= \textstyle\sum_{(v,v')\in \mathsf{E}_r} w_{v,v'} W^2_{2,\varepsilon/w_{v,v'}}(\nu_v, \nu_{v'}) + \varepsilon \sum_{v\in\mathsf{V}} \mathrm{card}(\mathrm{C}_v)\mathrm{H}(\nu_v) + \varepsilon \mathrm{KL}(\nu_r \mid \mu_0) \, , \\
G(\nu_r, \{\nu^{(v,v')}\}) &= \textstyle\sum_{(v,v')\in \mathsf{E}_r} \{w_{v,v'} \mathbb{E}_{\nu^{(v,v')}}[\|X_v - X_{v'}\|^2] - \varepsilon \mathrm{H}(\nu^{(v,v')})\} \\
&\quad + \varepsilon \textstyle\sum_{(v,v')\in \mathsf{E}_r} \mathrm{H}(\nu^{(v,v')}_v) + \varepsilon \mathrm{KL}(\nu_r \mid \mu_0) \, .
\end{aligned} \tag{F.36}$$

By definition of the regularized Wasserstein distance given in (7.7), we have for any $\{\nu_v\}_{v\in\mathsf{V}} \in \mathscr{P}^{\ell+1}$

$$F(\{\nu_v\}) = \min\{G(\nu_r, \{\nu^{(v,v')}\}) : \nu^{(v,v')} \in \mathscr{P}^{(2)}, \nu^{(v,v')}_v = \nu_v, \nu^{(v,v')}_{v'} = \nu_{v'}, \forall (v, v') \in \mathsf{E}_r\} \, . \tag{F.37}$$

In particular, we have $F(\{\pi_v\}) \leq G(\pi_r, \{\pi_{v,v'}\})$ for any $\pi \in \mathscr{P}^{(\ell+1)}$. We now prove the result of Proposition 129 in two steps denoted by **Step 1** and **Step 2**.

Step 1. Let us not assume **A**7 for now. In this case, we prove in **Step 1.a** and **Step 1.b** that solving (WP) is equivalent to solving a modified version of (TreeSB) given by

$$\pi^\star = \arg\min\{\mathrm{KL}(\pi|\pi^0) \ : \ \pi \in \mathscr{P}_{\mathsf{T}_r}, \ \pi_i = \mu_i \, , \forall i \in \mathsf{S}\} \, . \tag{T_r-TreeSB}$$

Remark that any solution to (T_r-TreeSB) is a solution to (TreeSB), but the converse result may not be true.

Step 1.a: (WP) $\Longrightarrow$ (T_r-TreeSB). Consider a solution $\{\nu^\star_v\}_{v\in\mathsf{V}}$ to (WP). For any $(v, v') \in \mathsf{E}_r$, $W^2_{2,\varepsilon/w_{v,v'}}(\nu^\star_v, \nu^\star_{v'})$ is well defined and thus, there exists $\nu^{(v,v')} \in \Pi(\nu^\star_v, \nu^\star_{v'})$ such that

$$\nu^{(v,v')} \in \arg\min\{\mathbb{E}_\pi[\|X_v - X_{v'}\|^2] - (\varepsilon/w_{v,v'})\mathrm{H}(\pi) : \pi \in \Pi(\nu^\star_v, \nu^\star_{v'})\} \, . \tag{F.38}$$

Using the gluying lemma, we build the probability measure $\pi^\star = \nu^\star_r \prod_{(v,v')\in\mathsf{E}_r} \nu^{(v,v')}_{v'|v}$ such that (i) $\pi^\star \in \mathscr{P}_{\mathsf{T}_r}$, and (ii) $\pi^\star_{v,v'}$ and $\nu^{(v,v')}$ have the same distribution for any $(v, v') \in \mathsf{E}_r$. In particular, we have $\pi^\star_i = \mu_i$ for any $i \in \mathsf{S}$.

Let us show now that $\pi^\star$ is a solution to (T_r-TreeSB). Let $\pi \in \mathscr{P}_{\mathsf{T}_r}$ such that $\pi_i = \mu_i$ for any $i \in \mathsf{S}$. We have

$$\begin{aligned}
\epsilon\mathrm{KL}(\pi \mid \pi^0) &= G(\pi_r, \{\pi_{v,v'}\}) && \text{(Lemma 128)}\\
&\geq F(\{\pi_v\}) &&\\
&\geq F(\{\nu_v^\star\}) && \text{(definition of } \nu^\star\text{)}\\
&= G(\nu_r^\star, \{\nu^{(v,v')}\}) && \text{(see (F.38))}\\
&= G(\pi_r^\star, \{\pi_{(v,v')}^\star\}) && \text{(definition of } \pi^\star\text{)}\\
&= \epsilon\mathrm{KL}(\pi^\star \mid \pi^0)\,. && \text{(Lemma 128)}
\end{aligned}$$

Therefore, $\pi^\star$ is a solution to (T_r-TreeSB).

Step 1.b: (T_r-TreeSB) $\Longrightarrow$ (WP)**.** Consider now a solution $\pi^\star$ to (T_r-TreeSB). Since $\pi^\star \in \mathscr{P}_{\mathsf{T}_r}$, we have $\pi^\star = \pi_r^\star \prod_{(v,v')\in\mathsf{E}_r} \pi^\star_{v'|v}$ and $\pi_i^\star = \mu_i$ for any $i \in \mathsf{S}$.

Let us show that $\{\pi_v^\star\}_{v\in\mathsf{V}}$ is a solution to (WP). Let $\{\nu_v\}_{v\in\mathsf{V}} \in \mathscr{P}^{\ell+1}$ such that $\nu_i = \mu_i$ for any $i \in \mathsf{S}$.

Let $\{\nu^{(v,v')}\}_{(v,v')\in\mathsf{E}_r}$ be a family of probability measures such that $\nu^{(v,v')} \in \mathscr{P}^{(2)}, \nu_v^{(v,v')} = \nu_v, \nu_{v'}^{(v,v')} = \nu_{v'}$ for any $(v, v') \in \mathsf{E}_r$.

Using the gluying lemma, we build the probability measure $\pi = \nu_r \prod_{(v,v')\in\mathsf{E}_r} \nu^{(v,v')}_{v'|v}$, such that (i) $\pi \in \mathscr{P}_{\mathsf{T}_r}$ and (ii) $\pi_{v,v'}$ and $\nu^{(v,v')}$ have the same distribution for any $(v, v') \in \mathsf{E}_r$. We have

$$\begin{aligned}
\varepsilon\mathrm{KL}(\pi \mid \pi^0) &= G(\pi_r, \{\pi_{(v,v')}\}) && \text{(Lemma 128)}\\
&= G(\nu_r, \{\nu^{(v,v')}\}) && \text{(definition of } \pi\text{)}\\
&\geq \varepsilon\mathrm{KL}(\pi^\star \mid \pi^0) && \text{(definition of } \pi^\star\text{)}\\
&= G(\pi_r^\star, \{\pi_{(v,v')}^\star\})\,. && \text{(Lemma 128)}
\end{aligned}$$

By taking the infimum in the previous inequality over the families $\{\nu^{(v,v')}\}_{(v,v')\in\mathsf{E}_r}$, we obtain by (F.37) that

$$F(\{\nu_v\}) \geq G(\pi_r^\star, \{\pi_{(v,v')}^\star\}) \geq F(\{\pi_v^\star\}),$$

and therefore, $\{\pi_v^\star\}_{v\in\mathsf{V}}$ is a solution to (WP).

Step 2. We now assume **A**7. By Proposition 13, there exists a unique solution $\pi^\star \in \mathscr{P}^{(\ell+1)}$ to (TreeSB) such that we π^0-a.s. have $(\mathrm{d}\pi^\star/\mathrm{d}\pi^0) = \exp[\bigoplus_{i\in\mathsf{S}} \psi_i^\star]$, where $\{\psi_i^\star\}_{i\in\mathsf{S}}$ are measurable potentials with $\psi^\star : \mathbb{R}^d \to \mathbb{R}$. Since $\pi^0 \in \mathscr{P}_{\mathsf{T}_r}$, we also have $\pi^\star \in \mathscr{P}_{\mathsf{T}_r}$, *i.e.*, the potentials $\{\psi_i^\star\}_{i\in\mathsf{S}}$ do not modify the Markovian nature of π^0. Therefore, $\pi^\star$ is also the unique solution to (T_r-TreeSB). Using the equivalence between (T_r-TreeSB) and (WP) established in **Step 1**, we finally obtain the result of Proposition 129. □

In particular, Proposition 17 directly derives from Proposition 129 by taking $r = i_{K-1}$ and $\mu_0 = \mu_{i_{K-1}}$.

F.4.4 Comparison with Haasler et al. [2021]

In their work, Haasler et al. [2021] study the *static* and *discrete-state* counterpart of our approach. Given a state space X such that $|\mathsf{X}| = n + 1$ with $n \in \mathbb{N}$, they establish a correspondence between multi-marginal EOT with a general tree-based cost and discrete-time multi-marginal static Schrödinger bridge, and provide an efficient method to solve these problems. In this section, we provide details on their framework and give a precise comparison between our theory and their results.

To be coherent with the setting of Haasler et al. [2021], we adapt here some of our notation. Let us define $\mathsf{Z}^{(q)} = \mathbb{R}_+^{(n+1)^q}$. For any $q \in \mathbb{N}^*$, the set of probability measures on X^q is defined as $\mathscr{P}^{(q)} = \{M \in \mathsf{Z}^{(q)} : \langle M, \mathbf{1}\rangle = 1\}$. We denote $\mathscr{P} = \mathscr{P}^{(1)}$. For any tensors $M, P \in \mathsf{Z}^{(q)}$, the Kullback-Leibler divergence between M and P is defined as $\mathrm{KL}(M \mid P) = \langle M \log(M/P) - M + P, \mathbf{1}\rangle$ and the entropy of M is defined as $\mathrm{H}(M) = -\mathrm{KL}(M \mid \mathbf{1})$, where the operations are meant componentwise. In the rest of the section, we consider an undirected tree $\mathsf{T} = (\mathsf{V}, \mathsf{E})$ with $|\mathsf{V}| = \ell + 1$ such that V may be identified with $\{0, \dots, \ell\}$.

Details on the results of Haasler et al. [2021]. In their paper, the authors consider a cost tensor $C \in \mathsf{Z}^{(\ell+1)}$ that factorizes along T, *i.e.*, for any $\{j_0, \dots, j_\ell\}$ with for any $i \in \{0, \dots, \ell\}$, $j_i \in \{0, \dots, n\}$, we have

$$C_{j_0,\dots,j_\ell} = \sum_{(v,v')\in \mathsf{E}} C^{\{v,v'\}}_{j_v, j_{v'}}, \tag{F.39}$$

where $C^{\{v,v'\}} \in \mathsf{Z}^{(2)}$ is a cost matrix for transportation between the marginals at vertices v and v', see [Haasler et al., 2021, Eq. (3.1)]. In particular, this cost can be seen as the discrete counterpart of the tree-based cost introduced in (7.3) in the quadratic setting.

Given a subset $\mathsf{S} \subset \mathsf{V}$ with $|\mathsf{S}| = K$ and a set of marginals $\{\mu_i\}_{i\in\mathsf{S}} \in \mathscr{P}^K$, Haasler et al. [2021] study the EOT problem associated to T, see [Haasler et al., 2021, Eq. (2.4)], which is given by

$$\arg\min\{\langle C, M\rangle - \varepsilon \mathrm{H}(M) : M \in \mathscr{P}^{(\ell+1)}, \mathrm{proj}_i(M) = \mu_i, \forall i \in \mathsf{S}\} \,. \tag{discrete-EmOT}$$

This problem may be solved with Sinkhorn algorithm [Cuturi, 2013, Knight, 2008, Sinkhorn and Knopp, 1967], for which the authors provide an efficient implementation adapted to the tree-based setting, see [Haasler et al., 2021, Algorithm 3.1]. Moreover, they state the convergence of their method in [Haasler et al., 2021, Theorem 3.5], as a direct consequence of the results presented in Luo and Tseng [1992].

In [Haasler et al., 2021, Section 4.2], it is assumed that S corresponds to the set of the leaves of T, as we do, and it is shown an equivalence between (discrete-EmOT) and the discrete-state static SB problem stated in [Haasler et al., 2021, Eq 4.2], which is given by

$$\begin{aligned}\arg\min\{&\textstyle\sum_{(v,v')\in\mathsf{E}_r} \mathrm{KL}(M^{(v,v')} \mid \mathrm{diag}(\nu_v) A^{(v,v')}) : \\ &M^{(v,v')} \in \mathscr{P}^{(2)}, \{\nu_v\}_{v\in\mathsf{V}} \in \mathscr{P}^{\ell+1}, M^{(v,v')}\mathbf{1} = \nu_v, {M^{(v,v')}}^\top \mathbf{1} = \nu_{v'},\ \nu_i = \mu_i, \forall i \in \mathsf{S}\} \,,\end{aligned} \tag{discrete-TreeSB}$$

where $\mathsf{T}_r = (\mathsf{V}, \mathsf{E}_r)$ is the directed version of T rooted in an arbitrary vertex $r \in \mathsf{S}$, and $A^{(v,v')} = \exp(-C^{(v,v')}/\varepsilon) \in \mathsf{Z}^{(2)}$ for any $(v, v') \in \mathsf{E}_r$. Remark that $A^{(v,v')}$ may not necessarily be a transition probability matrix.

Finally, Haasler et al. [2021] provide two main numerical experiments. In [Haasler et al., 2021, Section 5.2], they consider a tree with 15 vertices, 14 edges and 8 leaves, combined to the state-space $\mathsf{X} = \{0, 1\}^{50\times 50}$, and solve the corresponding (discrete-EmOT) problem for the quadratic cost. In [Haasler et al., 2021, Section 6], they apply their methodology to estimate ensemble flows on a hidden Markov chain. Given $\tau \in \mathbb{N}^*$, they consider a tree T with τ internal vertices (modeling the distribution of N agents at time $t \in \{1, \dots, \tau\}$), that are linearly linked, and such that each of these vertices is independently linked to S leaves of T (modeling observations at time $t \in \{1, \dots, \tau\}$). In this setting, the state space is given by $\mathsf{X} = \{1, \dots, 100\}^N$. They solve the formulation (discrete-TreeSB) where the reference measure is chosen as a random walk.

Comparison with our results. We now establish remarks on the main differences between our methodology and the work of Haasler et al. [2021].

First of all, the continuous state-space counterpart of (discrete-TreeSB) is given by

$$\arg\min\{\mathrm{KL}(\pi \mid \pi^0) : \pi \in \mathscr{P}_{\mathsf{T}_r}, \pi_i = \mu_i, \forall i \in \mathsf{S}\} \,, \tag{F.40}$$

where π^0 is a reference measure which factorizes along T_r. In this case, $\pi_{v,v'}$, π_v and $\pi^0_{v'|v}$ in (F.40) respectively correspond to the continuous version of $M^{(v,v')}$, ν_v and $A^{(v,v')}$ in (discrete-TreeSB). In contrast, our formulation of the multi-marginal Tree Schrödinger Bridge problem given in (TreeSB) is a minimization problem over all probability measures $\pi \in \mathscr{P}^{(\ell+1)}$, and is not restricted to the distributions that admit a Markovian factorization along T as in (F.40). Hence, our framework may be considered more general. Remark that under **A**6, **A**7 and **A**8, Proposition 13 states that (TreeSB) admits a unique solution $\pi^\star \ll \pi^0$ such that $(\mathrm{d}\pi^\star/\mathrm{d}\pi^0)$ can be written with potentials. Then, $\pi^\star \in \mathscr{P}_{\mathsf{T}_r}$ since $\pi^0 \in \mathscr{P}_{\mathsf{T}_r}$, and (TreeSB) is then equivalent to (F.40).

Furthermore, (EmOT) is more general than the continuous version of (discrete-EmOT), which we can recover by taking any measure ν of the form $(\mathrm{d}\nu/\mathrm{dLeb}) = \exp[\bigoplus_{i\in\mathsf{S}} \varphi_i]$ in (EmOT), where $\{\varphi_i\}_{i\in\mathsf{S}}$ is a family of potentials such that $\left|\int_{\mathbb{R}^d} \varphi_i \mathrm{d}\mu_i\right| < \infty$ for any $i \in \mathsf{S}$. As a consequence, our setting allows us to choose the root $r \in \mathsf{V}\backslash\mathsf{S}$ for the SB problem, whereas Haasler et al. [2021] only consider the case where $r \in \mathsf{S}$. In the latter case, we establish in Appendix F.5 that r can be chosen arbitrarily, as stated by [Haasler et al., 2021, Corollary 4.3].

Finally, TreeDSB deeply differs from the framework of Haasler et al. [2021] due its *dynamic* nature. Although we solve the same tree-based static SB problem (up to continuous/discrete state-space consideration), our

approach consists in computing dynamic iterates (*i.e.*, path measures) using diffusion-based methods instead of static iterates (*i.e.*, distributions) using Sinkhorn algorithm. This paradigm is at the core of the DSB [Bortoli et al., 2021] methodology, and offers an efficient approach to tackle high-dimensional settings, where Sinkhorn algorithm would fail.

Here, we present some advantages of the method proposed by Haasler et al. [2021] compared to ours. First, Haasler et al. [2021] may choose any kind of tree-based cost in practice, while our methodology only holds for the quadratic cost. This limitation is shared with all approaches based on the DSB [Bortoli et al., 2021] methodology. Indeed, since the cost is determined by the reference path measure, we often choose quadratic costs associated with Brownian motions or Ornstein-Uhlenbeck processes. Moreover, Haasler et al. [2021] may consider various inhomogeneous (discrete) state spaces for the vertices of T, as presented in their numerical experiments. In our case, this approach is not compatible with our diffusion-based method. Finally, unlike Haasler et al. [2021], our method is not scalable with the number of vertices or edges in T due to computational limits. This limitation is common to all multi-marginal approaches which rely on neural networks to parameterize the potential and/or the distributions of the multi-marginal OT method, see Li et al. [2020], Fan et al. [2021], Korotin et al. [2022, 2021] for instance.

F.5 Further results on TreeSB

Choice of the root r in (TreeSB). We recall that the reference measure π^0 considered in (TreeSB), which is defined in (7.4), verifies $\pi^0 \in \mathscr{P}_{\mathsf{T}_r}$ for some fixed root $r \in \mathsf{V}$ and $\pi^0_r \ll \mathrm{Leb}$ with density φ_r. Moreover, we have $\pi^0_{v'|v}(\cdot \mid x_v) = \mathrm{N}(x_v, \varepsilon/(2w_{v,v'})\mathrm{I}_d)$ for any $(v, v') \in \mathsf{E}_r$, and thus, π^0 is entirely determined by the choice of the root r and the density on the corresponding vertex φ_r.

As presented in Appendix F.4.1, we recall that (TreeSB) is equivalent to any multi-marginal Tree-SB problem with a reference measure $\bar{\pi}^0$ given by (F.21), *i.e.*, $\bar{\pi}^0$ writes as $(\mathrm{d}\bar{\pi}^0/\mathrm{d}\pi^0) = \exp[\bigoplus_{i\in\mathsf{S}} \psi^0_i]$, where $\{\psi^0_i\}_{i\in\mathsf{S}}$ is a family of measurable potentials with $\psi^0_i : \mathbb{R}^d \to \mathbb{R}$ such that $\left|\int_{\mathbb{R}^d} \psi^0_i \mathrm{d}\mu_i\right| < \infty$ for any $i \in \mathsf{S}$. In the case where r is chosen as a leaf of T, this result implies that (TreeSB) is unchanged if

(a) $\varphi_r = \mathrm{d}\nu/\mathrm{dLeb}$ where $\nu \in \mathscr{P}$ is such that $\mathrm{KL}(\mu_r|\nu) < \infty$,

(b) r is replaced by $r' \in \mathsf{S}$, as long as $\mathrm{H}(\mu_r) < \infty$ and $\mathrm{H}(\mu_{r'}) < \infty$.

Therefore, under **A**5, the setting chosen in Section 7.3 is equivalent to any other setting where r is arbitrarily chosen in S and $\varphi_r = \mathrm{d}\nu/\mathrm{dLeb}$ where $\mathrm{KL}(\mu_r|\nu) < \infty$.

Consider now the case where $r \in \mathsf{S}^c$, *i.e.*, r is not a leaf of T. Then, the choice of φ_r can not be made arbitrarily anymore, since it determines a further regularization on the r-th marginal of the solution to (TreeSB). In this setting, the sequence defined by (mIPF) is unchanged. Hence, TreeDSB proceeds in the same manner as presented in Section 7.3, except for the first iteration, which we detail now.

Let us define $\mathsf{P} = \mathrm{path}_{\mathsf{T}_{i_0}}(i_0, r)$, where $\mathsf{T}_{i_0} = (\mathsf{V}, \mathsf{E}_{i_0})$ is the directed version of T rooted in i_0. We recall that first iterate of (mIPF) is defined by

$$\pi^1 = \arg\min\{\mathrm{KL}(\pi \mid \pi^0) : \pi \in \mathscr{P}^{(\ell+1)}, \pi_{i_0} = \mu_{i_0}\} .$$

Following the proof of Lemma 117, it is clear that

$$\pi^1 = \mu_{i_0} \bigotimes_{(v,v')\in\mathsf{P}} \pi^0_{v'|v} \bigotimes_{(v,v')\in\mathsf{E}_{i_0}\setminus\mathsf{P}} \pi^0_{v'|v} = \mu_{i_0} \bigotimes_{(v,v')\in\mathsf{E}_{i_0}} \pi^0_{v'|v} ,$$

where we emphasize that $\mathsf{P} = \{(v, v') \in \mathsf{E}_{i_0} : (v', v) \in \mathsf{E}_r\}$. Therefore, Proposition 12 still applies between r and i_0, by considering r instead of i_{K-1}. In practice, this means that the first iteration of TreeDSB consists in computing the time reversal of the path measures $\mathbb{P}^0_{(v',v)}$ for any $(v, v') \in \mathsf{P}$.

Extension of the regularized Wasserstein barycenter problem (regWB). Consider the regularized Wasserstein-2 barycenter problem defined as follows

$$\mu^\star_\varepsilon = \arg\min\{\textstyle\sum_{i=1}^\ell w_i W^2_{2,\varepsilon/w_i}(\mu, \mu_i) + \ell\varepsilon\mathrm{H}(\mu) + \varepsilon\mathrm{KL}(\mu \mid \mu_0) : \mu \in \mathscr{P}\} , \qquad (\mu_0\text{-regWB})$$

where $(w_i)_{i\in\{1,\dots,\ell\}} \in (0, +\infty)^\ell$ and $\mu_0 \in \mathscr{P}$ is a reference measure. This formulation admits a further regularization compared to (regWB), which tends to make $\mu^\star_\varepsilon$ closer to μ_0. In particular, given a Wasserstein

barycenter problem onto a star-shaped tree, the formulation (μ_0-regWB) may be more adapted than (regWB) if we have an *a priori* on the form of the regularized barycenter. In the case where $\mu_0 = \mathrm{N}(0, \sigma_0^2 \mathrm{I}_d)$, letting $\sigma_0 \to \infty$, we recover the $(\ell\varepsilon, (\ell-1)\varepsilon)$ doubly-regularized Wasserstein barycenter problem (regWB). In the same spirit as Proposition 17, we can derive the following result from Proposition 129, which proves that (μ_0-regWB) can be solved with TreeDSB.

Proposition 130. *Let $\varepsilon > 0$ and $\mu_0 \in \mathscr{P}$ such that $\mu_0 \ll \mathrm{Leb}$. Assume **A**5. Also assume that T is a star-shaped tree with central node indexed by 0, and that the reference measure of* (TreeSB) *defined in* (7.4) *verifies $r = 0$ and $\varphi_r = \mathrm{d}\mu_0/\mathrm{dLeb} > 0$. Under* **A**7, ($\mu_0$-regWB) *has a unique solution $\pi_0^\star$, where $\pi^\star$ is the solution to* (TreeSB).

Below, we provide practical guidelines to parameterize μ_0 when it is chosen as a Gaussian distribution.

Gaussian design of μ_0 in (μ_0-regWB). Consider an undirected star-shaped tree T with $K+1$ vertices and leaves $\{1, \dots, K\}$. In order to incorporate the marginal constraints in the penalization brought by μ_0 when it is a Gaussian distribution, we set its mean to $\sum_{i=1}^K \mathbb{E}[\mu_i]/K$ and its diagonal covariance matrix as $\alpha \times (\sum_{i=1}^K \mathrm{diag}(\mathrm{Cov}[\mu_i])^{-1}/K)^{-1}$, where the inverse operation is component-wise and α is a positive hyperparameter. This choice of variance helps to correctly explore the state-space at the very first iteration of TreeDSB, which is key to ensure numerical stability. In this setting, (TreeSB) verifies **A**7 and **A**8, by Proposition 120 and Proposition 121. In particular, we use this approach for two of our experiments: synthetic Gaussian datasets and Bayesian fusion, see Appendix F.7.

F.6 Algorithmic techniques

Time discretization in TreeDSB. Denote $k_n = (n-1) \bmod (K)$ for any $n \in \mathbb{N}$. Let $\mathsf{T} = (\mathsf{V}, \mathsf{E})$ be a weighted undirected tree and consider the multi-marginal Schrödinger bridge problem (TreeSB) associated to this tree. We recall that for any $\{v, v'\} \in \mathsf{E}$, we define $T_{v,v'} = \varepsilon/(2w_{v,v'})$.

Consider the path measures $\{\mathbb{P}^n_{(v,v')}\}_{n\in\mathbb{N},(v,v')\in\mathsf{E}_{k_n}}$ recursively defined by (a) and (b). By combining Proposition 11, Proposition 12 and results on time reversal theory [Haussmann and Pardoux, 1986], we obtain by recursion that for any $n \in \mathbb{N}$, any $(v, v') \in \mathsf{E}_{k_n}$, $\mathbb{P}^n_{(v,v')}$ is associated with a Stochastic Differential Equation on $[0, T_{v,v'}]$ given by

$$\mathrm{d}\mathbf{X}_t = f^n_{t,v,v'}(\mathbf{X}_t)\mathrm{d}t + \mathrm{d}\mathbf{B}_t, \quad \mathbf{X}_0 \sim \pi^n_v \,. \tag{F.41}$$

Let $N \in \mathbb{N}^*$. In order to sample from the dynamics (F.41) at iteration $n \in \mathbb{N}$, we consider its Euler-Maruyama discretization on $(N+1)$ time steps,

$$X_{m+1} = X_m + \gamma_{m+1} f^n_{t_m,v,v'}(X_m) + \sqrt{\gamma_{m+1}} Z_{m+1}, \quad X_0 \sim \pi^n_v \,, \tag{F.42}$$

where $Z_m \sim \mathrm{N}(0, \mathrm{I}_d)$ for any $m \in \{1, \dots, N\}$, $t_m = \sum_{i=1}^m \gamma_i$, and $\{\gamma_m\}_{m=1}^N \in (0, \infty)^N$ is a time schedule such that $\sum_{m=1}^N \gamma_m = T_{v,v'}$. This results in approximating the path measure $\mathbb{P}^n_{(v,v')}$ by the joint distribution $\pi^{n,N}_{(v,v')} \in \mathscr{P}^{(N+1)}$ defined by

$$\pi^{n,N}_{(v,v')} = \pi^n_v \bigotimes_{m=0}^{N-1} \pi^{n,N}_{(v,v'),m+1|m} \,,$$

where $\pi^{n,N}_{(v,v'),m+1|m}(\cdot|x_m) = \mathrm{N}(x_m + \gamma_{m+1} f^n_{t_m,v,v'}(x_m), \gamma_{m+1}\mathrm{I}_d)$ for any $m \in \{0, \dots, N-1\}$. If N is chosen large enough, then $\pi^{n,N}_{(v,v'),m}$ and $\mathbb{P}^n_{(v,v'),t_m}$ have approximately the same distribution for any $m \in \{0, \dots, N\}$. Consequently, $(\mathbb{P}^n_{(v,v')})^R$ is naturally approximated by the joint distribution $\tilde{\pi}^{n,N}_{(v,v')} \in \mathscr{P}^{(N+1)}$ defined by

$$\tilde{\pi}^{n,N}_{(v,v')} = \pi^n_{v'} \bigotimes_{m=0}^{N-1} \pi^{n,N}_{(v,v'),N-m-1|N-m} \,.$$

If N is chosen large enough, we obtain that

$$\begin{aligned}&\pi^{n,N}_{(v,v'),N-m-1|N-m}(\cdot|x_{N-m}) \\ &\qquad = \mathrm{N}(x_{N-m} - \gamma_{N-m} f^n_{t_{N-m},v,v'}(x_{N-m}) + \gamma_{N-m} \nabla \log p_{v,v',t_{N-m}}(x_{N-m}), \gamma_{N-m}\mathrm{I}_d) \,,\end{aligned}$$

where $p_{v,v',t}$ is the density of $\mathbb{P}^n_{(v,v'),t}$ w.r.t. the Lebesgue measure.

Following the construction of our dynamic iterates, we now explain how the sequence $\{\pi^n_{(v,v')}\}_{n\in\mathbb{N}^*,(v,v')\in\mathsf{E}_{k_n}}$ is recursively defined. Let $n\in\mathbb{N}$, $k_n=(n-1) \bmod(K)$. Define the path $\mathsf{P}_n=\mathrm{path}_{\mathsf{T}_{i_{k_n}}}(i_{k_n},i_{k_n+1})$. Then, for any $(v,v')\in\mathsf{E}_{k_n+1}$,

(a) if $(v,v')\in\mathsf{E}_{k_n}\backslash\mathsf{P}_n$, then $\pi^{n+1,N}_{(v,v')}=\pi^{n+1}_v\bigotimes_{m=0}^{N-1}\pi^{n,N}_{(v,v'),m+1|m}$,

(b) if $(v',v)\in\mathsf{P}_n$, then $\pi^{n+1,N}_{(v,v')}=\pi^{n+1}_v\bigotimes_{m=0}^{N-1}\pi^{n,N}_{(v',v),N-m-1|N-m}$.

These computations may be obtained by considering the sequence given by (mIPF) to solve the multi-marginal Tree-SB problem associated to $\mathsf{T}^{(N)}=(\mathsf{V}^{(N)},\mathsf{E}^{(N)})$, the N-discretized version of T (see Appendix F.2) with weights $w^{(N)}_{e_m}=2\gamma_m/\varepsilon$, which is given by

$$\pi^\star=\arg\min\{\mathrm{KL}(\pi|\pi^{0,N})\ :\ \pi\in\mathscr{P}(\mathsf{V}^{(N)}),\ \pi_i=\mu_i\ ,\forall i\in\mathsf{S}\}\ , \tag{F.43}$$

with $\pi^{0,N}=\pi^0_r\bigotimes_{(v,v')\in\mathsf{E}_r}\pi^{0,N}_{(v,v'),1:N|0}$.

To approximate the IPF recursion given by (a) and (b), we use **on each edge** of T the score-matching approach of Bortoli et al. [2021], which avoids heavy computations of score approximations. The next proposition is direct adaptation of [Bortoli et al., 2021, Proposition 3].

Proposition 131. *Assume that for any $n\in\mathbb{N}$, any $(v,v')\in\mathsf{E}_{k_n}$ with $k_n=(n-1) \bmod(K)$, we have*

$$\pi^{n,N}_{(v,v'),m+1|m}(\cdot|x_m)=\mathrm{N}(F^n_{m,v,v'}(x_m),\gamma_m\mathrm{I}_d)\ .$$

Let $n\in\mathbb{N}$. Consider the path $\mathsf{P}_n=\mathrm{path}_{\mathsf{T}_{i_{k_n}}}(i_{k_n},i_{k_n+1})$. Let $(v,v')\in\mathsf{E}_{k_n+1}$. Define $p^n=\pi^{n,N}_{(v,v')}$ and $m_N=N-m-1$. Then, if $(v',v)\in\mathsf{P}_n$, we have

$$F^{n+1}_{m,v,v'}=\arg\min_{\mathrm{F}\in\mathrm{L}^2(\mathbb{R}^d,\mathbb{R}^d)} \tag{F.44}$$

$$\mathbb{E}_{p^n_{m_N,m_N+1}}[\|\mathrm{F}(X_{m_N+1})-(X_{m_N+1}+F^n_{m_N,v',v}(X_{m_N})-F^n_{m_N,v',v}(X_{m_N+1}))\|^2], \tag{F.45}$$

otherwise, we have $F^{n+1}_{m,v,v'}=F^n_{m,v,v'}$.

In practice, we use two neural networks per edge $\{v,v'\}\in\mathsf{E}$, one for each possible direction of the edge, such that $F_{v,v'}(\theta^n_{v,v'},m,x)\approx F^n_{m,v,v'}(x)$ and $F_{v',v}(\theta^n_{v',v},m,x)\approx F^n_{m,v',v}(x)$. For any $\{v,v'\}\in\mathsf{E}$, the parameter $\theta^n_{v,v'}$ is updated at iteration n via the score matching loss defined by (F.44) in Proposition 131 if $(v,v')\in\mathrm{path}_{T_{i_{k_n}}}(i_{k_n},i_{k_n+1})$, see Algorithm 3.

F.7 Additional experimental results and details

The numerical experiments presented in Section 7.7 are obtained by our own Pytorch implementation, which is inspired from the code[1] provided by Bortoli et al. [2021]. We first provide information on the general setting of our experiments in Appendix F.7.1, and then give details on each of them in Appendix F.7.2 along with additional results. We recall that a mIPF cycle is defined as a subset of K consecutive iterations of (mIPF) and that the order of the leaves given by $\{i_0,\dots,i_{K-1}\}$ is randomly shuffled at each new mIPF cycle.

F.7.1 General experimental setup

Implementation of Algorithm 3 in practice. Let $n\in\mathbb{N}$, with $k_n=(n-1) \bmod(K)$, $k_n+1=n \bmod(K)$. Consider the path $\mathsf{P}_n=\mathrm{path}_{\mathsf{T}_{i_{k_n}}}(i_{k_n},i_{k_n+1})$. Assume that we are provided with a dataset $\mathsf{D}_{i_{k_n}}$, which contains M samples from $\pi^n_{i_{k_n}}$. Following Lines 7-9 in Algorithm 3, we apply processes (a) and (b) recursively on the edges $(v,v')\in\mathsf{P}_n$.

(a) Sampling step (Line 7). For any $x_0\in\mathsf{D}_v$, we sample from the diffusion trajectory (F.42) given by the Euler Maruyama discretization of $\mathbb{P}^n_{v,v'}$ starting from x_0. This gives us $M\times N$ trajectory samples. We then store the last iterate of each trajectory in a new dataset $\mathsf{D}_{v'}$, which thus approximates $\pi^n_{v'}$.

[1] https://github.com/JTT94/diffusion_schrodinger_bridge

(b) Training step (Lines 8-9). In order to avoid heavy computation, we approximate the *mean-matching* loss (F.44) by an unbiased estimator obtained by subsampling b elements from the *full* trajectories computed in the sampling process, see [Bortoli et al., 2021, Eq. (97)-(98)]. Here, b refers to the *batch-size* parameter of the neural networks. Then, we perform gradient descent to optimize the parameter $\theta_{v',v}$, which parameterizes the *backward* drift on the edge (v, v').

To avoid any bias issue, the whole trajectories obtained at process (a) are refreshed at a certain frequency over the training iterations of the neural networks by once again simulating the diffusion (F.42). In our experiments, this refresh occurs each 500 iterations.

Setting of the time discretization. The number of time-steps N in the time discretization of the diffusions is chosen to be even and identical for each of the edges of the tree. Let $\{v, v'\} \in \mathsf{E}$. We now give details on the design of the time schedule $\{\gamma_k\}_{k=1}^N$ related to the edge $\{v, v'\}$, see Appendix F.6. Following Bortoli et al. [2021], we choose this sequence to be invariant by time reversal and consider $\gamma_k = \gamma_0 + (2k/N)(\bar{\gamma} - \gamma_0)$ for any $k \in \{0, \dots, N/2\}$ (the rest of the sequence being obtained by symmetry) where γ_0 is a free parameter and $\bar{\gamma}$ is determined by $\sum_{k=1}^N \gamma_k = T_{v,v'}$. In our experiments, we set $N = 50$ and $\gamma_0 = 10^{|5}$.

Sampling improvement. In our code, we implemented the corrector scheme of Song et al. [2021b] and the *probability flow*-based sampling approach detailed in [Bortoli et al., 2021, Section H.3], but did not observe any significant improvement in our experiments using one of these techniques.

Choice of the architectures of the neural networks. In the case of the experiments related to synthetic datasets (two-dimensional toy datasets, Gaussian distributions) and to the subset posterior aggregation task, we implement the same architecture as presented in [Bortoli et al., 2021, Figure 3]. We refer to this model as "Basic Model" and detail it in Figure F.1. In the "Basic Model", the PositionalEncoding block applies the sine transform described in Vaswani et al. [2017], with output dimension equal to 32, and each MLP Block represents a Multilayer Perceptron Network. In particular, MLPBlock (1a) has shape $(d, 128, \max(256, 2d))$, MLPBlock (1b) has shape $(32, 128, \max(256, 2d))$, and MLPBlock (2) has shape $(2 \times \max(256, 2d), \max(256, 2d), \max(128, d), d)$, where d denotes the dimension of input data. We optimize the networks with ADAM [Kingma and Ba, 2014] with learning rate 10^{-4} and momentum 0.9. For each of the networks, we set the batch size to 4,096 and the number of iterations to 10,000 for the synthetic datasets and 15,000 for the subset posterior aggregation task. Our experiments ran on 1 Intel Xeon CPU Gold 6230 20 cores @ 2.1 Ghz CPU.

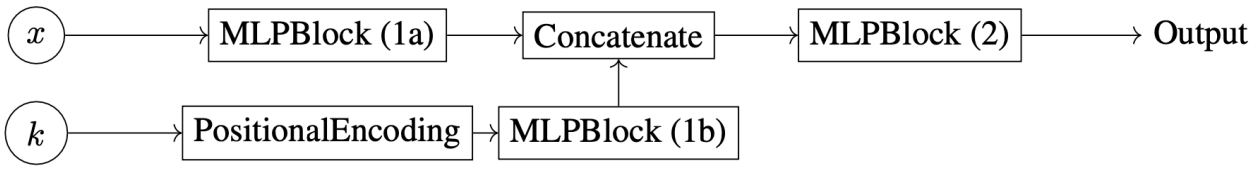


Figure F.1: Architecture of the "Basic Model".

In the case of the experiments related to MNIST dataset, we use a reduced UNET architecture based on Nichol and Dhariwal [2021], where we set the number of channels to 64 rather than 128. We implement an exponential moving average of network parameters across training iterations, with rate 0.999. We optimize the networks with ADAM [Kingma and Ba, 2014] with learning rate 10^{-4} and momentum 0.9. Finally, we set the batch size to 256 and the number of training iterations to 30,000. Our experiments ran using 1 Nvidia A100.

Details on regularized state-of the art methods. We run the fsWB algorithm [Cuturi and Doucet, 2014] with the implementation provided by Flamary et al. [2021]. For each experiment, we run 100 Sinkhorn iterations with 1500 samples for each dataset (*i.e.*, the maximum number of samples that it can generate) and set the regularization parameter ε to its lowest value such that the algorithm is stable. Finally, for sake of fairness with our method, we initialise the barycenter measure with π_r^0 when solving the problem (μ_0-regWB) for synthetic Gaussian datasets and Bayesian fusion. To run the crWB algorithm [Li et al., 2020], we use the code provided by the authors. We consider the quadratic regularization, which is shown to be empirically more stable than entropic regularization. Following Fan et al. [2021], we choose the potential networks to be fully connected neural networks with 3 hidden layers of shape $(\max(128, 2d), \max(128, 2d), \max(128, 2d))$. The activation functions are ReLu. We optimize the networks with ADAM [Kingma and Ba, 2014] with learning

rate 10^{-4} for the subset posterior aggregation task and 10^{-3} for the Gaussian experiment. Finally, we set the batch size to 4,096 and the number of training iterations to 50,000. We highlight that fsWB and crWB solve a regularized Wasserstein barycenter problem, which does not contain an additional *penalization* term on the entropy of the barycenter, contrary to TreeDSB.

F.7.2 Details on the experiments

Synthetic Gaussian datasets. For each dimension that we consider, we generate three different triplets of random non-diagonal covariance matrices whose condition number is less than 10. We then run the algorithms on each triplet and aggregate the obtained results. The Gaussian datasets contain 1,500 samples for fsWB, and 10,000 samples for crWB and TreeDSB. We run fsWB with the following settings $(d, \varepsilon) \in \{(2, 0.1), (16, 0.2), (64, 0.5), (128, 1.0), (256, 2.0)\}$. We run TreeDSB for 10 mIPF cycles with regularization parameter $\varepsilon = 0.1$, starting from the central node initialized to a Gaussian distribution μ_0 chosen as detailed in Appendix F.6 with $\alpha = 1$. Thus, we solve the regularized Wasserstein barycenter problem (μ_0-regWB), which contains an additional regularization with respect to μ_0. This choice is justified, since the non-regularized barycenter is known to be a Gaussian distribution, and μ_0 can be seen as an *a priori* for the regularized barycenter. For each of the three settings, we keep the best result among the 30 mIPF iterations. In this setting, TreeDSB and crWB have roughly the same training time.

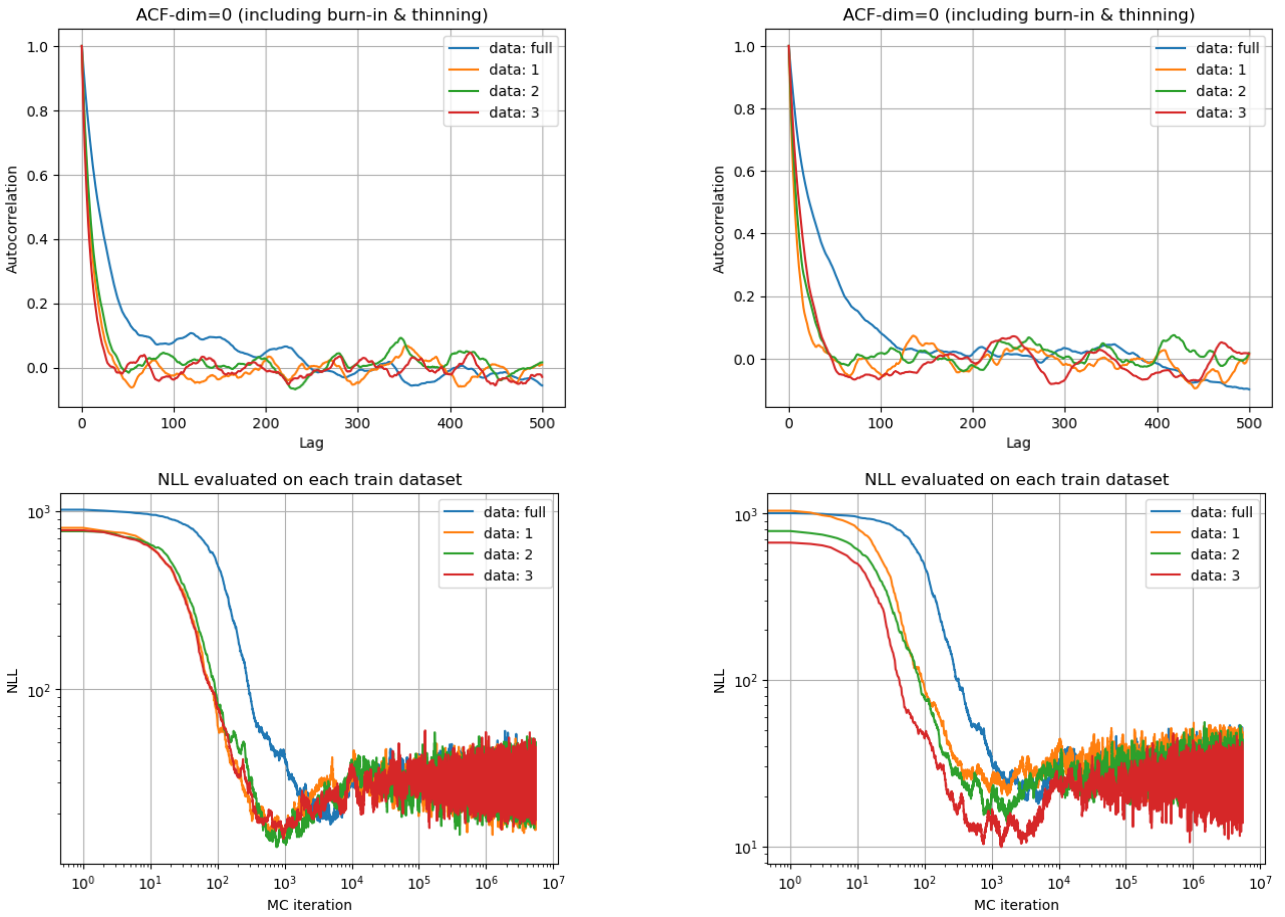


Figure F.2: Evaluation of the sampling process for `wine-homogeneous` (left) and `wine-heterogeneous` (right). We display the Autocorrelation function on 500 lags (above) and the evolution over the iterations of ULA of the negative log-likelihood (NLL) evaluated on each training dataset (below). In particular, the samples are decorrelated and the NLL has a satisfying profile.

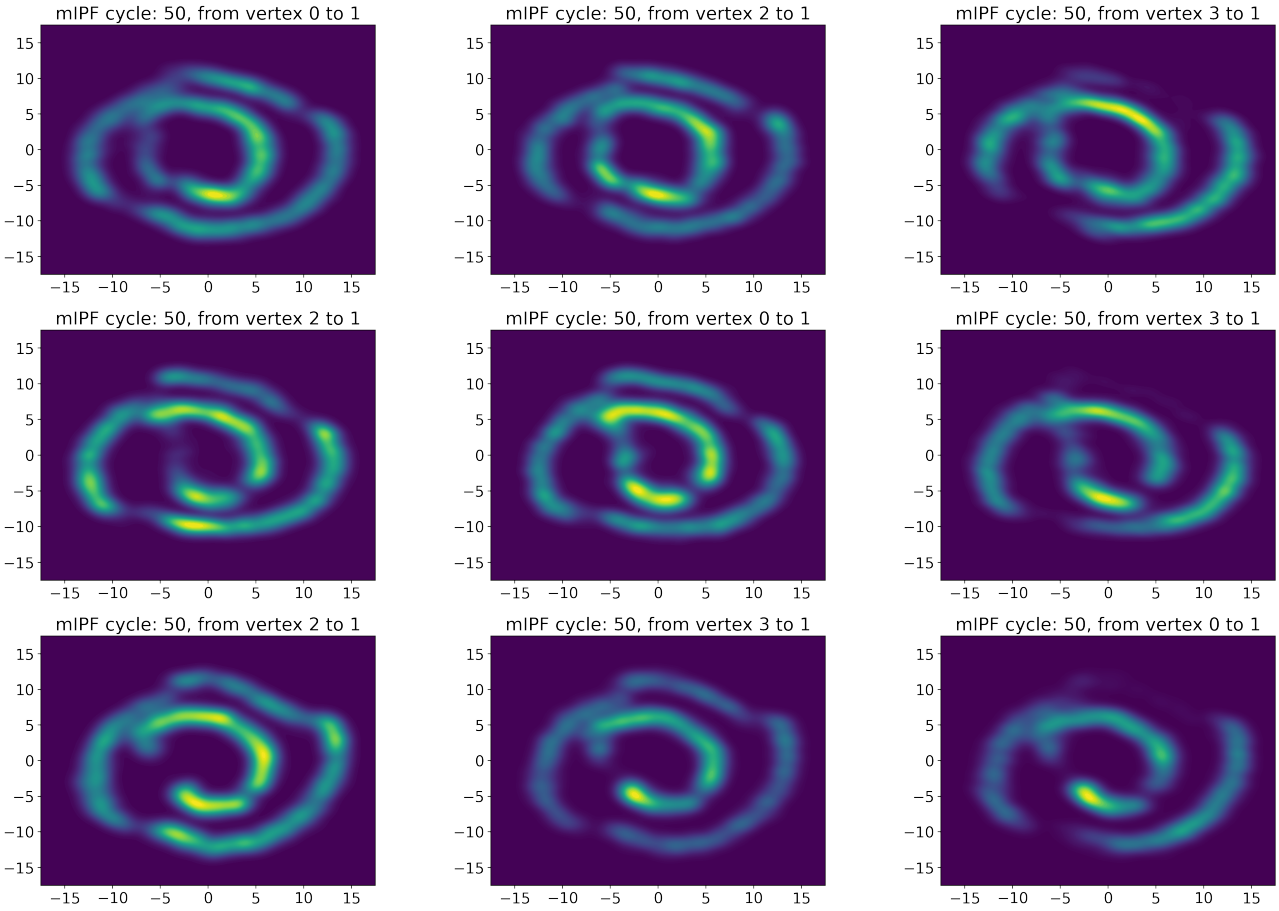


Figure F.3: Estimated 2D barycenter obtained by TreeDSB with $\varepsilon = 0.2$ (50 mIPF cycles). First row: starting from *Swiss-roll*. Second row: starting from *Circle*. Third row: starting from *Moons*.

Subset posterior aggregation. When considering a dataset splitted into several subdatasets, a common paradigm in bayesian inference consists in running Monte Carlo Markov Chain methods separately on these subdatasets, and then merge the obtained posteriors to recover the full posterior. The barycenter of these subdataset posteriors is proved to be close to the full data posterior under mild assumptions [Srivastava et al., 2018]. In our setting, we consider the posterior aggregation problem for the logistic regression model associated to the `wine` dataset[2] (d = 42) with 3 subdatasets. We consider here two splitting methods: (i) either, data is uniformly splitted between 3 subdatasets with respect to the label distribution, denoted by `wine-homogeneous`, or (ii) data is splitted with some heterogeneity according to a Dirichlet distribution whose parameter is randomly chosen, denoted by `wine-heterogeneous`. Following Korotin et al. [2021], we use the stochastic approximation trick so that the subset posterior samples do not vary consistently from the full posterior in covariance [Minsker et al., 2014]. We implement the Unadjusted Langevin Algorithm (ULA) to sample from each subdataset posterior and from the full posterior. In each case, we run ULA for $5.5 \cdot 10^6$ iterations with a well chosen step-size, and obtain 9,900 samples after applying a *burn-in* of order 10% and then a *thinning* of size 500. We provide in Figure F.2 some metrics which assess the quality of this sampling process. We recall that the the full posterior samples serve as ground truth in this experiment.

The results presented in Table 7.2 were computed as follows. For fsWB, we first subsample 1,500 samples out of the 9,900 samples from each posterior, and then run the algorithm with $\varepsilon = 0.5$. We repeat three times this procedure and then aggregate the results. In the case of crWB and TreeDSB, we run the algorithms three times with various seeds. Similarly to the Gaussian setting, we run TreeDSB for 10 mIPF cycles with regularization parameter $\varepsilon = 0.1$. We start from the central node with a Gaussian distribution μ_0 chosen as detailed in Appendix F.6 with $\alpha = 1$, and thus solve the barycenter formulation (μ_0-regWB). For each of the three settings, we keep the best result among the 30 IPF iterations. In this setting, TreeDSB and crWB have roughly the same training time.

[2] https://archive.ics.uci.edu/ml/datasets/wine

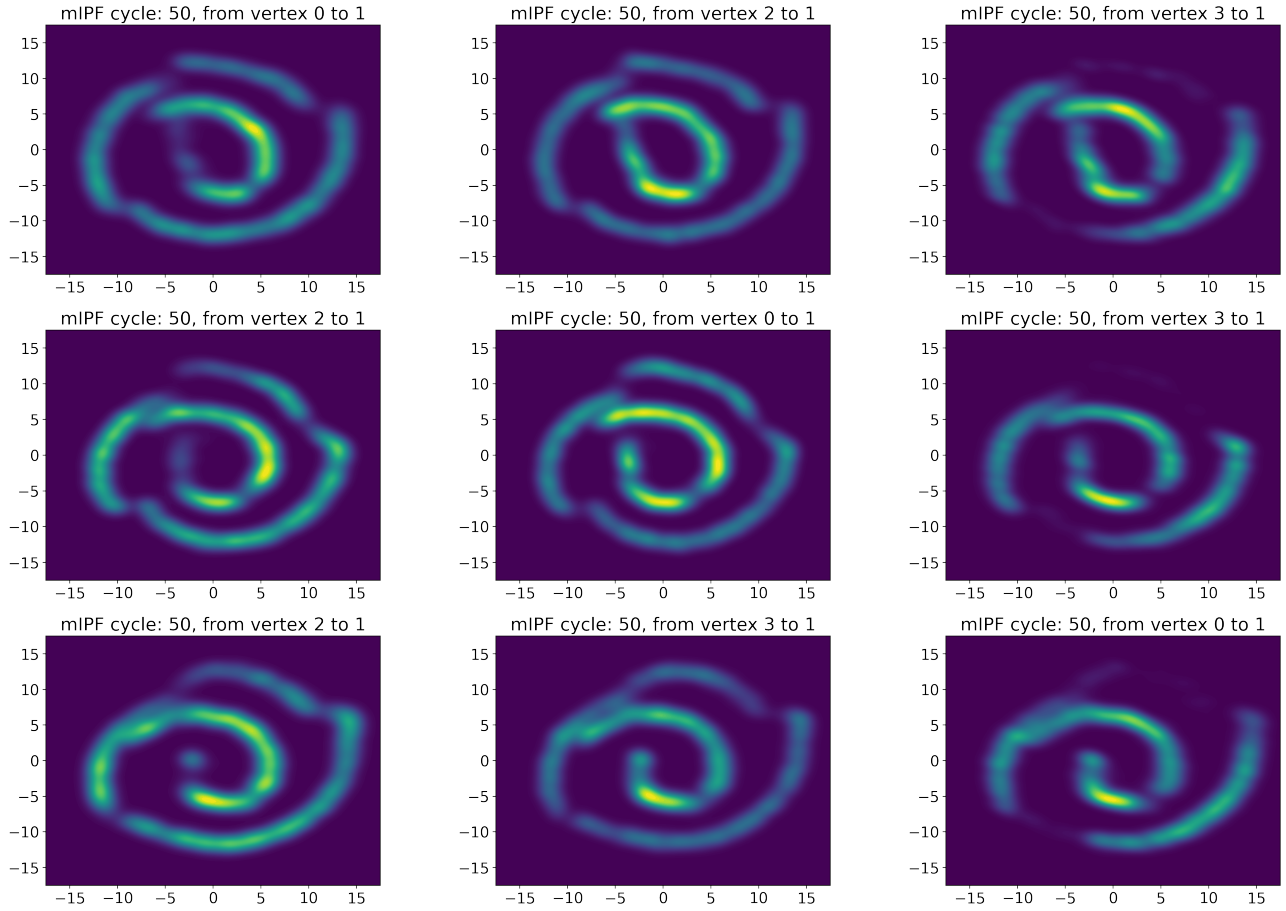


Figure F.4: Estimated 2D barycenter obtained by TreeDSB with $\varepsilon = 0.1$ (50 mIPF cycles). First row: starting from *Swiss-roll*. Second row: starting from *Circle*. Third row: starting from *Moons*.

Synthetic two-dimensional datasets. In this setting, we consider three different datasets (*Swiss-roll*, *Circle* and *Moons*) that each contain 10,000 samples. Since we do not have an *a priori* on the shape of the barycenter between these datasets, we consider the regularized Wasserstein barycenter problem (regWB), *i.e.*, r is chosen as a leaf and corresponds to one of the input datasets. We emphasize that this experiment is not intended to demonstrate the superiority of TreeDSB to compute 2D Wasserstein barycenters, but is rather meant to illustrate that (a) the marginals of the leaves are well recovered by the algorithm, see Figure 7.3, and that (b) the obtained barycenter is consistent when diffusing from the different leaves, see Figure 7.4. In all our experiments on 2D datasets, we observed that (a) was persistently verified without difficulty. In this section, we rather aim at illustrating (b) by providing additional results which assess the quality of the barycenter obtained by TreeDSB with respect to the choice of the starting leaf r and to the choice of the regularization parameter ε.

To do so, we consider three different choices of regularization in TreeDSB: (i) $\varepsilon = 0.2$ (50 mIPF cycles), see Figure F.3, (ii) $\varepsilon = 0.1$ (50 mIPF cycles), see Figure F.4 and (iii) $\varepsilon = 0.05$ (60 mIPF cycles), see Figure F.5. For each of these settings, we run TreeDSB with the starting leaf r chosen as *Swiss-roll* (first row), *Circle* (second row) or *Moons* (third row), and display the final barycenter obtained by diffusing from *Swiss-roll* (first column), *Circle* (second column) and *Moons* (third column). Note that the vertex 0 always corresponds to the starting leaf, the vertex 1 to the barycenter node and that Figure 7.4 corresponds to the first row of Figure F.4.

We can make the following observations. First, the estimated barycenter is always coherent within each row, which assesses the convergence of our method. Then, for each value of ε, the TreeDSB barycenter is rather consistent between the rows, *i.e.*, the choice of the starting leaf does not have a meaningful impact on our method. Finally, as expected, we observe that the support of the barycenter is less and less diffuse as long as ε decreases.

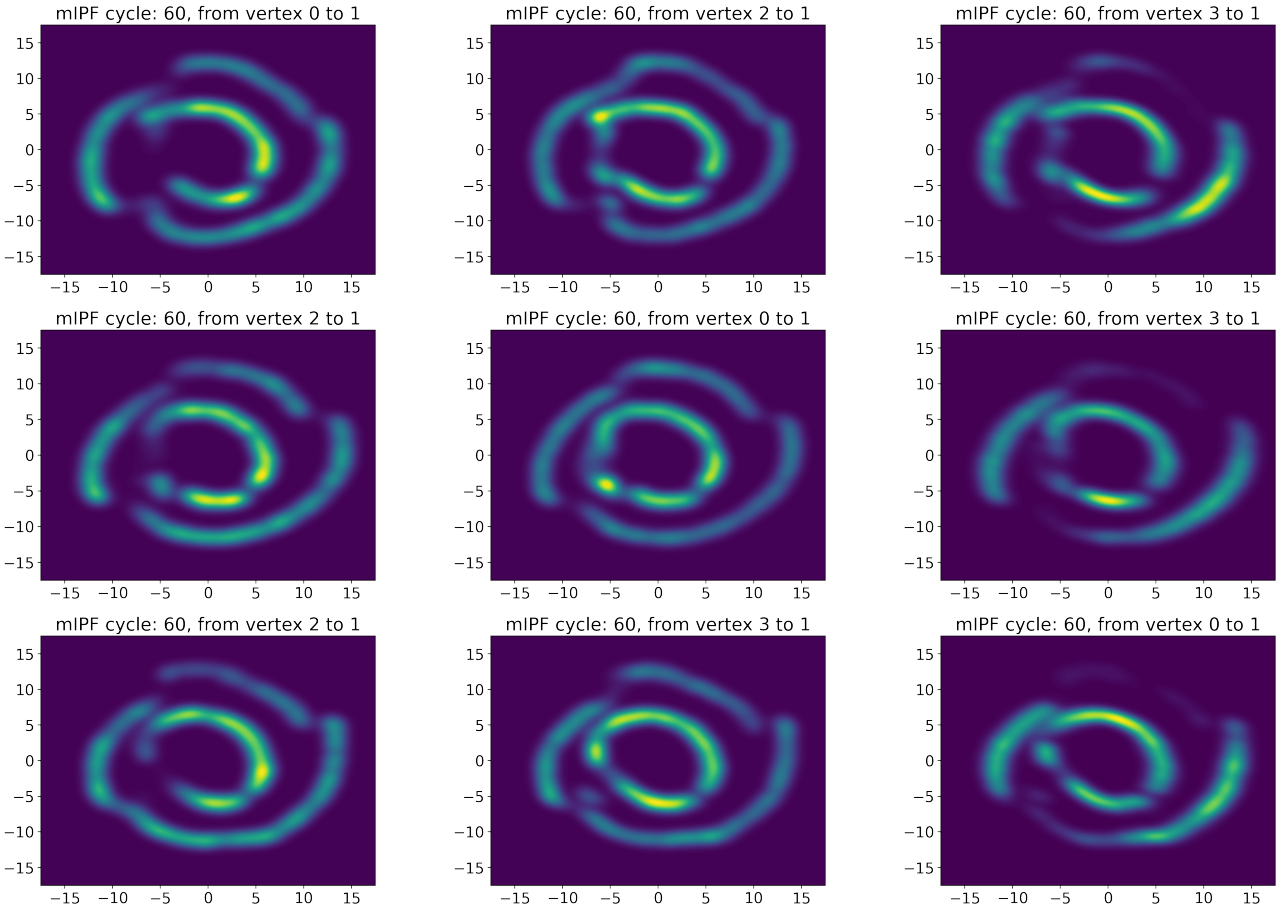


Figure F.5: Estimated 2D barycenter obtained by TreeDSB with $\varepsilon = 0.05$ (60 mIPF cycles). First row: starting from *Swiss-roll*. Second row: starting from *Circle*. Third row: starting from *Moons*.

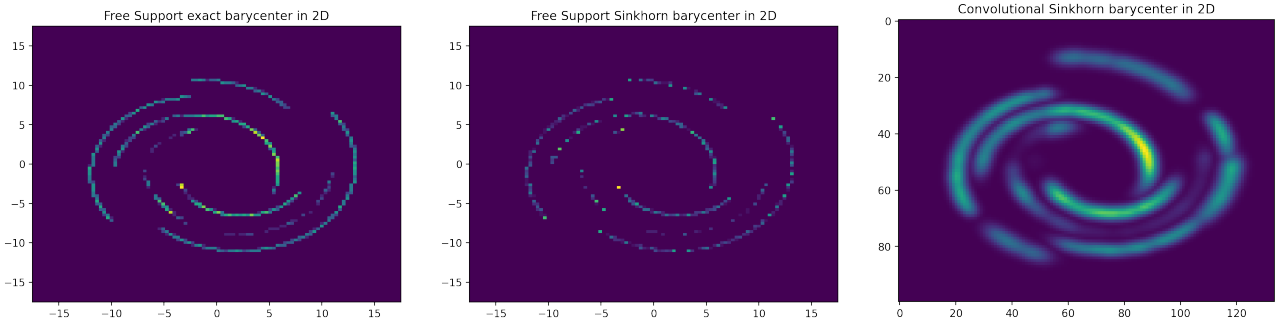


Figure F.6: Estimated 2D barycenter obtained by *in-sample* algorithms. From left to right: Cuturi and Doucet [2014] (non-regularized), Cuturi and Doucet [2014] (regularized), Solomon et al. [2015].

For purpose of illustration, we provide in Figure F.6 the barycenter obtained by state-of-the-art two-dimensional *in-sample* methods that are available in POT library [Flamary et al., 2021]: (i) non-regularized free-support Wasserstein barycenter [Cuturi and Doucet, 2014], (ii) entropic-regularized free-support Wasserstein barycenter (fsWB) with $\varepsilon = 0.5$ [Cuturi and Doucet, 2014] and (iii) entropic-regularized convolutional Wasserstein barycenter with $\varepsilon = 5.10^{-4}$ [Solomon et al., 2015], which is specifically designed for images. We notably observe that TreeDSB cannot capture the full complexity of the 2D barycenter compared to these methods. We infer that this gap comes from the *dynamic* nature of TreeDSB, since increasing the number of training iterations per IPF iteration or improving the complexity of the neural networks did not bring any significant change in our results. Finally, we recall that the methods (i), (ii) and (iii) do not scale well with dimension, and have to be completely run again when new data samples are available, contrary to TreeDSB.

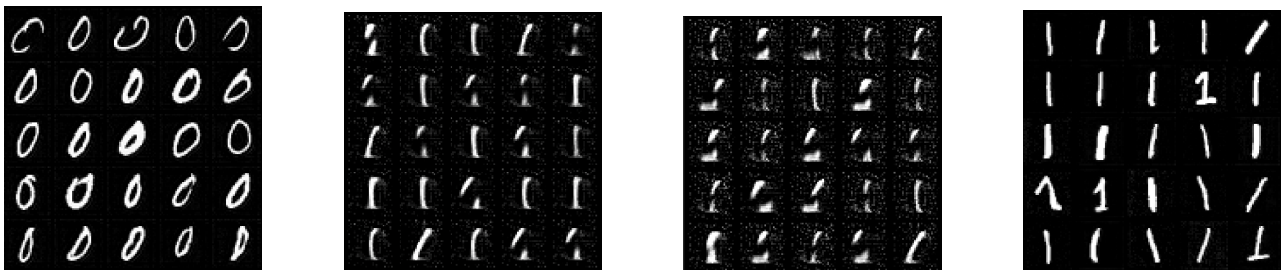


Figure F.7: Tree DSB results for MNIST digits 0 and 1, after 15 mIPF cycles with $\varepsilon = 0.5$.

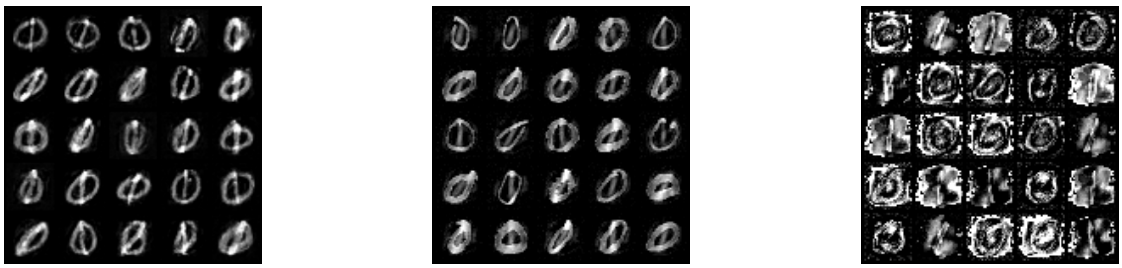


Figure F.8: From left to right: Fan et al. [2021], Korotin et al. [2021] and Li et al. [2020].

MNIST Wasserstein barycenter. This setting can be qualified as *high-dimensional*, since the data dimension is $d = 784$. Here, each digit dataset contains 1,000 samples. As in the two-dimensional setting, we do not have an *a priori* on the shape of the barycenter between MNIST digits, and thus consider the formulation (regWB), where the root r is chosen as a leaf. We propose below several experiments to assess the scalability of TreeDSB to this setting.

Digits 0 and 1. In Figure F.7, we report the results obtained by running TreeDSB on MNIST digits 0 and 1, for 15 mIPF cycles with $\varepsilon = 0.5$, starting from the leaf MNIST-0. We display 25 samples from the reconstructed MNIST-0 marginal (first column), from the reconstructed MNIST-1 marginal (fourth column), from the estimated barycenter by diffusing from MNIST-0 (second column) and diffusing from MNIST-1 (third column). We notably observe that the digits are well recovered and that the barycenter samples are consistent. We draw the reader's attention to the fact that TreeDSB showed numerical unstability with a regularization value ε lower than 0.5. For purpose of illustration, we display in Figure F.8 the Wasserstein barycenter obtained by *non-regularized* methods from Fan et al. [2021] and Korotin et al. [2021], and by the *regularized* approach from Li et al. [2020].

Digits 2,4 and 6. In Figure F.9, we report the results obtained by running TreeDSB on MNIST digits 2,4 and 6, for 10 mIPF cycles with $\varepsilon = 0.5$. Here, we consider three settings which differ by the starting leaf r in the algorithm: MNIST-2 (first row), MNIST-4 (second row), or MNIST-6 (third row). For each of these settings, we display 30 samples from the estimated barycenter by diffusing from MNIST-2 (first column), diffusing from MNIST-4 (third column) and diffusing from MNIST-6 (third column). We notably observe a global consistency of the barycenter samples across the various settings. In Figure F.10, we report the results obtained by running TreeDSB on MNIST digits 2,4 and 6, for 10 mIPF cycles with $\varepsilon = 0.2$, starting from MNIST-6. We display 30 samples from the reconstructed marginals (first row), from the estimated barycenter (second row) by diffusing from MNIST-2 (first column), diffusing from MNIST-4 (second column) and diffusing from MNIST-6 (third column). As expected, we observe less noisy barycenter samples compared to Figure F.9, while still well recovering MNIST digits.

Digits 0,1 and 4. In Figure F.11, we report the results obtained by running TreeDSB on MNIST digits 0,1 and 4, for 10 mIPF cycles with $\varepsilon = 0.5$. We consider two settings which differ by the starting leaf r in the algorithm: MNIST-0 (second row) and MNIST-1 (first/third rows), for which we display samples from the reconstructed measures (first row). In Figure F.12, we report the results obtained by running TreeDSB on MNIST digits 0,1 and 4, for 10 mIPF cycles with $\varepsilon = 0.2$. We consider two settings which differ by the starting leaf r in the algorithm: MNIST-0 (first/second row), for which display samples from the reconstructed measures (first row), and MNIST-1 (third row). For all of these settings, we display 30 samples from the estimated barycenter by diffusing from MNIST-0 (first column), diffusing from MNIST-1 (third column) and diffusing from MNIST-4 (third column). Similarly to the digits 2-4-6, we observe consistency within the barycenter samples, unconditionally to the starting leaf, and less noise as ε decreases. Note that the reconstructed MNIST digits are less truthful to the original datasets when ε is low.

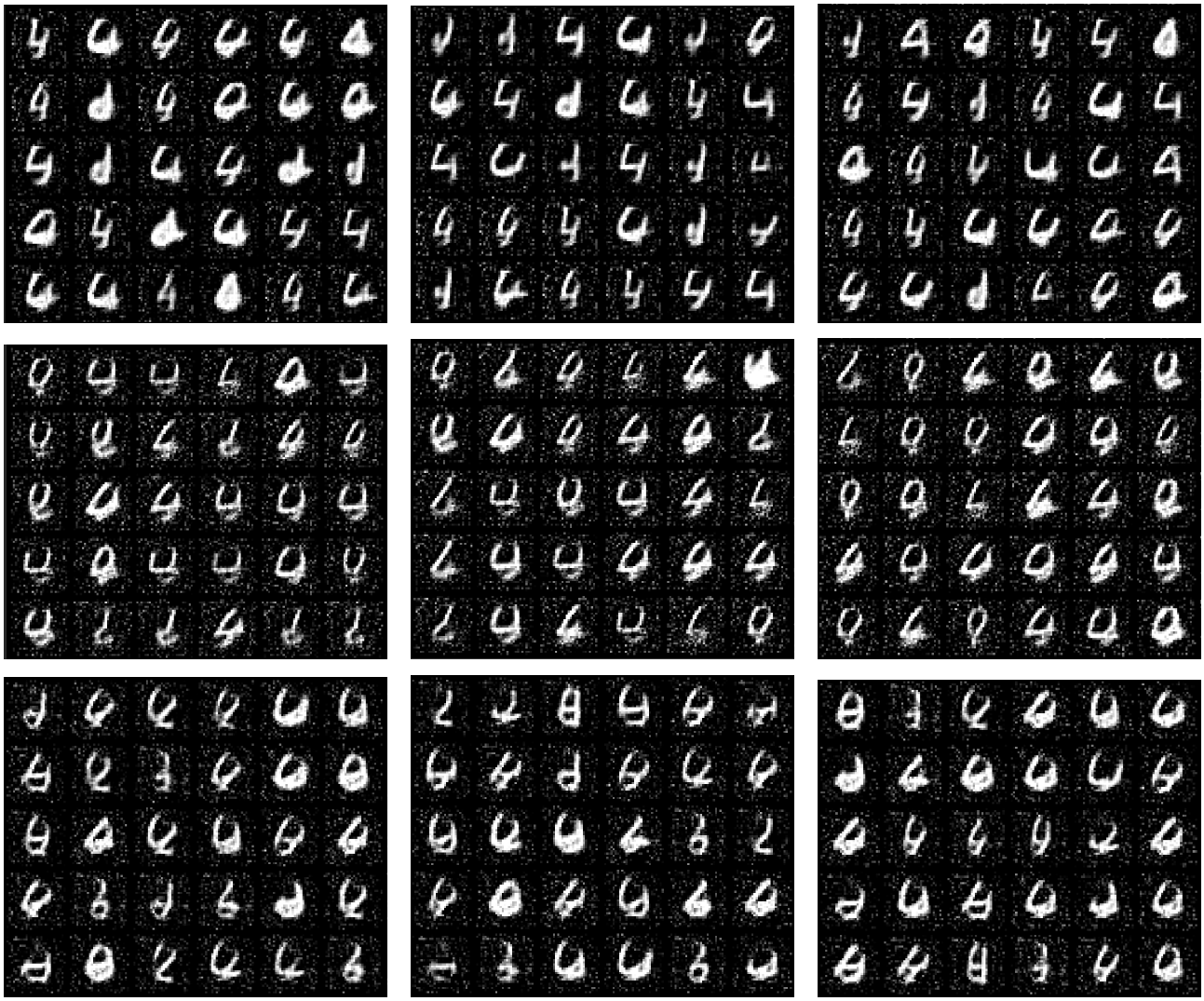


Figure F.9: Tree DSB results for MNIST digits 2,4 and 6, after 10 mIPF cycles with $\varepsilon = 0.5$. First row: starting from MNIST-2. Second row: starting from MNIST-4. Third row: starting from MNIST-6.

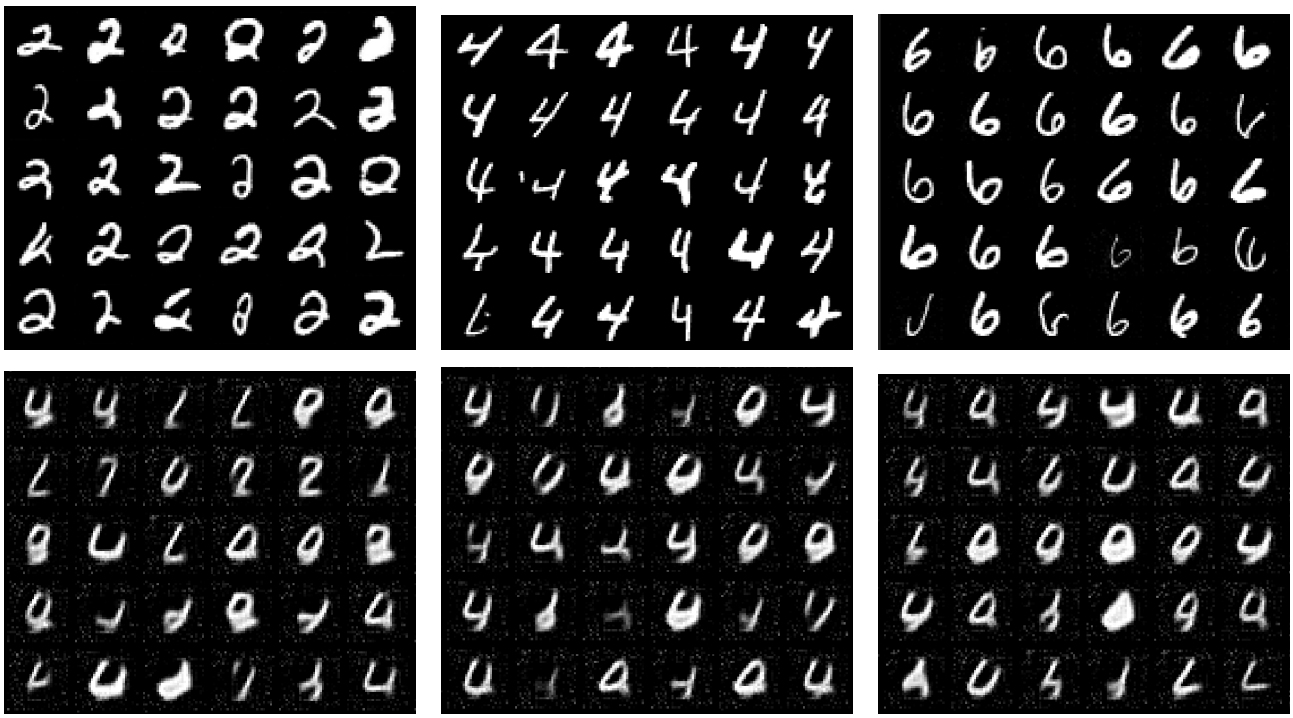


Figure F.10: Tree DSB results for MNIST digits 2,4 and 6, after 10 mIPF cycles with $\varepsilon = 0.2$, starting from MNIST-6. First row: samples from the reconstructed marginals. Second row: samples from the estimated barycenter.

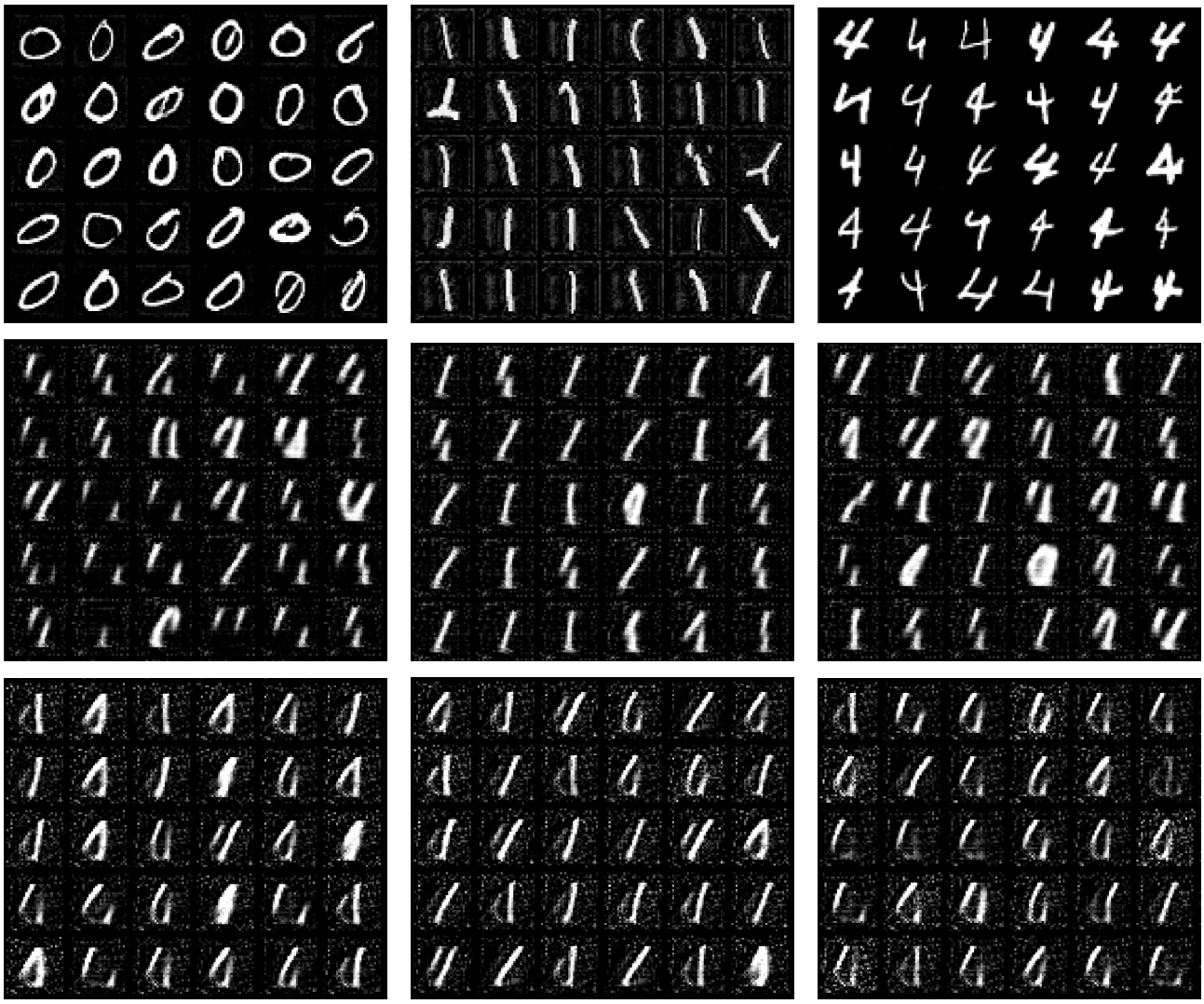


Figure F.11: Tree DSB results for MNIST digits 0,1 and 4, after 10 mIPF cycles with $\varepsilon = 0.5$. First row: samples from the reconstructed marginals, starting from MNIST-1. Second row: samples from the estimated barycenter, starting from MNIST-0. Third row: samples from the estimated barycenter, starting from MNIST-1.

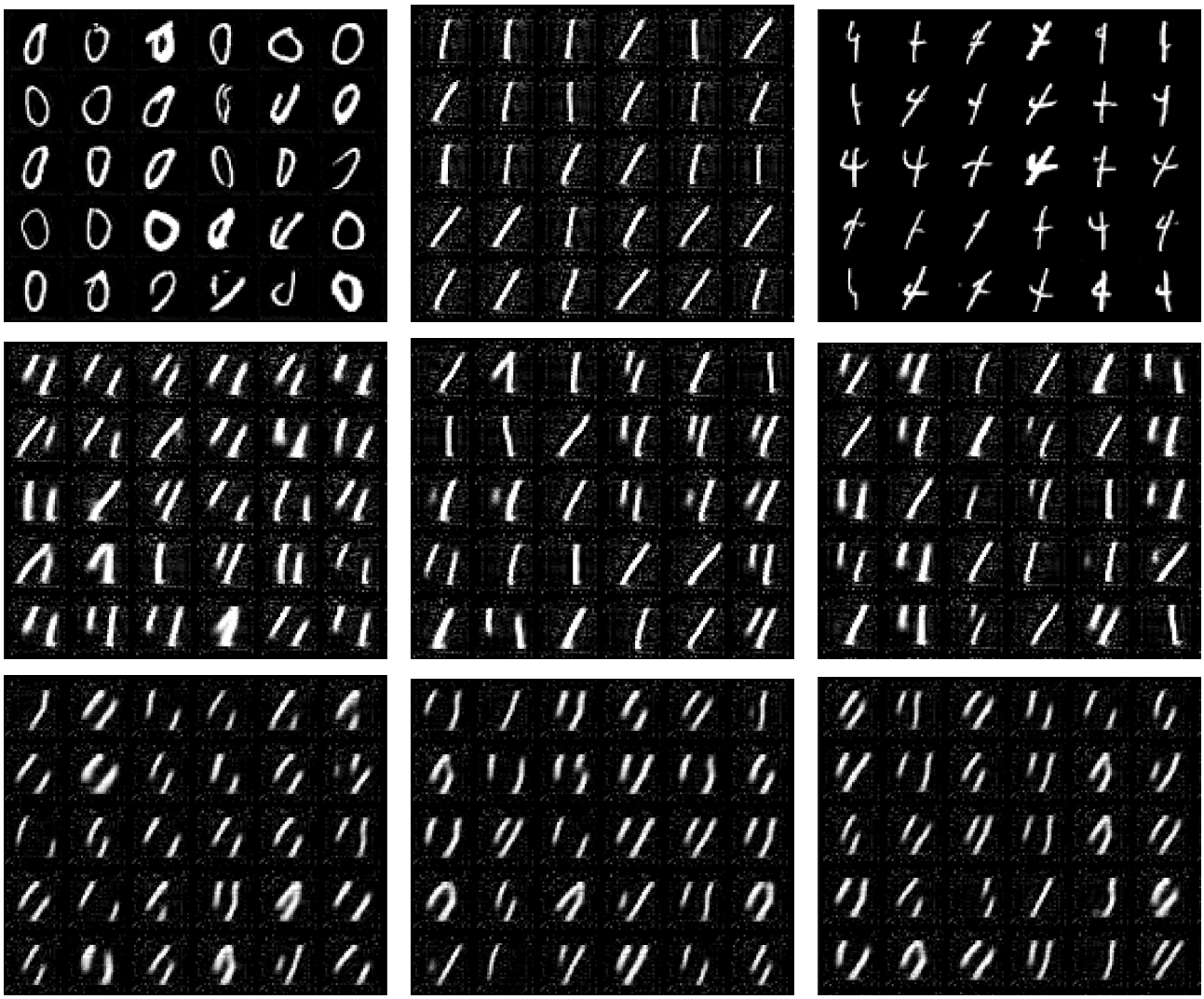


Figure F.12: Tree DSB results for MNIST digits 0,1 and 4, after 10 mIPF cycles with $\varepsilon = 0.2$. First row: samples from the reconstructed marginals, starting from MNIST-0. Second row: samples from the estimated barycenter, starting from MNIST-0. Third row: samples from the estimated barycenter, starting from MNIST-1.

Appendix G

Appendix of Chapter 8

The appendix is organized as follows. Appendix G.1 gathers complementary material for the continuous-time framework underlying the TSB problem and the associated TSBM methodology, while the discrete-time counterpart is presented in Appendix G.2.

G.1 Continuous-time setting

Throughout this section, we work on the canonical measurable space $(\mathbb{R}^d, \mathcal{B}(\mathbb{R}^d))$, where $\mathcal{B}(\mathbb{R}^d)$ denotes the Borel σ-field of $\mathbb{R}^d$. On this space, we consider a d-dimensional Brownian motion $(\mathbf{B}_t)_{t\geq 0}$. We define $\mathbf{A}_T$, the set of admissible control functions, by

$$\mathbf{A}_T = \{u \in \mathrm{C}^1([0,T]\times\mathbb{R}^d, \mathbb{R}^d) : \exists C > 0 : \forall (t,x) \in [0,T]\times\mathbb{R}^d, \|u_t(x)\| \leq C(1+\|x\|)\} \,.$$

We first recall that for any $(x_0, x_T) \in \mathbb{R}^d \times \mathbb{R}^d$, by simple application of Doob's h-transform [Rogers and Williams, 2000], the scaled Brownian bridge $\mathbb{Q}_{|0,T}(\cdot|x_0, x_T)$ is a pinned diffusion process equivalently associated to the forward and backward SDEs

$$\mathrm{d}\mathbf{X}_t^{0,T} = u_t^f(\mathbf{X}_t^{0,T})\mathrm{d}t + \sigma \mathrm{d}\mathbf{B}_t \,,\ \mathbf{X}_0^{0,T} = x_0 \,,\ \mathbf{X}_T^{0,T} = x_T \,, \quad u_t^f(x) = \sigma^2 \nabla_x \log \mathbb{Q}_{T|t}(x_T|x) = \frac{x_T - x}{T-t} \,, \tag{G.1}$$

$$\mathrm{d}\mathbf{Y}_t^{0,T} = u_{T-t}^b(\mathbf{Y}_t^{0,T})\mathrm{d}t + \sigma \mathrm{d}\tilde{\mathbf{B}}_t \,,\ \mathbf{Y}_0^{0,T} = x_T \,,\ \mathbf{Y}_T^{0,T} = x_0 \,, \quad u_t^b(x) = \sigma^2 \nabla_x \log \mathbb{Q}_{t|0}(x|x_0) = \frac{x_0 - x}{t} \,. \tag{G.2}$$

G.1.1 Results on twisted measures

We begin by recalling the following result from [Léonard, 2011, Proposition 4.2] concerning the twisted Brownian motion $\mathbb{Q}^V$ defined in (8.11).

Proposition 132. *Let $V \in \mathrm{C}^{0,1}([0,T]\times\mathbb{R}^d, \mathbb{R})$ be such that $Z_V < \infty$. Then it holds that (i) $\mathbb{Q}^V$ is Markov, (ii) for any $0 \leq s \leq t \leq T$, we have*

$$\mathrm{d}\mathbb{Q}^V_{s:t}(x_{[s:t]}) \propto \exp\left(-\textstyle\int_s^t \frac{V_u(x_u)}{\sigma^2}\mathrm{d}u\right) \mathrm{d}\mathbb{Q}_{s:t}(x_{[s:t]}) \,.$$

As a direct consequence of Proposition 132, we obtain the following unnormalized Radon–Nikodym densities:

$$\mathrm{d}\mathbb{Q}^V_{t:T|t}(x_{(t:T]}|x_t) \propto \exp\left(-\textstyle\int_t^T \frac{V_u(x_u)}{\sigma^2}\mathrm{d}u\right) \mathrm{d}\mathbb{Q}_{t:T|t}(x_{(t:T]}|x_t) \,,$$
$$\mathrm{d}\mathbb{Q}^V_{0:t|t}(x_{[0:t)}|x_t) \propto \exp\left(-\textstyle\int_0^t \frac{V_u(x_u)}{\sigma^2}\mathrm{d}u\right) \mathrm{d}\mathbb{Q}_{0:t|t}(x_{[0:t)}|x_t) \,,$$
$$\mathrm{d}\mathbb{Q}^V_{|0,T}(x_{(0:T)}|x_0, x_T) \propto \exp\left(-\textstyle\int_0^T \frac{V_u(x_u)}{\sigma^2}\mathrm{d}u\right) \mathrm{d}\mathbb{Q}_{|0,T}(x_{(0:T)}|x_0, x_T) \,.$$

These identities suggest that the twisted measure $\mathbb{Q}^V$ and its bridge $\mathbb{Q}^V_{|0,T}$ should admit stochastic differential representations. More precisely, we now show that they are associated with affine-controlled Brownian motions. To this end, we first establish a general result from which the desired statements follow as corollaries, see Corollary 134 and Corollary 135.

Proposition 133. *Consider $\mathbb{Q} \in \mathcal{M}_T$, associated to a SDE $\mathrm{d}X_t = f_t(X_t)\mathrm{d}t + \sigma \mathrm{d}B_t$, $X_0 \sim \mathbb{Q}_0$, where $f \in \mathrm{C}^1([0,T]\times\mathbb{R}^d, \mathbb{R}^d)$ satisfies a linear growth condition, i.e., there exists $C > 0$ such that for any $(t,x) \in [0,T]\times\mathbb{R}^d$, we have $\|f_t(x)\| \leq C(1 + \|x\|)$. Let $V \in \mathrm{C}^{0,1}([0,T] \times \mathbb{R}^d, \mathbb{R})$ be such that for any $t \in [0,T]$ and any $\mathbb{Q}_t$-almost surely $x \in \mathbb{R}^d$, we have*

$$Z_V(t,x) = \mathbb{E}_{\mathbb{Q}_{|t}}\left[\exp\left(-\int_t^T \tfrac{V_s(\mathbf{X}_s)}{\sigma^2}\mathrm{d}s\right) | \mathbf{X}_t = x\right] < \infty \, .$$

Then, the path measure $\mathbb{Q}^V$, defined in (8.11), is Markov, associated with the following affine-controlled SDE

$$\mathrm{d}\mathbf{X}_t = \{f_t(\mathbf{X}_t) + \sigma u_t^V(\mathbf{X}_t)\}\mathrm{d}t + \sigma \mathrm{d}\mathbf{B}_t \, , \; \mathbf{X}_0 \sim \mathbb{Q}_0^V \quad \textit{where} \quad u_t^V(x) = \sigma \nabla_x \log Z_V(t,x) \, . \tag{G.3}$$

Moreover, for any arbitrary continuously differentiable function $m_t : [t,T] \to \mathbb{R}$ such that $m_t(t) = 1$, the control u_t^V admits a closed-form solution depending on the potential V via the path-wise reparameterization trick *[Domingo i Enrich et al., 2024] as*

$$u_t^V(x) = \frac{\sigma}{Z_V(t,x)}\mathbb{E}_{\mathbb{Q}_{|t}}\left[\left\{-\int_t^T m_t(s)\frac{\nabla V_s(\mathbf{X}_s)}{\sigma^2}\mathrm{d}s + \frac{1}{\sigma}\int_t^T (m_t(s)\nabla f_s(\mathbf{X}_s) - \partial_s m_t(s)\mathrm{I}_d)\,\mathrm{d}\mathbf{B}_s\right\}\right. \tag{G.4}$$

$$\left. \times \exp\left(-\int_t^T \frac{V_s(\mathbf{X}_s)}{\sigma^2}\mathrm{d}s\right) \middle| \mathbf{X}_t = x\right] \, . \tag{G.5}$$

Proof. Let $x_0 \sim \mathbb{Q}_0^V$ and consider the following minimization problem over path measures:

$$\arg\min\{\mathrm{KL}(\mathbb{M}_{|0}(\cdot|x_0)\|\mathbb{Q}_{|0}^V(\cdot|x_0)) \; : \; \mathbb{M} \in \mathcal{M}_{T,c}\} \tag{G.6}$$

where $\mathcal{M}_{T,c}$ is the set of Markov measures that are associated with controlled SDEs of the form

$$\mathrm{d}\mathbf{X}_t^u = f_t(\mathbf{X}_t^u)\mathrm{d}t + u_t(\mathbf{X}_t^u)\mathrm{d}t + \sigma \mathrm{d}B_t \, , \; \mathbf{X}_0^u = x_0 \, ,$$

where $u \in \mathbf{A}_T$. Let $\mathbb{M}^V$ denote the solution to (G.6).

By Girsanov's formula, we have for any $\mathbb{M} \in \mathcal{M}_{T,c}$, the following KL decomposition

$$\mathrm{KL}(\mathbb{M}_{|0}(\cdot|x_0)\|\mathbb{Q}_{|0}^V(\cdot|x_0)) = J(u;x_0,0) + \log Z_V \, ,$$

where

$$J(u;x,t) = \frac{1}{\sigma^2}\mathbb{E}\left[\int_t^T \left\{\frac{1}{2}\|u_t(X_t^u)\|^2 + V_t(X_t^u)\right\}\mathrm{d}t \mid X_t^u = x\right] \, .$$

Hence, solving (G.6) is equivalent to solving the following optimization problem

$$\arg\min\{J(u;x_0,0) \; : \; u \in \mathbf{A}_T\} \, . \tag{G.7}$$

In fact, this is a SOC problem associated to the cost functional $(u,x,t) \mapsto J(u;x,t)$. Combining [Nüsken and Richter, 2021, Theorem 2.2] and [Nüsken and Richter, 2021, Remark 2.6], we deduce under the stated assumptions that the unique minimizer u^V satisfies

$$u_t^V(x) = \sigma \nabla_x \log Z_V(t,x) \quad \text{and} \quad J(u^V;x_0,0) = -\log Z_V \, .$$

In particular, for any x_0, $\mathbb{Q}_0^V$-almost surely,

$$\mathrm{KL}(\mathbb{M}_{|0}^V(\cdot|x_0)\|\mathbb{Q}_{|0}^V(\cdot|x_0)) = 0 \, ,$$

which proves the result (G.3). The representation (G.4) follows directly from [Domingo i Enrich et al., 2024, Proposition 1]. □

Notably, Proposition 133 extends [Léonard, 2011, Theorem 5.4], where $\mathbb{Q}$ is assumed to be stationary, an assumption that does not hold for Brownian motion.

Corollary 134 (Application to the Brownian motion.)**.** *If $\mathbb{Q}$ is the distribution of the scaled Brownian motion $(\sigma B_t)_{t\in[0,T]}$, then $\mathbb{Q}^V$ is associated with*

$$\mathrm{d}\mathbf{X}_t = \mathbf{u}_t^V(\mathbf{X}_t)\mathrm{d}t + \sigma \mathrm{d}\mathbf{B}_t \ , \ \mathbf{X}_0 \sim \mathbb{Q}_0^V \ , \tag{G.8}$$

where

$$\mathbf{u}_t^V(x) = \mathbb{E}_{\mathbb{Q}^V_{|t}}\left[-\int_t^T \nabla V_s(\mathbf{X}_s)\mathrm{d}s | \mathbf{X}_t = x\right] \tag{G.9}$$

Proof. This is an immediate application of Proposition 133 in the case where $f \equiv 0$ and $m_t \equiv 1$ for any $t \in [0,T]$. □

Corollary 135 (Application to the *pinned* Brownian motion.)**.** *Let $(x_0, x_T) \in \mathbb{R}^d \times \mathbb{R}^d$. If $\mathbb{Q}_{|0,T}(\cdot|x_0,x_T)$ is the distribution of the scaled Brownian motion $(\sigma B_t)_{t\in[0,T]}$* pinned *down at initial and terminal points (x_0,x_T), then $\mathbb{Q}^V(\cdot|x_0,x_T)$ is equivalently associated with the forward and backward SDEs*

$$\mathrm{d}\mathbf{X}_t = \sigma^2\nabla_{\mathbf{X}_t}\log\mathbb{Q}_{T|t}(x_T|\mathbf{X}_t) + \mathbf{u}_t^{f,V}(\mathbf{X}_t|x_T)\mathrm{d}t + \sigma\mathrm{d}\mathbf{B}_t \ , \mathbf{X}_0 = x_0 \ , \tag{G.10}$$

$$\mathrm{d}\mathbf{Y}_t = \sigma^2\nabla_{\mathbf{Y}_t}\log\mathbb{Q}_{T-t|0}(\mathbf{Y}_t|x_0) + \mathbf{u}_{T-t}^{b,V}(\mathbf{Y}_t|x_0)\mathrm{d}t + \sigma\mathrm{d}\tilde{\mathbf{B}}_t \ , \mathbf{Y}_0 = x_T \ , \tag{G.11}$$

where

$$\mathbf{u}_t^{f,V}(x|x_T) = \mathbb{E}_{\mathbb{Q}^V_{|t,T}}\left[-\int_t^T \frac{T-s}{T-t}\nabla V_s(\mathbf{X}_s)\mathrm{d}s | \mathbf{X}_t = x, \mathbf{X}_T = x_T\right] \ ,$$

$$\mathbf{u}_t^{b,V}(x|x_0) = \mathbb{E}_{\mathbb{Q}^V_{|0,t}}\left[-\int_0^t \frac{s}{t}\nabla V_s(\mathbf{X}_s)\mathrm{d}s | \mathbf{X}_0 = x_0, \mathbf{X}_t = x\right] \ .$$

Proof. Obtaining the forward SDE results from the application of Proposition 133 when considering the pinned Brownian bridge (in forward time direction) as base process $\mathbb{Q}$, with drift given in (G.1), and by choosing $m_t(s) = \frac{T-s}{T-t}$, which simplifies the stochastic integral term in (G.4). To obtain the backward SDE, we apply Proposition 133 to the backward pinned Brownian bridge $(\mathbb{Q})^R$, whose SDE is exactly given in (G.2), considering as twisted path measure $\tilde{\mathbb{Q}}^V(y_{[0:T]}) \propto \exp(-\int_0^T V_{T-s}(y_s)\mathrm{d}s)(\mathbb{Q})^R(y_{[0:T]})$ and taking once again $m_t(s) = \frac{T-s}{T-t}$. □

G.1.2 Results on the Reciprocal projection

Proof of Proposition 18. Consider such variational family. Then for any $\psi \in \Psi$, and any $(x_0,x_T) \in \mathbb{R}^d \times \mathbb{R}^d$, we have, $\mathbb{P}_{0,T}$-almost surely,

$$\begin{aligned}
&\mathrm{KL}(\mathbb{Q}^\psi_{|0,T}(\cdot|x_0,x_T)\|\mathbb{Q}^V_{|0,T}(\cdot|x_0,x_T)) \\
&= \mathrm{KL}(\mathbb{Q}^\psi_{|0,T}(\cdot|x_0,x_T)\|\mathbb{Q}_{|0,T}(\cdot|x_0,x_T)) + \int_0^T \frac{1}{\sigma^2}\mathbb{E}_{\mathbb{Q}^\psi_{t|0,T}}[V_t(\mathbf{X}_t)|\mathbf{X}_0 = x_0, \mathbf{X}_T = x_T]\mathrm{d}t + \mathbf{c} \\
&= \frac{1}{\sigma^2}\int_0^T \mathbb{E}_{\mathbb{Q}^\psi_{t|0,T}}\left[\frac{1}{2}\|u_t^\psi(\mathbf{X}_t|\mathbf{X}_0,\mathbf{X}_T) - \sigma^2\nabla\log\mathbb{Q}_{T|t}(\mathbf{X}_T|\mathbf{X}_t)\|^2 + V_t(\mathbf{X}_t)|\mathbf{X}_0 = x_0, \mathbf{X}_T = x_T\right]\mathrm{d}t + \mathbf{c}
\end{aligned}$$

where $\mathbf{c}$ is a constant independent of ψ (namely, the normalizing constant of $\mathbb{Q}^V_{|0,T}(\cdot|x_0,x_T)$), and the second equality comes from Girsanov's theorem application exploiting (G.1). Therefore, we obtain the forward objective identity

$$\mathbb{E}_{\mathbb{P}_{0,T}}[\mathrm{KL}(\mathbb{Q}^\psi_{|0,T}\|\mathbb{Q}^V_{|0,T})] = \frac{1}{\sigma^2}\mathcal{L}^f_{\mathcal{R}}(\psi) + \mathbf{c} \ .$$

Similarly, by exploiting (G.2), we can obtain the backward objective identity

$$\mathbb{E}_{\mathbb{P}_{0,T}}[\mathrm{KL}(\mathbb{Q}^\psi_{|0,T}\|\mathbb{Q}^V_{|0,T})] = \frac{1}{\sigma^2}\mathcal{L}^b_{\mathcal{R}}(\psi) + \mathbf{c} \ .$$

The result of the proposition immediately follows these observations. □

In this section, we detail the construction of the variational family based on the stochastic interpolants introduced by [Albergo et al., 2023]. We assume access to conditioning pairs $(x_0,x_T) \sim \mathbb{P}_{0,T}$ and emphasize that the families described below can be straightforwardly parallelized across multiple conditioning pairs in large data batches.

Gaussian variational family. We say that the stochastic process $\mathbb{P}_{|0,T}(\cdot|x_0, x_T)$ is a Gaussian stochastic interpolant between x_0 and x_T [Albergo et al., 2023] if its marginals satisfy

$$\begin{aligned}\mathbb{P}_{t|0,T}(\cdot|x_0, x_T) &= \mathrm{N}(I_t, \gamma_t^2 \mathrm{I}_d) \ , \\ \text{where } I \in \mathcal{I}(x_0, x_T) &= \{I \in \mathrm{C}^1([0,T], \mathbb{R}^d) \ : \ I_0 = x_0 \ , I_T = x_T\} \ , \\ \gamma \in \Gamma &= \{\gamma \in \mathrm{C}^1([0,T], (0,\infty)) \ : \ \gamma_0 = \gamma_T = 0\} \ .\end{aligned}$$

We denote by $\mathrm{SI}(x_0, x_T)$ the resulting class of stochastic interpolants.

For any $I \in \mathcal{I}(x_0, x_T)$ and $\gamma \in \Gamma$, we define the time-dependent velocity fields $(t,x) \mapsto v_t^f(x|x_0, x_T)$ (forward direction) and $(t,x) \mapsto v_t^b(x|x_0, x_T)$ (backward direction) by

$$v_t^f(x|x_0, x_T) = \partial_t I_t + \left(\frac{\partial_t \gamma_t}{\gamma_t} - \frac{\sigma^2}{2\gamma_t^2}\right)(x - I_t) \ , \quad v_t^b(x|x_0, x_T) = -\partial_t I_t + \left(-\frac{\partial_t \gamma_t}{\gamma_t} - \frac{\sigma^2}{2\gamma_t^2}\right)(x - I_t) \ .$$

As shown in Albergo et al. [2023], the SDEs

$$\mathrm{d}\mathbf{X}_t^{0,T} = v_t^f(\mathbf{X}_t^{0,T}|x_0, x_T)\mathrm{d}t + \sigma \mathrm{d}\mathbf{B}_t \ , \mathbf{X}_0^{0,T} = x_0 \ , \tag{G.12}$$

$$\mathrm{d}\mathbf{Y}_t^{0,T} = v_{T-t}^b(\mathbf{Y}_t^{0,T}|x_0, x_T)\mathrm{d}t + \sigma \mathrm{d}\mathbf{B}_t \ , \mathbf{Y}_0^{0,T} = x_T \ , \tag{G.13}$$

induce identical path measures up to time reversal, and share the same marginals as $\mathbb{P}_{|0,T}(\cdot|x_0, x_T)$. In particular, the boundary conditions are satisfied almost surely, namely $\mathbf{X}_T^{0,T} = x_T$ and $\mathbf{Y}_T^{0,T} = x_0$.

Motivated by this characterization, we define our variational family as a parametric subset of $\mathrm{SI}(x_0, x_T)$ of the form

$$\mathbb{Q}_{|0,T}^{\psi}(\cdot|x_0, x_T) = \mathrm{N}(I_t^{\psi}, (\gamma_t^{\psi})^2 \mathrm{I}_d) \ ,$$

where the mean path $I^{\psi} \in \mathcal{I}(x_0, x_T)$ and the standard deviation $\gamma^{\psi} \in \Gamma$ are parameterized by $\psi \in \Psi$. In this case, $\mathbb{Q}_{|0,T}^{\psi}(\cdot|x_0, x_T)$ is associated with forward SDE (G.12) and backward SDE (G.13) where

$$\begin{aligned}v_t^{f,\psi}(x|x_0, x_T) &= \partial_t I_t^{\psi} + \left(\frac{\partial_t \gamma_t^{\psi}}{\gamma_t^{\psi}} - \frac{\sigma^2}{2(\gamma_t^{\psi})^2}\right)(x - I_t) \ , \\ v_t^{b,\psi}(x|x_0, x_T) &= -\partial_t I_t^{\psi} + \left(-\frac{\partial_t \gamma_t^{\psi}}{\gamma_t^{\psi}} - \frac{\sigma^2}{2(\gamma_t^{\psi})^2}\right)(x - I_t) \ .\end{aligned}$$

Simplification in the deterministic regime. When $\sigma = 0$, the SDEs (G.12) and (G.13) turn into ODEs, whose simulation is completely tractable. Indeed, we have for any $s > t$, the deterministic rules

$$\begin{aligned}\mathbf{X}_s^{0,T} &= \frac{\gamma_s^{\psi}}{\gamma_t^{\psi}}(x_t - I_t^{\psi}) + I_s^{\psi} && \text{conditionally to } \mathbf{X}_t^{0,T} = x_t \ , \\ \mathbf{Y}_s^{0,T} &= \frac{\gamma_s^{\psi}}{\gamma_t^{\psi}}(y_t - I_t^{\psi}) + I_s^{\psi} && \text{conditionally to } \mathbf{Y}_t^{0,T} = y_t \ .\end{aligned}$$

Combination with spline parameterization. Following Liu et al. [2024], we may parameterize the mean path I^{ψ} as a d-dimensional linear spline and the standard deviation γ^{ψ} as a one-dimensional linear spline defined through control points. More precisely, we fix $C \geq 1$ intermediate control times $\{t_i\}_{i=1}^C$ in $(0,T)$ such that

$$0 < t_1 < t_2 < \ldots < t_C < T \ .$$

For a given conditioning pair (x_0, x_T), we define

$$I_t^{\psi} = \text{Linear Spline}(t; x_0, \{X_{t_i}\}_{i=1}^C, x_T), \qquad \gamma_t^{\psi} = \text{Linear Spline}(t; 0, \{\gamma_{t_i}\}_{i=1}^C, 0) \ ,$$

where the control points $\{X_{t_i}\}_{i=1}^C \subset \mathbb{R}^d$ and $\{\gamma_{t_i}\}_{i=1}^C \subset (0,\infty)$ are the parameters to optimize. In practice, the number of control points C can be chosen significantly smaller than the number of discretization steps used to simulate the SDEs. This yields a favorable trade-off between expressive power of the interpolant and computational efficiency.

Gaussian mixture variational family. Assume that we are provided with M Gaussian stochastic interpolants described by $\{I^{\psi_m}, \gamma^{\psi_m}\}_{m=1}^M$ with associated drift functions $\{v^{f,\psi_m}, v^{b,\psi_m}\}_{m=1}^M$. Following [Du et al., 2024, Proposition 4], we can build Gaussian mixture stochastic interpolant with the following setting

$$\mathbb{Q}^{\psi}_{t|0,T} = \sum_{m=1}^M w_m \mathbb{Q}^{\psi_m}_{t|0,T} ,$$

$$v_t^{f,\psi}(x|x_0,x_T) = \sum_{m=1}^M \tilde{w}_m(x|x_0,x_T) v_t^{f,\psi_m}(x|x_0,x_T) , \quad v_t^{b,\psi}(x|x_0,x_T) = \sum_{m=1}^M \tilde{w}_m(x|x_0,x_T) v_t^{b,\psi_m}(x|x_0,x_T) ,$$

$$\text{with } \tilde{w}_m(x|x_0,x_T) = \frac{w_m \mathbb{Q}^{\psi_m}_{t|0,T}(x|x_0,x_T)}{\sum_{j=1}^M w_j \mathbb{Q}^{\psi_j}_{t|0,T}(x|x_0,x_T)} ,$$

where $\{w_m\}_{m=1}^M$ are mixture weights, that can be parameterized as well.

G.1.3 Results on the Markovian projection

Proposition 136 (Extended version of Proposition 19)**.** *Let $\Pi \in \mathcal{R}(\mathbb{Q}^V)$. Then, $\mathrm{proj}_{\mathcal{M}_T}(\Pi)$ is equivalently induced by the diffusion processes $(\mathbf{X}_t)_{t\in[0,T]}$ and $(\mathbf{Y}_{T-t})_{t\in[0,T]}$, which are time-reversed of each other and respectively defined by the SDEs*

$$\mathrm{d}\mathbf{X}_t = v_t^{f,\star}(\mathbf{X}_t)\mathrm{d}t + \sigma \mathrm{d}\mathbf{B}_t , \quad \mathbf{X}_0 \sim \Pi_0 , \tag{G.14}$$

$$\mathrm{d}\mathbf{Y}_t = v_{T-t}^{b,\star}(\mathbf{Y}_t)\mathrm{d}t + \sigma \mathrm{d}\tilde{\mathbf{B}}_t , \quad \mathbf{Y}_0 \sim \Pi_T , \tag{G.15}$$

where

$$v_t^{f,\star}(x) = \mathbb{E}_{\Pi_{|t}}[\sigma^2 \nabla_{\mathbf{X}_t} \log \mathbb{Q}_{T|t}(\mathbf{X}_T|\mathbf{X}_t) - \textstyle\int_t^T \frac{T-s}{T-t} \nabla V_s(\mathbf{X}_s)\mathrm{d}s | \mathbf{X}_t = x] , \tag{G.16}$$

$$v_t^{b,\star}(x) = \mathbb{E}_{\Pi_{|t}}[\sigma^2 \nabla_{\mathbf{X}_t} \log \mathbb{Q}_{t|0}(\mathbf{X}_t|\mathbf{X}_0) - \textstyle\int_0^t \frac{s}{t} \nabla V_s(\mathbf{X}_s)\mathrm{d}s | \mathbf{X}_t = x] . \tag{G.17}$$

Moreover, $v^{f,\star}$ and $v^{b,\star}$ can be approximated by $v^{\theta_f^\star}$ and $v^{\theta_b^\star}$, where $\theta_f^\star$ and $\theta_b^\star$ are respectively the minimizers of forward objective $\mathcal{L}^f_{\mathcal{M}_T}$, given in (8.18), and backward objective $\mathcal{L}^b_{\mathcal{M}_T}$, given in (8.19).

Proof. The structure of the proof closely follows that of [Shi et al., 2023, Proposition 2] and relies on analogous technical assumptions, which we omit here for readability. We detail the argument in the forward setting; the backward case follows by entirely similar arguments.

Since $\Pi \in \mathcal{R}(\mathbb{Q}^V)$, we have $\Pi_{|0} = \phi_{T|0}\mathbb{Q}^V_{|0}$, where $\phi_{T|0} = \frac{\mathrm{d}\Pi_{T|0}}{\mathrm{d}\mathbb{Q}^V_{T|0}}$. In particular, for any almost surely $x_0 \sim \Pi_0$, $\Pi_{|0}(\cdot|x_0)$ is obtained from $\mathbb{Q}^V_{|0}(\cdot|x_0)$ through a Doob's h-transform, and is therefore associated with the SDE

$$\mathrm{d}\mathbf{X}_t = \mathbf{u}_t^V(\mathbf{X}_t)\mathrm{d}t + \sigma^2 \nabla \log \phi_{t|0}(\mathbf{X}_t|\mathbf{X}_0)\mathrm{d}t + \sigma \mathrm{d}\mathbf{B}_t , \ \mathbf{X}_0 = x_0 ,$$

where $\mathbf{u}^V$ denotes the drift of the SDE associated with $\mathbb{Q}^V$ (see (G.9)), and the potential $\phi_{t|0}$ is given by

$$\phi_{t|0}(x_t|x_0) = \mathbb{E}_{\mathbb{Q}^V_{T|t}}[\phi_{T|0}(\mathbf{X}_T|x_0) \mid \mathbf{X}_t = x_t] .$$

Observing that $\phi_{t|0}(x_t|x_0) = \frac{\mathrm{d}\Pi_{t|0}}{\mathrm{d}\mathbb{Q}^V_{t|0}}(x_t|x_0)$, we may apply Tweedie's formula to obtain

$$\nabla \log \phi_{t|0}(x_t|x_0) = \mathbb{E}_{\Pi_{T|0,t}}[\nabla_{\mathbf{X}_t} \log \mathbb{Q}^V_{T|t}(\mathbf{X}_T|\mathbf{X}_t) \mid \mathbf{X}_t = x_t, \mathbf{X}_0 = x_0] .$$

We now compute the score $\nabla \log \mathbb{Q}^V_{T|t}$. By definition,

$$\mathrm{d}\mathbb{Q}^V_{T|t}(x_T|x_t) = \frac{1}{Z_t(x_t)} \overbrace{\int \exp\left(-\int_t^T \frac{V_s(x_s)}{\sigma^2}\mathrm{d}s\right) \mathrm{d}\mathbb{Q}(x_{(t,T)}|x_t,x_T)}^{F(x_t,x_T)} \mathrm{d}\mathbb{Q}_{T|t}(x_T|x_t)$$

$$\text{where } Z_t(x_t) = \int \exp\left(-\int_t^T \frac{V_s(x'_s)}{\sigma^2}\mathrm{d}s\right) \mathrm{d}\mathbb{Q}(x'_{(t,T]}|x_t) .$$

Therefore,

$$\nabla_{x_t} \log \mathbb{Q}^V_{T|t}(x_T|x_t) = -\nabla \log Z_t(x_t) + \nabla_{x_t} \log \mathbb{Q}_{T|t}(x_T|x_t) + \nabla_{x_t} \log F(x_t, x_T) \ .$$

To compute $\nabla_{x_t} \log F(x_t, x_T)$, we use the Brownian bridge reparameterization inside $F(x_t, x_T)$. This yields

$$F(x_t, x_T) = \int_0^{T-t} \exp\left(- \int_t^T \frac{1}{\sigma^2} V_s \left(\frac{s-t}{T-t} x_T + \frac{T-s}{T-t} x_t + \sigma \{ \mathbf{B}_{s-t} - \frac{s-t}{T-t} \mathbf{B}_{T-t} \} \right) \right) \mathrm{d}\mathbf{B}_u \ .$$

Differentiating under the integral sign, we obtain

$$\nabla_{x_t} \log F(x_t, x_T) = \frac{1}{\sigma^2} \mathbb{E}_{\mathbb{Q}^V_{|t,T}} \left[- \int_t^T \frac{T-s}{T-t} \nabla V_s(\mathbf{X}_s) \mathrm{d}s \mid \mathbf{X}_t = x_t, \mathbf{X}_T = x_T \right] \ .$$

Using the same reparameterization argument for $Z_t(x_t)$, we further obtain

$$\nabla_{x_t} \log Z_t(x_t) = \frac{1}{\sigma^2} \mathbb{E}_{\mathbb{Q}^V_{|t}} \left[- \int_t^T \nabla V_s(\mathbf{X}_s) \mathrm{d}s \mid \mathbf{X}_t = x_t \right] = \frac{\mathbf{u}^V_t(x_t)}{\sigma^2} \ .$$

Combining the above identities, we conclude that $\Pi_{|0}(\cdot|x_0)$ is associated with the SDE

$$\mathrm{d}\mathbf{X}_t = \underbrace{\mathbb{E}_{\Pi_{|0,t}} \left[\sigma^2 \nabla_{x_t} \log \mathbb{Q}_{T|t}(\mathbf{X}_T|\mathbf{X}_t) - \int_t^T \frac{T-s}{T-t} \nabla V_s(\mathbf{X}_s) \mathrm{d}s \mid \mathbf{X}_t, \mathbf{X}_0 \right]}_{\tilde{v}^{f,\star}_t(\mathbf{X}_t, \mathbf{X}_0)} \mathrm{d}t + \sigma \mathrm{d}\mathbf{B}_t \ , \ \mathbf{X}_0 = x_0 \ .$$

Let $v \in \mathbf{A}_T$, and denote by $\mathbb{M}$ the path measure associated with

$$\mathrm{d}\mathbf{X}_t = v_t(\mathbf{X}_t)\mathrm{d}t + \sigma \mathrm{d}\mathbf{B}_t \ , \quad \mathbf{X}_0 \sim \Pi_0 \ .$$

Since $\mathbb{M} = \Pi_0$, we have by Girsanov's theorem,

$$\mathrm{KL}(\Pi \| \mathbb{M}) = \frac{1}{2\sigma^2} \int_0^T \mathbb{E}_{\Pi_{0,t}} \left[\left\| \tilde{v}^{f,\star}_t(\mathbf{X}_t, \mathbf{X}_0) - v_t(\mathbf{X}_t) \right\|^2 \right] \mathrm{d}t \ .$$

This expression is minimized when $v_t = v^{f,\star}_t$, proving that $\mathrm{proj}_{\mathcal{M}_T}(\Pi)$ is associated with (G.14).

To obtain a computable formulation of the Markovian projection step, we replace the original Markovian projection step by the variational problem

$$\arg\min \{ \mathrm{KL}(\mathrm{proj}_{\mathcal{M}_T}(\Pi) \| \mathbb{M}_\theta) \ : \ \theta \in \Theta \} \ ,$$

where $\mathbb{M}_\theta$ is associated with

$$\mathrm{d}\mathbf{X}_t = v^\theta_t(\mathbf{X}_t)\mathrm{d}t + \sigma \mathrm{d}\mathbf{B}_t \ , \quad \mathbf{X}_0 \sim \Pi_0 \ .$$

For any $\theta \in \Theta$, by Girsanov's theorem,

$$\mathrm{KL}(\mathrm{proj}_{\mathcal{M}_T}(\Pi) \| \mathbb{M}_\theta) = \frac{1}{2\sigma^2} \int_0^T \mathbb{E}_{\mathrm{proj}_{\mathcal{M}_T}(\Pi)_t} \left[\left\| v^{f,\star}_t(\mathbf{X}_t) - v^\theta_t(\mathbf{X}_t) \right\|^2 \right] \mathrm{d}t \tag{G.18}$$

$$= \frac{1}{2\sigma^2} \int_0^T \mathbb{E}_{\Pi_t} \left[\left\| v^{f,\star}_t(\mathbf{X}_t) - v^\theta_t(\mathbf{X}_t) \right\|^2 \right] \mathrm{d}t \tag{G.19}$$

$$= \frac{1}{2\sigma^2} \mathcal{L}^f_{\mathcal{M}_T}(\theta) + \mathrm{c} \ , \tag{G.20}$$

where c is independent of θ. This equivalence shows that minimizing $\mathcal{L}^f_{\mathcal{M}_T}(\theta)$ ensures that $v^{\theta^\star_f}_t$ recovers $v^{f,\star}_t$. □

Link with canonical setting. These expressions of $\mathrm{proj}_{\mathcal{M}_T}(\Pi)$ naturally derive from the Markovian structure of the twisted bridge $\mathbb{Q}^V_{|0,T}$ presented in Corollary 135. Indeed, using the fact that $\Pi \in \mathcal{R}(\mathbb{Q}^V)$, we observe that

$$v^{f,\star}_t(x) = \mathbb{E}_{\Pi_{T|t}} \left[\sigma^2 \nabla_{\mathbf{X}_t} \log \mathbb{Q}_{T|t}(\mathbf{X}_T|\mathbf{X}_t) + \mathbf{u}^{f,V}_t(\mathbf{X}_t|\mathbf{X}_0) | \mathbf{X}_t = x \right] \ , \qquad \text{derived from (G.10)} \ , \tag{G.21}$$

$$v^{b,\star}_t(x) = \mathbb{E}_{\Pi_{0|t}} \left[\sigma^2 \nabla_{\mathbf{X}_t} \log \mathbb{Q}_{t|0}(\mathbf{X}_t|\mathbf{X}_0) + \mathbf{u}^{b,V}_t(\mathbf{X}_t|\mathbf{X}_0) | \mathbf{X}_t = x \right] \ , \qquad \text{derived from (G.11)} \ . \tag{G.22}$$

In particular, taking the conditional expectation of the conditional velocity field amounts to exactly the same operation as in the canonical setting.

G.2 Discrete-time setting

G.2.1 Results on twisted measures

In this section, we present general results concerning the Markovian decomposition of the twisted bridge $q^g_{|0,N+1}$ relative to the discrete-time Brownian bridge $q_{|0,N+1}$. In particular, we derive closed-form expressions for the corresponding quantities. Although explicit, these expressions remain computationally intractable in practice.

Lemma 137. *The bridge $q^g_{1:N|0,N+1} \in \mathcal{M}_{N-2}$ admits the following forward Markov decomposition*

$$q^g_{1:N|0,N+1}(x_{1:N}|x_0, x_{N+1}) = \prod_{n=1}^{N} q^g_{n|n-1,N+1}(x_n|x_{n-1}, x_{N+1})$$

where for any $n \in \{1, \dots, N\}$, we have

$$q^g_{n|n-1,N+1}(x_n|x_{n-1}, x_{N+1}) = \frac{h_n(x_n|x_{N+1})g_n(x_n)}{h_{n-1}(x_{n-1}|x_{N+1})} q_{n|n-1,N+1}(x_n|x_{n-1}, x_{N+1})$$

with

$$h_n(x_n|x_{N+1}) = \int_{(\mathbb{R}^d)^{N-n}} \prod_{k=n+1}^{N} g_k(x_k) \mathrm{d}q_{k|k-1,N+1}(x_k|x_{k-1}, x_{N+1}) \, .$$

Lemma 138. *For any $n \in \{1, \dots, N\}$, the score of the forward bridge kernel $q^g_{n|n-1,N+1}$ is given by*

$$\begin{aligned}\nabla \log q^g_{n|n-1,N+1}(x_n|x_{n-1}, x_{N+1}) &= \nabla \log q_{n|n-1,N+1}(x_n|x_{n-1}, x_{N+1}) \\ &+ \sum_{k=n}^{N} \mathbb{E}_{q^g_{k|n,N+1}} \left[\frac{T - t_k}{T - t_n} \nabla \log g_k(X_k) | X_n = x_n, X_{N+1} = x_{N+1} \right]\end{aligned}$$

Proof. Since we have for any $n \in \{1, \dots, N\}$,

$$\nabla \log q^g_{n|n-1,N+1}(x_n|x_{n-1}, x_{N+1}) = \nabla \log q_{n|n-1,N+1}(x_n|x_{n-1}, x_{N+1}) + \nabla \log h_n(x_n|x_{N+1}) + \nabla \log g_n(x_n) \, ,$$

proving this result amounts to prove the identity

$$\nabla \log h_n(x_n|x_{N+1}) = \sum_{k=n+1}^{N} \mathbb{E}_{q^g_{k|n,N+1}} \left[\frac{T - t_k}{T - t_n} \nabla \log g_k(X_k) | X_n = x_n, X_{N+1} = x_{N+1} \right] \, . \tag{G.23}$$

We prove this equivalent result via backward recursion on n. First note that this result stands trivially for $n = N$, since $h(x_N|x_{N+1}) = 1$, and therefore, $\nabla \log h_N(x_N|x_{N+1}) = 0$, validating the result. Assume (G.23) is verified at rank $(n+1)$ for some $n \in \{1, \dots, N-1\}$. At rank n, we obtain by Lemma 137 the following recursion relation

$$h_n(x_n|x_{N+1}) = \int h_{n+1}(x_{n+1}|x_{N+1}) g_{n+1}(x_{n+1}) \mathrm{d}q_{n+1|n,N+1}(x_{n+1}|x_n, x_{N+1}) \, .$$

Hence, using the reparameterization trick in the discrete Brownian bridge $q_{n+1|n,N+1}$ along with Tweedie formula, we obtain

$$\begin{aligned}\nabla \log h_n(x_n|x_{N+1}) &= \frac{T - t_{n+1}}{T - t_n} \mathbb{E}_{q^g_{n+1|n,N}} \left[\nabla \log h_{n+1}(X_{n+1}|x_{N+1}) + \nabla \log g_{n+1}(X_{n+1}) \middle| X_n = x_n, X_{N+1} = x_{N+1} \right] \, , \\ &= \frac{T - t_{n+1}}{T - t_n} \sum_{k=n+2}^{N} \mathbb{E}_{q^g_{k|n,N+1}} \left[\frac{T - t_k}{T - t_{n+1}} \nabla \log g_k(X_k) | X_n = x_n, X_{N+1} = x_{N+1} \right] \\ &+ \mathbb{E}_{q^g_{n+1|n,N+1}} \left[\frac{T - t_{n+1}}{T - t_n} \nabla \log g_{n+1}(X_{n+1}) | X_n = x_n, X_{N+1} = x_{N+1} \right] \, ,\end{aligned}$$

where the simplification comes from the recursion assumption from rank $(n+1)$. Hence, we obtain the result at rank n, which proves the result for any $n \in \{1, \dots, N\}$. □

Lemma 139. *Symmetrically, $q^g_{1:N|0,N+1} \in \mathcal{M}_{N-2}$ admits the following backward Markov decomposition*

$$q^g_{1:n|0,N+1}(x_{1:N}|x_0, x_{N+1}) = \prod_{n=2}^{N+1} q^g_{n-1|0,n}(x_{n-1}|x_0, x_n)$$

where for any $n \in \{2, \dots, N+1\}$

$$q^g_{n-1|0,n}(x_{n-1}|x_0, x_n) = \frac{h_{n-1}(x_{n-1}|x_0) g_{n-1}(x_{n-1})}{h_n(x_n|x_0)} q_{n-1|0,n}(x_{n-1}|x_0, x_n) \,,$$

where

$$h_n(x_n|x_0) = \int_{(\mathbb{R}^d)^{n-1}} \prod_{k=2}^{n} g_{k-1}(x_{k-1}) \mathrm{d}q_{k-1|0,k}(x_{k-1}|x_0, x_k) \,.$$

Lemma 140. *For any $n \in \{2, \dots, N+1\}$, the score of the forward bridge kernel $q^g_{n-1|0,n}$ is given by*

$$\nabla \log q^g_{n-1|0,n}(x_{n-1}|x_0, x_n) = \nabla \log q_{n-1|0,n}(x_{n-1}|x_0, x_n) \tag{G.24}$$

$$+ \sum_{k=1}^{n-1} \mathbb{E}_{q^g_{k|0,n-1}} \left[\frac{t_k}{t_{n-1}} \nabla \log g_k(X_k) | X_0 = x_0, X_{n-1} = x_{n-1} \right] \tag{G.25}$$

Proof. The proof structure follows the same recursion principle as for Lemma 138, but with increasing n, and exploiting the formula from Lemma 139. □

G.2.2 Results on the Twisted SB problem

Proposition 141 (Formal version of Proposition 20)**.** *Let $V \in \mathrm{C}^{0,1}([0,T] \times \mathbb{R}^d, \mathbb{R})$ be such that $Z_V < \infty$. Define the potentials $g_n = \exp(-\delta_n V_{t_n}/\sigma^2)$ for any $n \in \{1, \dots, N\}$. Denote by $\mathbb{P}^N$ the solution to the continuous-time SB problem (8.1) with reference measure given by*

$$\mathrm{d}\mathbb{Q}^N(x_{[0:T]}) = \frac{1}{Z_N} \prod_{n=1}^{N} g_n(x_{t_n}) \mathrm{d}\mathbb{Q}(x_{[0:T]}) \,, \quad \text{where} \quad Z_N = \int \prod_{n=1}^{N} g_n(x_{t_n}) \mathrm{d}\mathbb{Q}(x_{[0:T]}) \,.$$

Assume that $\mathbb{P}^N$ converges to a certain $\mathbb{P}^\infty \in \mathscr{P}(\mathbf{C}_T)$ as $N \to \infty$ in the weak topology. Then, $\mathbb{P}^\infty$ is solution to (8.1) with reference measure $\mathbb{Q}^V$.

Proof. Denote by $\mathbb{P}^\star$ the solution to (8.1) with reference measure $\mathbb{Q}^V$. First note that for any $\mathbb{P} \in \mathscr{P}(\mathbf{C}_T)$,

$$\mathbb{E}_{\mathbb{P}}[\log \frac{\mathrm{d}\mathbb{Q}^V}{\mathrm{d}\mathbb{Q}^N}] = \mathbb{E}_{\mathbb{P}} \left[\exp \left(\sum_{n=0}^{N} \int_{t_n}^{t_{n+1}} \frac{V_{t_{n+1}}(\mathbf{X}_{t_{n+1}}) - V_t(\mathbf{X}_t)}{\sigma^2} \mathrm{d}t \right) \right] + \log \frac{Z_N}{Z_V} \to 0 \tag{G.26}$$

using the dominated convergence theorem. Then, we have

$$\begin{aligned}
\mathrm{KL}(\mathbb{P}^\infty \| \mathbb{Q}^V) &\leq \liminf \mathrm{KL}(\mathbb{P}^N \| \mathbb{Q}^N) && \text{(KL property)} \\
&\leq \liminf \mathrm{KL}(\mathbb{P}^\star \| \mathbb{Q}^N) && \text{(definition of } \mathbb{Q}^N\text{)} \\
&= \mathrm{KL}(\mathbb{P}^\star \| \mathbb{Q}^V) + \liminf \mathbb{E}_{\mathbb{P}^\star}[\log \frac{\mathrm{d}\mathbb{Q}^V}{\mathbb{Q}^N}] \\
&= \mathrm{KL}(\mathbb{P}^\star \| \mathbb{Q}^V) \,.
\end{aligned}$$

Then, we conclude by uniqueness of $\mathbb{P}^\star$ that $\mathbb{P}^\infty = \mathbb{P}^\star$. □

G.2.3 Results on the Markovian projection

Proofs of Lemma 22 and Lemma 23. These results follow from the observation that any $p \in \mathcal{R}(q^g)$ writes

$$\begin{aligned} p_{0:N+1} &= p_{0,N+1} q^g_{1:N|0,N+1} \\ &= p_0 p_{N+1|0} \prod_{n=1}^{N} q^g_{n|n-1,N+1} && \text{(forward decomposition)} \\ &= p_{N+1} p_{0|N+1} \prod_{n=2}^{N+1} q^g_{n-1|0,n} && \text{(backward decomposition)} , \end{aligned}$$

using the Markov structure of the twisted bridge, see Lemmas 137 and 139. Then, we obtain the KL decompositions by applying the KL chain rule. □

Proof of Proposition 24. In this proof, we solely consider the forward setting, since the backward result can be proved with very similar tools. Let $\theta_f \in \Theta$, and denote by $m \in \mathcal{M}_N$ the Markovian measure obtained via forward denoiser-based parameterization (8.32) with θ^f. In this case, we have for any $n \in \{1, \dots, N\}$, any (x_{n-1}, x_{N+1}) $p_{n-1,N+1}$-almost surely, up to additive constants independent from θ^f,

$$\begin{aligned} &\mathrm{KL}(q^g_{n|n-1,N+1}(\cdot|x_{n-1}, x_{N+1}) \| m_{n|n-1}(\cdot|x_{n-1})) \\ &\quad \overset{c}{=} \frac{1}{2\sigma^2(\gamma^f_n)^2} \mathbb{E}\left[\left\| X_n - \alpha^f_n x_{n-1} - \beta^f_n \tilde{x}_{N+1} \right\|^2 \right] \\ &\quad \overset{c}{=} \frac{1}{2} \mathbb{E}\left[\left\| -\sigma\gamma^f_n \nabla \log q_{n|n-1,N+1}(X_n|x_{n-1}, x_{N+1}) + \frac{\beta^f_n}{\sigma\gamma^f_n}(x_{N+1} - \tilde{x}_{N+1}) \right\|^2 \right] \end{aligned}$$

where $\tilde{x}_{N+1} = \tilde{x}^{\theta_f}_{N+1}(t_{n-1}, x_{n-1})$, and the expectation is taken with respect to $q^g_{n|n-1,N+1}(\cdot|x_{n-1}, x_{N+1})$. Using the result from Lemma 138, combined with the fact that

$$\mathbb{E}_{q^g_{n|n-1,N+1}}[\nabla \log q^g_{n|n-1,N+1}(X_n|x_{n-1}, x_{N+1})] = 0 ,$$

we obtain, up to additive constants independent from θ^f,

$$\begin{aligned} &\mathrm{KL}(q^g_{n|n-1,N+1}(\cdot|x_{n-1}, x_{N+1}) \| m_{n|n-1}(\cdot|x_{n-1})) \\ &\quad \overset{c}{=} \frac{1}{2} \mathbb{E}\left[\left\| \sigma\gamma^f_n \sum_{k=n}^{N} \mathbb{E}\left[\frac{T - t_k}{T - t_n} \nabla \log g_k(X_k) | X_n = x_n, X_{N+1} = x_{N+1} \right] + \frac{\beta^f_n}{\sigma\gamma^f_n}(x_{N+1} - \tilde{x}_{N+1}) \right\|^2 \right] \\ &\quad \overset{c}{=} \frac{1}{2(N-n+1)\sigma^2} \sum_{k=n}^{N} \mathbb{E}\left[\left\| \sigma^2 \gamma^f_n (N-n+1) \frac{T - t_k}{T - t_n} \nabla \log g_k(X_k) + \frac{\beta^f_n}{\gamma^f_n}(x_{N+1} - \tilde{x}_{N+1}) \right\|^2 \right] . \end{aligned}$$

By deriving this identity for all $n \in \{1, \dots, N\}$, and combining the results, we conclude that

$$\mathrm{KL}(p|m) = \frac{1}{2\sigma^2} \mathcal{L}^f_{\mathcal{M}_N}(\theta_f) + \mathrm{C} ,$$

where C is a constant independent from θ_f. This equivalence shows that minimizing $\mathcal{L}^f_{\mathcal{M}_N}(\theta_f)$ ensures that $\theta^\star_f$ yields the forward Markovian projection of p within the Markov measures with denoiser-like parameterization. □

Appendix H

Fast, faithful and photorealistic diffusion-based image super-resolution with enhanced Flow Map models

Figure H.1: **Qualitative super-resolution results of the *Shortcut* variant of FlowMapSR for $\times 4$ and $\times 8$ upscaling.** A single Flow Map model is used for both upscaling factors, with 2 inference steps.

Diffusion-based image super-resolution (SR) has recently attracted significant attention by leveraging the expressive power of large pretrained text-to-image diffusion models. A central practical challenge in this setting is resolving the trade-off between reconstruction faithfulness and photorealism. To address inference efficiency, many recent works have explored knowledge distillation strategies specifically tailored to SR, enabling one-step diffusion-based approaches. However, these teacher–student formulations are inherently constrained by information compression, which can degrade perceptual cues such as lifelike textures and depth of field, despite strong perceptual quality. In parallel, self-distillation diffusion models, also known as Flow Map models, have emerged as a promising alternative for image generation tasks, enabling fast inference while preserving the expressivity and training stability of standard diffusion models. Building on these developments, we propose FlowMapSR, a novel diffusion-based framework for image super-resolution explicitly designed for efficient inference. Beyond adapting Flow Map models to the SR task, we introduce two complementary enhancements: (i) positive–negative prompting guidance, based on a generalization of classifer free-guidance paradigm to Flow Map models, and (ii) adversarial fine-tuning using Low-Rank Adaptation (LoRA). Among the considered Flow Map formulations (*Eulerian*, *Lagrangian*, and *Shortcut*), we find that the *Shortcut* variant consistently achieves the best performance when combined with these enhancements. Extensive experiments show that FlowMapSR achieves a better balance between reconstruction faithfulness and photorealism than recent state-of-the-art methods for both $\times 4$ and $\times 8$ upscaling, while maintaining competitive inference time. Notably, a single model is used for both upscaling factors, without any scale-specific conditioning or degradation-guided mechanisms.

H.1 Introduction

Image super-resolution (SR) is a fundamental problem in computer vision, which aims to reconstruct a plausible high-resolution (HR) image from a low-resolution (LR) input. The task is inherently ill-posed, as SR exhibits a strong *one-to-many* ambiguity: the LR observation provides only partial structural information, leaving high-frequency details fundamentally underdetermined. Consequently, SR methods must navigate a delicate trade-off between reconstruction *faithfulness*—recovering details consistent with the underlying HR image—and *photorealism*—producing visually plausible textures for the human eye. This trade-off becomes increasingly pronounced as the upscaling factor (commonly $\times 4$ or $\times 8$) increases, since the LR input becomes less informative and the space of plausible solutions expands, and is further complicated by unknown degradations such as blur, sensor noise, and compression artefacts.

Owing to these challenges, SR has long served as a driving force for advances in generative modeling. While early learning-based approaches were dominated by GAN-based methods [Wang et al., 2018, 2021] and Transformer architectures [Lu et al., 2022], recent research has increasingly focused on diffusion models (DMs) [Sohl-Dickstein et al., 2015, Ho et al., 2020, Song et al., 2021b] and related flow-matching (FM) formulations [Lipman et al., 2023], which have demonstrated a strong capacity to synthesize fine details and realistic textures in image generation tasks. As a result, diffusion-based approaches now define the state-of-the-art for image SR. Existing diffusion-based SR approaches can be broadly divided into two categories. The first consists of inference-only methods, which leverage a pre-trained DM and apply advanced posterior sampling techniques to solve the SR inverse problem without need of ground truth LR-HR data pairs or additional training; see Janati et al. [2025] for a comprehensive overview. Although these methods achieve consistently strong results across a variety of SR scenarios, they typically require dozens of model evaluations per image, leading to prohibitive computational costs for large-scale applications. The second category specifically relies on LR-HR training data pairs (the LR inputs being synthetically obtained from available HR images) and focuses on training DMs explicitly for SR, either from scratch [Li et al., 2022, Saharia et al., 2022, Yue et al., 2023, 2024, Luo et al., 2023, Delbracio and Milanfar, 2023] or by fine-tuning large pre-trained backbones [Lin et al., 2024, Wang et al., 2024a, Wu et al., 2024b, Yang et al., 2024]. Although these approaches deliver strong performance, they also rely on tens to hundreds of sampling steps at inference time.

Beyond the specific context of SR, reducing the number of inference steps has become an active research direction for DMs more broadly. Prominent strategies include the development of efficient samplers [Song et al., 2021a, Lu et al., 2025], the design of straight or near-optimal transport paths [Liu et al., 2023, Esser et al., 2024, Wang et al., 2025a, Kim et al., 2025a], and various knowledge distillation techniques [Salimans and Ho, 2022, Sauer et al., 2024b,a, Yin et al., 2024, Chadebec et al., 2025a]. While these approaches have proven highly effective for unconditional or text-to-image generation, they do not directly translate to SR, where generation is strongly conditioned on the input image whose informativeness depends on both the degradation process and the upscaling factor. To address this challenge, several one-step diffusion-based SR methods (i.e., requiring only a single inference step) have recently emerged, primarily relying on teacher–student distillation. These approaches notably include adversarial distillation [Zhang et al., 2024a, Wu et al., 2025b, Tai et al., 2026, He et al., 2024a], score distillation [Wu et al., 2024a], and consistency techniques [Wang et al., 2024b], often combined with perceptual regularization losses. Despite their efficiency and strong empirical performance, those methods are fundamentally constrained by information loss between the teacher and student models, which may legitimately degrade photorealism, restrict texture diversity, and reduce robustness across different upscaling factors.

Recently, a new paradigm for fast diffusion inference, referred to as Flow Map models, has been introduced [Boffi et al., 2025a,b], with concrete instantiations proposed in the literature [Geng et al., 2025, Frans et al., 2025, Sabour et al., 2025]. These models adopt a *self-distillation* paradigm: starting from a FM formulation, they learn a dynamical mapping between a source and a target distribution while enforcing self-consistency across different discretization scales through an additional training objective. This formulation enables stable training in high-dimensional spaces and supports inference with very few (potentially a single) model evaluations. Flow Map models have recently demonstrated state-of-the-art performance for image generation at resolution 512×512 [Sabour et al., 2025], and ongoing work has begun exploring their scalability to higher resolution [Zheng et al., 2025, Hu et al., 2025] as well as their extension to training-free general reward alignment tasks (including SR) [Huang et al., 2026, Potaptchik et al., 2026]. Crucially, unlike distillation approaches that compress an iterative denoising process into the student network—thereby potentially incurring information loss—Flow Map models preserve the expressivity of large generative models through self-consistency, making them particularly appealing for the SR task.

In this work, we investigate the use of Flow Map models for image SR. Our objective is to combine the expressive power of large DMs with fast inference, while maintaining both faithfulness to the underlying HR content and strong perceptual quality. Our contributions are threefold:

- We introduce **FlowMapSR**, a simple and generic diffusion-based framework for SR built upon the three major Flow Map formulations (*Eulerian*, *Lagrangian*, *Shortcut*). To adapt Flow Map models to the specific challenges of SR, we propose a base implementation that guarantees perceptual quality along with two enhancements: positive–negative prompting guidance inspired by Zhang et al. [2024a], and lightweight adversarial fine-tuning using LoRA modules [Hu et al., 2022].
- We empirically demonstrate that the FlowMapSR instance based on the *Shortcut* formulation [Frans et al., 2025] (equivalently the *Progressive* approach from Boffi et al. [2025a]) provides the best performance and scalability in high-resolution SR scenarios. With inference time comparable to existing one-step state-of-the-art SR methods, this version of FlowMapSR produces reconstructions that are in general more faithful to the HR reference while exhibiting improved photorealistic characteristics, including realistic textures, depth of field, and high-frequency details. We validate these results on both synthetic and real-world benchmarks, and demonstrate consistent performance across $\times 4$ and $\times 8$ upscaling scenarios using a single model, without additional degradation-specific guidance or conditioning.
- Finally, we present a comprehensive ablation study of the proposed framework. Our analysis highlights a clear trade-off between reconstruction faithfulness and perceptual realism, providing insights into how FlowMapSR can be tuned to different application requirements.

H.2 Related works

Diffusion models have emerged as the dominant paradigm for image SR, thanks to their strong generative capabilities and their ability to overcome limitations of earlier GAN-based approaches Wang et al. [2018, 2021], that were prone to training instability and lack of photorealism. Early diffusion-based SR methods such as SR3 [Saharia et al., 2022] and SRDiff [Li et al., 2022] adapted the conditional DDPM framework [Ho et al., 2020] by explicitly conditioning the generation process on the LR input in pixel space. IDM [Gao et al., 2023] further extends this line of work by incorporating an implicit neural representation, enabling SR across arbitrary continuous upscaling factors. Thereafter, several methods extended diffusion-based SR to latent space by leveraging large pre-trained text-to-image backbones, such as Stable Diffusion [Rombach et al., 2022]. Notable examples include DiffBIR [Lin et al., 2024], StableSR [Wang et al., 2024a], and SeeSR [Wu et al., 2024b], which exploit rich semantic and perceptual representations learned at scale to improve visual quality. Despite their scalability and improved semantic modeling, these approaches still rely on the standard iterative diffusion inference process, and consequently require a large number of inference steps—typically more than 50. To attenuate this issue, some methods such as ResShift [Yue et al., 2023, 2024], IR-SDE [Luo et al., 2023], and InDI [Delbracio and Milanfar, 2023] accelerated inference by designing shorter Markov chains or optimizing transition schedules, cutting the number of inference steps from hundreds to a dozen with minimal loss in reconstruction quality.

A complementary and increasingly active research direction investigates one-step (or few-step) diffusion-based formulations via distillation techniques. In this setting, a large pre-trained text-to-image DM acts as a teacher and guides the training of a lightweight student network, enabling SR with a drastically reduced number of model evaluations. Existing methods primarily differ in the choice of distillation objective and the regularization mechanisms used to preserve fidelity and perceptual quality. We may distinguish: **(a) consistency-based distillation**, in which the student is trained to match the teacher's outputs through direct regression or bidirectional consistency constraints, see for instance SinSR [Wang et al., 2024b]; **(b) score distillation**, which aims to minimize a predefined divergence between the teacher and student output distributions. OSEDiff [Wu et al., 2024a] follows this approach and incorporates the LPIPS perceptual loss [Zhang et al., 2018] to encourage the recovery of high-frequency details; **(c) adversarial distillation**, which explicitly aligns the student's output distribution with that of the teacher using an auxiliary discriminator. For example, ADDSR [Tai et al., 2026] introduces a timestep-adaptive adversarial loss that dynamically balances teacher–student alignment across diffusion steps, leading to improved perceptual realism. Similarly, TAD-SR [He et al., 2024a] combines a time-aware adversarial loss with high-frequency-enhanced score distillation to better recover fine textures. Along the same line, S3Diff [Zhang et al., 2024a] enhances adversarial distillation by integrating LPIPS regularization, positive–negative prompting guidance, and LoRA-based degradation-aware conditioning, significantly improving robustness to real-world degradations. Finally, VPD-SR [Wu et al., 2025b] augments adversarial distillation with semantic-aware supervision and high-frequency perceptual losses, leveraging pre-trained vision–language representations to further improve semantic alignment and texture fidelity.

H.3 Preliminaries on Flow Map models

Consider two probability distributions π_0 (target) and π_1 (source) supported on $\mathbb{R}^d$, for some $d \geq 1$. We assume access to a training dataset $\mathcal{D} = \{(x_0^i, x_1^i)\}_{i=1}^N$, consisting of N paired samples drawn from a joint coupling of π_0 and π_1, denoted by $\pi_{0,1}$. In this section, we study the problem of generating new samples from π_0 based on *unseen* samples from π_1.

A natural approach to this conditional generative modeling problem is to learn a transport map $\mathrm{T} : \mathbb{R}^d \to \mathbb{R}^d$ that pushes π_1 to π_0 (such that $\mathrm{T}(x_1) \sim \pi_0$ whenever $x_1 \sim \pi_1$) by leveraging the available samples from $\mathcal{D}$. In practice, this is commonly addressed within the Flow Matching (FM) [Lipman et al., 2023] setting, which we briefly recall in Appendix H.3.1. Basically, FM models learn the drift of an Ordinary Differential Equation (ODE) whose solution transports π_1 to π_0, using a neural network. The resulting dynamics (neural ODE) defines a valid candidate for T, by composing a sequence of local transport maps obtained by numerically integrating the neural ODE over small time intervals.

While this approach is conceptually simple and has proven effective across a wide range of data modalities, it comes with a significant computational cost at inference time. In particular, FM inference generally requires dozens of neural network evaluations to control time discretization error, which can make FM-based methods impractical in large-scale settings. Flow Map models [Boffi et al., 2025a,b] extend the FM framework by explicitly addressing this limitation. Instead of relying on ODE numerical integration, they introduce an additional self-distillation objective which aims to learn *exact* integrations of the FM-based neural ODE on larger time intervals, thus enabling efficient inference with substantially fewer evaluations. In Appendix H.3.2, we review the main components of this recent approach. We then introduce in Appendix H.3.3 additional elements on conditioned Flow Map models that will serve as the foundation of our FlowMapSR framework.

H.3.1 Background on Flow Matching

Interpolant & ODE. FM can be seen as a particular instantiation of the Stochastic Interpolant (SI) framework [Albergo et al., 2023], whose paradigm is the following : (i) defining a continuous-time path of probability distributions $(p_t)_{t\in[0,1]}$ that interpolates between the boundary conditions $p_0 = \pi_0$ and $p_1 = \pi_1$, and (ii) constructing a marginal-preserving ODE by identifying a velocity field $v : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$ such that the stochastic process $(X_t)_{t\in[0,1]}$ governed by

$$\mathrm{d}X_t = v_t(X_t)\mathrm{d}t \ , \quad X_1 \sim \pi_1 \ , \tag{H.1}$$

verifies $X_t \sim p_t$ for any $t \in [0,1]$. Then, one may generate samples from π_0 by simulating ODE (H.1) backward in time, starting from samples drawn from π_1. In practice, the bridging distributions $(p_t)_{t\in[0,1]}$ are defined as the marginals of a *linear* interpolant $(\mathrm{I}_t)_{t\in[0,1]}$ of the form $\mathrm{I}_t = \alpha_t X_0 + \beta_t X_1$, where $(X_0, X_1) \sim \pi_{0,1}$ and $\alpha, \beta : [0,1] \to [0,1]$ are continuously differentiable functions satisfying boundary conditions $\alpha_0 = \beta_1 = 1$ and $\alpha_1 = \beta_0 = 0$. The standard FM setting is recovered by taking $\alpha_t = 1 - t$ and $\beta_t = t$. *Throughout the main paper, we restrict our attention to this standard formulation*, and defer the general case to Appendix H.7.1. In this setting, a valid—though generally intractable—expression for the velocity field is $v_t(x) = \mathbb{E}[v_{\mathrm{FM}}^{\mathrm{target}} | \mathrm{I}_t = x]$, where $v_{\mathrm{FM}}^{\mathrm{target}} = X_1 - X_0$ corresponds to the conditional target velocity.

Training & Inference. If v_t were known, a valid transport map pushing π_1 to π_0 would be given for any $x \sim \pi_1$ by

$$\mathrm{T}_{\mathrm{FM}}(x) = x - \int_0^1 v_r(X_r^{1,x})\mathrm{d}r \ , \tag{H.2}$$

where $(X_r^{t,x})_{r\in[0,t]}$ denotes for any $t \in [0,1]$ the solution to ODE (H.1) on $[0,t]$, with terminal condition $X_t = x$. In practice, v_t has no closed form but it can still be efficiently learned by learning a parametric model v_t^θ (e.g., a neural network) via the mean-square regression problem

$$\arg\min_\theta \mathbb{E}[\mathcal{L}_{\mathrm{FM}}(v_t^\theta, \mathrm{I}_t)] \ , \ \mathcal{L}_{\mathrm{FM}}(v_t^\theta, \mathrm{I}_t) = \|v_t^\theta(\mathrm{I}_t) - v_{\mathrm{FM}}^{\mathrm{target}}\|^2 \ , \tag{H.3}$$

where the expectation is taken over $(X_0, X_1) \sim \pi_{0,1}$ (approximated by samples from $\mathcal{D}$) and $t \sim q(t)$, q being a predefined time distribution on $[0,1]$. At inference time, replacing v_t with its learned approximation v_t^θ yields a neural ODE whose associated transport map remains intractable and must itself be approximated numerically. The standard approach relies on discretizing the integral in (H.2) on small time intervals (backward in time)

with numerical integration methods, starting from $t = 1$. Whether using the first-order Euler scheme or high-order ODE solvers such as Heun or Runge–Kutta methods, this inference procedure induces a large number of neural network evaluations, which can imply a prohibitive cost for large scale applications. Finally, the FM interpolation framework allows one to derive an *approximate* x_0-prediction given a state (x_t, t) of the ODE trajectory, if we assume that the target velocity field is straight and constant :

$$\hat{x}_0(x_t) = x_t - t\, v_t^\theta(x_t)\ . \tag{H.4}$$

While this approximation is accurate for small time values (where it coincides with a first-order Euler expansion around $t = 0$), it becomes generally loose at larger times for arbitrary couplings $\pi_{0,1}$, as it fails to capture the curvature of the neural ODE trajectory. However, in the special case where $\pi_{0,1}$ is the optimal transport plan associated with the quadratic cost between π_0 and π_1, the x_0-prediction (H.4) holds exactly for all $t \in [0, 1]$ [Pooladian et al., 2023, Tong et al., 2023]. This observation has motivated a substantial body of work on few-step (and theoretically single-step) FM models [Liu, 2022, Lee et al., 2024, Kim et al., 2025a, Kornilov et al., 2024], which provide an alternative to Flow Map approaches. A detailed study of these methods is beyond the scope of this paper.

FM enhancing via CFG. When data samples are associated with an additional class information $\mathbf{c}$ (e.g., a text label), the natural extension of FM consists in learning a class-conditional velocity field $v_t(x|\mathbf{c}) = \mathbb{E}[v_{\mathrm{FM}}^{\mathrm{target}}|\mathrm{I}_t = x, \mathbf{c}]$, which replaces v_t in ODE (H.1). In image generation, however, it has been empirically observed that combining $v_t(x|\mathbf{c})$ with its unconditional counterpart $v_t(x|\emptyset)$ (that is learned in standard FM) significantly improves improves sample quality.

This popular trick, known as *Classifier-Free Guidance* (CFG) [Ho and Salimans, 2022], amounts to substitute $v_t(x|\mathbf{c})$ at inference by the following linear combination

$$v_t^{\mathrm{cfg}}(x|\mathbf{c}) = w\, v_t(x|\mathbf{c}) + (1 - w)v_t(x|\emptyset)\ , \tag{H.5}$$

where $w \in (1, \infty)$ denotes the guidance strength, recovering the canonical setting in the limit $w \to 1$. In practice, increasing w enhances perceptual quality but often reduces faithfulness to realistic content. The conditional velocity $v_t(x|\mathbf{c})$ is learned via a class-conditional neural network $v_t^\theta(x|\mathbf{c})$ by minimizing a conditional variant of (H.3),

$$\arg\min_\theta \mathbb{E}[\mathcal{L}_{\mathrm{cFM}}(v_t^\theta, \mathrm{I}_t, \mathbf{c})]\ ,\ \mathcal{L}_{\mathrm{cFM}}(v_t^\theta, \mathrm{I}_t, \mathbf{c}) = \|v_t^\theta(\mathrm{I}_t|\mathbf{c}) - v_{\mathrm{FM}}^{\mathrm{target}}\|^2\ , \tag{H.6}$$

where the expectation is taken over $(X_0, X_1, \mathbf{c}) \sim \pi_{0,1}$ and $t \sim q(t)$. To expose v^θ to class-unconditional samples, a common practice is to drop the class condition $\mathbf{c}$ in (H.6) with a prescribed probability during training.

H.3.2 Background on Flow Map models

Definition & parameterization. Closely related to consistency-based DMs [Kim et al., 2024, Zheng et al., 2024, Heek et al., 2024, Li and He, 2024, Kim et al., 2025b], Flow Map models [Boffi et al., 2025b,a] and related approaches [Geng et al., 2025, Sabour et al., 2025, Frans et al., 2025] extend the FM framework to alleviate its high inference cost. Their key idea is to directly approximate the *integrated* velocity field over large time intervals, thereby enabling efficient inference with only a few steps. Formally, these methods aim to estimate the so-called *Flow Map* $X : [0, 1]^2 \times \mathbb{R}^d \to \mathbb{R}^d$ associated with the ODE (H.1). For any pair of timesteps $(s, t) \in [0, 1]^2$ such that $s \leq t$ and any point $x \in \mathbb{R}^d$, the Flow Map is defined as

$$X_{s,t}(x) = x - \int_s^t v_r(X_r^{t,x})\mathrm{d}r\ . \tag{H.7}$$

In particular, the transport map $x \mapsto X_{0,1}(x) = \mathrm{T}_{\mathrm{FM}}(x)$ can be obtained with a single evaluation of the Flow Map, making this formulation especially well suited for one-step inference. Following prior work [Boffi et al., 2025a, Frans et al., 2025], we consider in the rest of the paper an Euler-like parameterization of the Flow Map given by

$$X_{s,t}(x) = x - (t - s)u_{s,t}(x)\ ,\ u_{s,t}(x) = \frac{1}{t - s}\int_s^t v_r(X_r^{t,x})\mathrm{d}r\ , \tag{H.8}$$

where $u_{s,t}$ denotes the *average* velocity field on time interval $[s, t]$ associated to ODE (H.1) conditioned on terminal point $X_t = x$, that we now aim to estimate with a neural network $u_{s,t}^\theta$. Note that the instantaneous velocity field $v_t(x)$ corresponds to a zeroth-order approximation of $u_{s,t}(x)$ in the limit regime $s \to t$, i.e., $u_{t,t}(x) = v_t(x)$, which establishes a straightforward bridge between FM and Flow Map frameworks. Since we already know how to learn v_t, see (H.3), the remaining challenge is to estimate $u_{s,t}$ for $s < t$.

Self-distillation losses. To this end, prior works introduce three consistency-based characterizations of the Flow Map, leading to so-called *Self-Distillation* (SD) objective functions. Given timesteps $s < t$ along with the FM interpolant I_t, the SD losses take the general form

$$\mathcal{L}_{\mathrm{SD}}(u^\theta_{s,t}, \mathrm{I}_t) = \|u^\theta_{s,t}(\mathrm{I}_t) - \mathrm{sg}(u^{\mathrm{target}}_{\mathrm{SD}})\|^2 \,, \tag{H.9}$$

where sg is the stopgradient operator, and $u^{\mathrm{target}}_{\mathrm{SD}}$ depends on the chosen characterization:

1. *Lagrangian* SD (LSD) loss [Boffi et al., 2025a, Eq. (13)]
$$u^{\mathrm{target}}_{\mathrm{LSD}} = v^{\mathrm{target}}_{\mathrm{FM}} + (t-s)\partial_s u^\theta_{s,t}(\mathrm{I}_t)$$
2. *Eulerian* SD (ESD) loss [Geng et al., 2025, Eq. (9)], or equivalently [Boffi et al., 2025a, Eq. (14)]
$$u^{\mathrm{target}}_{\mathrm{ESD}} = v^{\mathrm{target}}_{\mathrm{FM}} - (t-s)\left(\nabla u^\theta_{s,t}(\mathrm{I}_t) v^{\mathrm{target}}_{\mathrm{FM}} + \partial_t u^\theta_{s,t}(\mathrm{I}_t)\right)$$
3. *Shortcut* SD (SSD) loss [Frans et al., 2025, Eq. (5)], or equivalently [Boffi et al., 2025a, Eq. (15)]
$$u^{\mathrm{target}}_{\mathrm{SSD}} = \frac{1}{2}u^\theta_{r,t}(\mathrm{I}_t) + \frac{1}{2}u^\theta_{s,r}\left(\mathrm{I}_t - \frac{t-s}{2}u^\theta_{r,t}(\mathrm{I}_t)\right), \quad r = \frac{s+t}{2}$$

For completeness, we provide detailed derivations of these objectives in Appendix H.7.1. We note that SD losses are more expensive to evaluate than the standard FM loss: the LSD and ESD losses indeed require computing the derivatives of u^θ with respect to time and space (which can be efficiently implemented via jvp operations at the cost of an additional backward pass), while the SSD loss requires two extra evaluations of u^θ.

Training & Inference. Learning the Flow Map amounts to simply combining the flow matching loss $\mathcal{L}_{\mathrm{FM}}$ and the self-distillation loss $\mathcal{L}_{\mathrm{SD}}$ leading to the optimization problem

$$\arg\min_\theta \mathbb{E}[\mathcal{L}_{\text{FM-SD}}(u^\theta_{s,t}, \mathrm{I}_t)] \,, \; \mathcal{L}_{\text{FM-SD}}(u^\theta_{s,t}, \mathrm{I}_t) = \begin{cases} \mathcal{L}_{\mathrm{FM}}(u^\theta_{t,t}, \mathrm{I}_t) & \text{if } s = t \,, \\ \mathcal{L}_{\mathrm{SD}}(u^\theta_{s,t}, \mathrm{I}_t) & \text{else} \,, \end{cases} \tag{H.10}$$

where the expectation is taken over $(X_0, X_1) \sim \pi_{0,1}$ and $(s,t) \sim \bar{q}(s,t)$, $\bar{q}$ being a predefined joint distribution over $\{(s,t) \in [0,1]^2 : s \le t\}$. Due to the Euler-like parameterization given in (H.8), inference with a Flow Map model is straightforward. Given a time discretization $\{t_k\}_{k=0}^K$ of $[0,1]$ of size $(K+1)$, with $K \ge 1$, $t_0 = 0$ and $t_K = 1$, and an initial sample $x_K \sim \pi_1$, one can recursively compute the iterates $\{x_k\}_{k=0}^K$ via the update

$$x_k = x_{k+1} - \delta_k u^\theta_{t_k, t_{k+1}}(x_{k+1}) \,, \tag{H.11}$$

where $\delta_k = t_{k+1} - t_k$ denotes the k-th step-size. In particular, x_0 is approximately sampled from π_0 if u^θ accurately approximates u. Importantly, this formulation avoids the discretization error inherent to standard FM, allowing the number of inference steps K to remain very small, i.e., $K \in \{1, 2, 4\}$ [Boffi et al., 2025a, Sabour et al., 2025]. As in the FM setting, Flow Map models admit direct x_0-prediction at any intermediate step of the ODE trajectory; indeed, given a current state (x_t, t), the corresponding Flow Map prediction is given by

$$\hat{x}_0(x_t) = x_t - t\, u^\theta_{0,t}(x_t) \,. \tag{H.12}$$

Unlike the FM prediction in (H.4), the Flow Map prediction (H.12) is *exact* for any given timestep $t \in [0,1]$.

H.3.3 Generalization of CFG to the Flow Map setting

Assume now that each training sample $(x_0, x_1) \in \mathcal{D}$ is provided with an additional knowledge $\mathbf{c}$ (e.g., a class conditioning), that we wish to exploit to improve sample quality. In the FM setting, the CFG paradigm discussed in Appendix H.3.1 has been shown to bring substantial gains in image generation tasks, which naturally motivates its use within Flow Map models, especially in the context of SR. However, its application is not straightforward, as CFG is originally formulated for instantaneous velocity fields, whereas Flow Map models operate on averaged velocity fields over time intervals. Building on the insights of Geng et al. [2025] in the ESD setting, we propose to integrate CFG directly into Flow Map models at *training stage*, by exploiting the connection between FM and Flow Map formulations across all three SD settings. This leads to new CFG-aware expressions for the LSD and SSD objectives. To promote model flexibility and reduce the risk of overfitting, we further propose to *randomly* sample the guidance scale w during training from the interval $(1, w_{\max})$, where $w_{\max} > 1$ is treated as a hyperparameter[1]. This design prevents specialization to a single guidance strength while still retaining the benefits of CFG at inference. Below, we detail the derivation of the Flow Map training objectives in the CFG setting.

[1] In contrast, Geng et al. [2025] adopt a deterministic training procedure with a fixed guidance scale.

Flow matching loss. Given a guidance scale $w \in (1, w_{\max})$, we rewrite the loss $\mathcal{L}_{\text{cFM}}$ in (H.6) to directly introduce CFG at the velocity level, yielding the following objective

$$\mathcal{L}_{\text{cfg-FM}}(u^\theta_{t,t}, \mathrm{I}_t, \mathbf{c}, w) = \|u^\theta_{t,t}(\mathrm{I}_t|\mathbf{c}) - \text{sg}(v^{\text{target}}_{\text{cfg-FM}})\|^2 \ , \ v^{\text{target}}_{\text{cfg-FM}} = w\, v^{\text{target}}_{\text{FM}} + (1-w)u^\theta_{t,t}(\mathrm{I}_t|\emptyset) \ . \tag{H.13}$$

This loss naturally extends the standard CFG formula (H.5) to the Flow Map setting, by leveraging the identity $v_t = u_{t,t}$. Compared to the unconditional setting, each loss evaluation now requires one extra model evaluation.

Self-distillation losses. In the same spirit as in the FM setting, we derive the following CFG-based SD loss

$$\mathcal{L}_{\text{cfg-SD}}(u^\theta_{s,t}, \mathrm{I}_t, \mathbf{c}, w) = \|u^\theta_{s,t}(\mathrm{I}_t|\mathbf{c}) - \text{sg}(u^{\text{target}}_{\text{cSD}})\|^2 \ , \tag{H.14}$$

where $u^{\text{target}}_{\text{cSD}}$ still depends on the chosen characterization:

1. CFG-based *Lagrangian* SD (cfg-LSD) loss

$$u^{\text{target}}_{\text{cfg-LSD}} = v^{\text{target}}_{\text{cfg-FM}} + (t-s)\partial_s u^\theta_{s,t}(\mathrm{I}_t|\mathbf{c}) \ , \ v^{\text{target}}_{\text{cfg-FM}} = w\, v^{\text{target}}_{\text{FM}} + (1-w)u^\theta_{s,s}\left(\mathrm{I}_t - (t-s)u^\theta_{s,t}(\mathrm{I}_t|\emptyset) \mid \emptyset\right) \ ,$$

2. CFG-based *Eulerian* SD (cfg-ESD) loss, already given in [Geng et al., 2025, Eq. (17)]

$$u^{\text{target}}_{\text{cfg-ESD}} = v^{\text{target}}_{\text{cfg-FM}} - (t-s)\left(\nabla u^\theta_{s,t}(\mathrm{I}_t|\mathbf{c})v^{\text{target}}_{\text{cfg-FM}} + \partial_t u^\theta_{s,t}(\mathrm{I}_t|\mathbf{c})\right) \ , \ v^{\text{target}}_{\text{cfg-FM}} = w\, v^{\text{target}}_{\text{FM}} + (1-w)u^\theta_{t,t}(\mathrm{I}_t|\emptyset) \ ,$$

3. CFG-based *Shortcut* SD (cfg-SSD) loss

$$u^{\text{target}}_{\text{cfg-SSD}} = \frac{1}{2}u^\theta_{r,t}(\mathrm{I}_t|\mathbf{c}) + \frac{1}{2}u^\theta_{s,r}(\hat{\mathrm{I}}_s|\mathbf{c}), \ \hat{\mathrm{I}}_s = \mathrm{I}_t - \frac{t-s}{2}u^\theta_{r,t}(\mathrm{I}_t|\mathbf{c}) \ , \ r = \frac{s+t}{2} \ .$$

Detailed derivations of these objectives are provided in Appendix H.7.2. Relative to the unconditional setting, computing the SD loss incurs two additional model evaluations in the Lagrangian setting, one in the Eulerian setting, and none in the Shortcut setting.

Training & Inference. When combining both objectives previously established, we obtain the CFG-based objective

$$\arg\min_\theta \mathbb{E}[\mathcal{L}_{\text{cfg-FM-SD}}(u^\theta_{s,t}, \mathrm{I}_t, \mathbf{c}, w)] \ , \ \mathcal{L}_{\text{cfg-FM-SD}}(u^\theta_{s,t}, \mathrm{I}_t, \mathbf{c}, w) = \begin{cases} \mathcal{L}_{\text{cfg-FM}}(u^\theta_{t,t}, \mathrm{I}_t, \mathbf{c}, w) & \text{if } s = t \ , \\ \mathcal{L}_{\text{cfg-SD}}(u^\theta_{s,t}, \mathrm{I}_t, \mathbf{c}, w) & \text{else,} \end{cases}$$

where the expectation is taken over $(X_0, X_1, \mathbf{c}) \sim \pi_{0,1}$, $(s,t) \sim \bar{q}(s,t)$ and $w \sim \mathrm{U}(1, w_{\max})$. Analogously to standard CFG practice, we randomly drop the class condition $\mathbf{c}$ in (H.13) and (H.14) with a probability of 10%, ensuring that the drop is applied consistently across all evaluations of u^θ within a given setting. When this occurs, we additionally set $w = 1$, thereby recovering the original *unconditional* FM formulation for (H.13) and the corresponding *unconditional* SD formulation for (H.14). We observed in early experiments that this design choice significantly improves the quality of the generated samples. One could further incorporate additional evaluations of the instantaneous velocity $u^\theta_{t,t}(\cdot|\mathbf{c})$ or $u^\theta_{s,s}(\cdot|\mathbf{c})$ when computing $v^{\text{target}}_{\text{FM}}$, at the cost of introducing an additional hyperparameter, as discussed in [Geng et al., 2025, Appendix B.1] for the ESD setting. To avoid extra hyperparameter tuning, we do not explore this extension and leave it for future work.

Importantly, the proposed CFG-based Flow Map formulation avoids the inference-time cost increase typically associated with traditional CFG. In particular, it incurs the same computational cost as the unconditional setting, allowing us to benefit from CFG without additional overhead. Given an initial sample $x^{\mathbf{c}}_K \sim \pi_1$ with class condition $\mathbf{c}$, the unconditional update rule (H.11) naturally becomes

$$x^{\mathbf{c}}_k = x^{\mathbf{c}}_{k+1} - \delta_k u^\theta_{t_k,t_{k+1}}(x^{\mathbf{c}}_{k+1}|\mathbf{c}) \ . \tag{H.15}$$

Similarly, the x_0-prediction defined in (H.12) extends to the conditional setting as

$$\hat{x}_0(x_t|\mathbf{c}) = x_t - t\, u^\theta_{0,t}(x_t|\mathbf{c}) \ . \tag{H.16}$$

H.4 Methodology

We propose to address the SR task within the Flow Map framework. In this setting, each paired sample $(x_0, x_1) \sim \pi_{0,1}$ consists of a low-resolution (LR) image $x_1 \in \mathbb{R}^{3\times w\times h}$ and its high-resolution (HR) counterpart $x_0 \in \mathbb{R}^{3\times W\times H}$, with a fixed aspect ratio satisfying $H/h = W/w \geq 1$. We refer to this ratio as the SR "upscaling factor" and denote it by s_{up}, while its inverse $s_{\text{down}} = 1/s_{\text{up}}$ corresponds to the "downscaling factor". In Appendix H.4.1, we introduce a complete methodology tailored to the SR setting, which we term *FlowMapSR*. To further enhance both reconstruction faithfulness and perceptual quality, we extend FlowMapSR with a positive–negative prompting strategy inspired by Zhang et al. [2024a], described in Appendix H.4.2, as well as an additional adversarial fine-tuning stage detailed in Appendix H.4.3.

H.4.1 Base implementation of FlowMapSR

Data. Each training pair in our dataset $\mathcal{D}$ consists of a ground-truth HR image x_0 and a corresponding LR image x_1, synthetically generated by applying a sequence of random degradations to x_0, including downscaling by a factor s_{down}. We build on the widely used degradation pipeline of Wang et al. [2021], which we adapt to better reflect real-world LR images similar to those shown in Figure H.4. In particular, we reduce the strength of noise components (e.g., Gaussian noise, gray noise, and blur) to emphasize the effects of downscaling and compression; implementation details are provided in Appendix H.8. To improve robustness across diverse SR scenarios and promote generalization, we further adopt the continuous SR strategy of Gao et al. [2023] by sampling $s_{\text{down}} \sim \mathrm{U}(0.1, 1)$ independently for each training pair.

The resulting LR image is then resized back to the spatial resolution of x_0 using a randomly selected interpolation mode (nearest, area, bilinear, or bicubic), followed by additional blur and pixel-value clamping. This procedure yields an LR source x_1 with the same spatial dimensions as x_0, which is required by the Flow Map framework, while implicitly defining a continuous upscaling factor $s_{\text{up}} \sim \mathrm{U}(1, 10)$. Although this setup theoretically enables SR up to $\times 10$, we restrict our evaluation to $\times 4$ and $\times 8$ upscaling, which are the most common scenarios in practice and facilitate comparison with existing methods. Finally, to enable large-scale training and inference, we operate in latent space by encoding image pairs with a non-trainable variational encoder. As a result, both training and inference of the Flow Map models described in Appendix H.3 are conducted in latent space, with a final decoding step, denoted by **Dec**, applied at inference time to recover the HR image.

Time-dependent regularization with perceptual loss. Following prior work [Wu et al., 2024a, Zhang et al., 2024a], we augment the standard Flow Map training objective (H.10) with an additional LPIPS perceptual loss [Zhang et al., 2018], which aligns the decoded HR prediction with the ground-truth HR image. This regularization significantly accelerates convergence and improves the recovery of meaningful image features at inference. The resulting objective is

$$\mathcal{L}^{\text{reg}}_{\text{FM-SD}}(u^\theta_{s,t}, \mathrm{I}_t) = \mathcal{L}_{\text{FM-SD}}(u^\theta_{s,t}, \mathrm{I}_t) + \lambda^{\text{LPIPS}}_s \,\text{LPIPS}(\hat{X}_0, X_0)\ , \text{where } \hat{X}_0 = \mathbf{Dec}(\mathrm{I}_t - t\, u^\theta_{s,t}(\mathrm{I}_t))\ ,$$

where $\lambda^{\text{LPIPS}} : [0,1] \to [0,\infty)$ is a time-dependent weight and $\hat{X}_0$ denotes an approximate pixel-space x_0-prediction from I_t. Unlike previous approaches that use a constant λ^{LPIPS}, we deliberately design this weight to be *strictly decreasing* with time, reflecting the accuracy of direct x_0 prediction from I_t under our Flow Map model: (a) when $s = t$ (FM case), $\hat{X}_0$ corresponds to the decoded approximation in (H.4), whose accuracy decreases as s increases; accordingly, λ^{LPIPS} is set higher for small s and lower for large s; (b) when $s < t$ (SD case), $\hat{X}_0$ coincides with the exact prediction in (H.12) only for $s = 0$, but remains reliable for small s, motivating a similar time-dependent weighting. The specific form of λ^{LPIPS} used in our experiments is provided in Appendix H.8.

Dynamic loss weighting. To ensure meaningful self-consistency, the FM-SD loss must be evaluated over a broad range of timestep pairs (s, t). In practice, this can induce strong time-dependent variance that hampers stable optimization. To mitigate this issue, we follow Boffi et al. [2025a], inspired by Karras et al. [2024], and jointly learn a *dynamic weighting* function $\lambda^\psi_{s,t}$, parameterized as a lightweight neural network described in Appendix H.8, alongside the Flow Map model $u^\theta_{s,t}$. This leads to the reweighted objective

$$\mathcal{L}(\theta, \psi) = \mathbb{E}[\mathcal{L}(u^\theta_{s,t}, \mathrm{I}_t; \psi)]\ ,\ \ \mathcal{L}(u^\theta_{s,t}, \mathrm{I}_t; \psi) = \exp(-\lambda^\psi_{s,t})\mathcal{L}^{\text{reg}}_{\text{FM-SD}}(u^\theta_{s,t}, \mathrm{I}_t) + \lambda^\psi_{s,t}\ . \tag{H.17}$$

At optimality, this formulation encourages loss gradients with respect to θ to have comparable magnitudes across different (s, t) values [Karras et al., 2024, Eq. (26)], thereby stabilizing training.

Discrete timestep sampling. As described in the following section, our experiments rely on models with discrete time inputs, which precludes the use of continuous-time sampling strategies for designing $\bar{q}$ [Boffi et al., 2025a]. Instead, inspired by Frans et al. [2025], we sample timestep pairs (s, t) by first selecting a discretization level d of the interval $[0, 1]$. Each level d is drawn from the set $0, 1, \ldots, d_{\max}$, where $d_{\max} \geq 1$ is a hyperparameter. This induces a grid of $2^d + 1$ uniformly spaced timesteps on $[0, 1]$, allowing up to $K_d = 2^d$ inference steps. In this formulation, $d = 0$ corresponds to one-step inference, $d = 1$ to two-step inference, and so on. To prevent the self-distillation loss from overwhelming the FM loss, we adopt a non-uniform allocation between FM and SD training samples by fixing the probability of sampling boundary cases with $s = t$. Following prior work, we set this probability to 75% in all experiments, resulting in 75% FM samples and 25% SD samples. Details on how this scheme is adapted to each SD setting are provided in Appendix H.8.

FM training warm-start. Finally, to prevent the self-distillation objective from dominating early training dynamics, we initialize each Flow Map model by training *without* the SD component in the FM-SD loss for a predefined number of iterations. In our experiments, we first train our model for 5,000 gradient steps using the FM objective, followed by an additional 5,000 gradient steps using the FM-SD loss. Further implementation details are provided in Appendix H.8.

H.4.2 Enhancing FlowMapSR via positive-negative prompting guidance

Our base implementation of FlowMapSR, described in Appendix H.4.1, is unconditional in the sense that it takes only the LR image as input and predicts the corresponding HR image at inference using the update rule (H.11). While this formulation effectively captures meaningful structures from the LR input, see Figure H.2, and renders them in a visually plausible manner, the resulting images often lack sharpness and contrast, and therefore fall short in terms of photorealism. To address this limitation, we propose to incorporate CFG into FlowMapSR by adapting the positive–negative prompting strategy introduced in Zhang et al. [2024a] to our framework, as detailed below. Specifically, we consider the following two generic text prompts:

- a negative prompt $\mathbf{c}^{\text{neg}}$: *"oil painting, cartoon, blur, dirty, messy, low quality, deformation, low resolution, oversmooth"*
- a positive prompt $\mathbf{c}^{\text{pos}}$: *"a high-resolution, 8K, ultra-realistic image with sharp focus, vibrant colors, and natural lighting"*

The objective is to enhance the performance of FlowMapSR conditioned on $\mathbf{c}^{\text{pos}}$ by explicitly modeling poor visual quality through $\mathbf{c}^{\text{neg}}$, and enforcing a strong separation between the two. This strategy provides an effective guidance signal while avoiding reliance on external label extractors, which may be inaccurate or brittle.

Following the CFG formalism introduced in Appendix H.3.3, we treat $\mathbf{c}^{\text{pos}}$ as the *target* conditioning (replacing $\mathbf{c}$) and $\mathbf{c}^{\text{neg}}$ as the *unconditional* conditioning (replacing $\emptyset$). To explicitly learn the notion of degraded quality, we introduce a key modification to the CFG–FM–SD loss related to the SR setting: when the class condition $\mathbf{c}^{\text{pos}}$ is dropped and $\mathbf{c}^{\text{neg}}$ is used instead, we replace the HR target x_0 with a synthetic low-quality version $x_0^{\text{neg}} \neq x_1$. This target is constructed to be less degraded than x_1, ensuring that the model does not learn a trivial noise trajectory. Concretely, x_0^{neg} is obtained by applying the same degradation pipeline used to generate x_1, except that the downscaling factor is sampled as $s_{\text{down}}^{\text{neg}} \sim \mathrm{U}(s_{\text{down}}, 1)$, where s_{down} denotes the downscaling factor associated with x_1. This design guides the unconditional (negative) model toward generating low-quality versions of the HR image, while still preserving informative structure relative to the input x_1 through the difference in upscaling factors. At inference time, we exclusively use the Flow Map model conditioned on $\mathbf{c}^{\text{pos}}$ to generate HR images via the update rule (H.15). As a result, each text-conditioned Flow Map model (LSD, ESD, or SSD) is enhanced with CFG so as to improve perceptual quality relative to its unconditional counterpart. In early experiments, however, we observe that this strategy is beneficial only for the *Shortcut* formulation, as illustrated in Figure H.2. In contrast, the *Lagrangian* and *Eulerian* formulations exhibit noticeable artefacts when combined with CFG; qualitative examples illustrating this failure are provided in Appendix H.9.

We attribute the poor performance of our *Lagrangian* and *Eulerian* CFG-based formulations to unstable training dynamics, likely amplified by the use of JVP operators, which are challenging to handle in practice. This instability is consistent with observations reported in prior unconditional implementations, see [Boffi et al., 2025a, Figure 4], and could potentially be mitigated through specialized training strategies, such as those proposed in [Sabour et al., 2025, Section 3.4]. In contrast, out *Shortcut* CFG-based formulation exhibits stable training behavior without requiring additional heuristics, which may explain its superior empirical performance. In the following, we denote by $\mathcal{L}(u_{s,t}^{\theta}, \mathrm{I}_t, \mathbf{c}, w; \psi)$ the CFG-based adaptation of the previously established loss $\mathcal{L}(u_{s,t}^{\theta}, \mathrm{I}_t; \psi)$ given in (H.17) with positive-negative conditioning $\mathbf{c}$ and guidance scale w, where $\mathcal{L}_{\text{FM-SD}}$ is now replaced by $\mathcal{L}_{\text{cfg-FM-SD}}$.

Figure H.2: **Qualitative comparison of positive-negative CFG enhancement applied to LSD-, ESD-, and SSD-based FlowMapSR for $\times 4$ upscaling.** The HR reference image exhibits sharp edges and rich high-frequency details. The base FlowMapSR model ("Without CFG") successfully recovers structural details but lacks perceptual sharpness. Applying positive–negative CFG improves visual realism in the *Shortcut* formulation (e.g., more natural hair textures), while it introduces noticeable degradations and artefacts in the *Lagrangian* and *Eulerian* formulations.

H.4.3 Enhancing FlowMapSR via LoRA-based adversarial fine-tuning

Although positive–negative CFG improves the performance of the *Shortcut*-based formulation of FlowMapSR, we still observe residual artefacts in the generated HR images, which are attributable to the guidance mechanism itself. To further mitigate these artefacts and strengthen the alignment between the model outputs and the ground-truth HR distribution, we introduce an additional fine-tuning stage that augments the CFG-based training objective with an adversarial loss, treating the Flow Map model as the generator.

Inspired by recent work Sabour et al. [2025], we adopt the Relativistic Pairing GAN (RPGAN) framework [Jolicoeur-Martineau, 2018], which has been shown to be more robust to mode collapse than standard GAN formulations [Sun et al., 2020]. The adversarial training is performed in latent space. Given a discriminator $\mathbf{D}^\phi$ parameterized by a neural network (described in Appendix H.8), we define the following adversarial objective, with separate generator (G-loss) and discriminator (D-loss) terms:

$$\mathcal{L}^{\text{adv}}(u^\theta_{s,t}, \mathrm{I}_t, \mathbf{c}, w; \psi, \phi) = \begin{cases} \text{Softplus}\left(\mathbf{D}^\phi(\hat{Z}_0^{\text{adv}}) - \mathbf{D}^\phi(Z_0)\right) + \lambda^{\text{adv}}\mathcal{L}(u^\theta_{s,t}, \mathrm{I}_t, \mathbf{c}, w; \psi) & \text{(G-loss)} \\ \text{Softplus}\left(\mathbf{D}^\phi(Z_0) - \mathbf{D}^\phi(\text{sg}(\hat{Z}_0^{\text{adv}}))\right) & \text{(D-loss)}, \end{cases}$$

where $\lambda^{\text{adv}} > 0$ controls the contribution of the original loss, Z_0 denotes the latent encoding of X_0, and $\hat{Z}_0^{\text{adv}}$ is the latent x_0-prediction produced by the FlowMapSR model from X_1. To fully exploit the expressivity of the Flow Map model, we compute $\hat{Z}_0^{\text{adv}}$ using a two-step prediction (rather than single-step), following (H.15). In particular, the intermediate state at $t = 1/2$ is detached to avoid the computation of higher-order derivatives, in line with Shen et al. [2025]. The parameter λ^{adv} balances adversarial alignment with preservation of the pretrained FlowMapSR behavior, as excessive adversarial pressure may induce divergence from the original model. As in standard GAN settings, we consider reaching adversarial convergence when we have equality between the "fake" score $\mathbf{D}^\phi(\hat{Z}_0^{\text{adv}})$ and the "real" score $\mathbf{D}^\phi(Z_0)$.

Owing to the relative nature of RPGAN, which promotes stable training dynamics, we alternate optimization of the G-loss and D-loss at each training step. We found it unnecessary to include additional R_1 or R_2 regularization terms in the discriminator loss; however, pretraining the discriminator for a small number of iterations was observed to accelerate convergence, and we therefore adopt this strategy in all experiments.

Table H.1: **Comparison of inference efficiency between FlowMapSR and competing SR methods.** We report the average inference time measured on LR RealSR inputs for ×4 upscaling (input resolution 128 × 128) using a single H100 GPU. For FlowMapSR, we evaluate three variants with 1, 2, and 4 inference steps. FlowMapSR-1 achieves inference time comparable to OSEDiff-1, while FlowMapSR-2 and FlowMapSR-4 remain competitive with S3Diff-1.

Method	Real-ESRGAN	OSEDiff-1	S3Diff-1	**FlowMapSR-1**	**FlowMapSR-2**	**FlowMapSR-4**
Inference time (s)	0.05	0.14	0.40	0.14	0.22	0.40

Finally, to keep the fine-tuning stage lightweight while preserving the knowledge acquired by the CFG-based Flow Map model, we restrict adversarial training to Low-Rank Adaptation (LoRA) layers [Hu et al., 2022] within the generator, following the strategy from Chadebec et al. [2025a]. This choice reduces computational cost during training and enables precise control of the LoRA scaling factor at inference time, allowing us to modulate the impact of the adversarial refinement.

H.5 Experiments

In this section, we evaluate FlowMapSR on synthetic and real-world LR inputs for ×4 and ×8 super-resolution. We first describe the experimental setup in Appendix H.5.1, then present a quantitative comparison with competing SR methods in Appendix H.5.2, complemented by representative qualitative examples (with additional results in Appendix H.9, including ×2 super-resolution). These visual comparisons emphasize the recovery of lifelike textures (e.g., fur, human skin, metals) and photorealistic attributes (e.g., depth of field, contrast, color coherence). Finally, in Appendix H.5.3, we study the key hyperparameters of FlowMapSR that control the trade-off between reconstruction faithfulness and perceptual quality, including the number of inference steps, the LoRA scaling factor at inference, and the maximum positive-negative CFG scale.

H.5.1 Experimental setup

FlowMapSR framework. Throughout this section, our default Flow Map model corresponds to the *Shortcut* variant of FlowMapSR, augmented with positive–negative CFG (Appendix H.4.2) and adversarial fine-tuning (Appendix H.4.3). We do not include results for the *Eulerian* and *Lagrangian* variants, as they consistently underperform on the SR task compared to the *Shortcut* formulation (see Figure H.2). Unless stated otherwise, all reported results are obtained by using the guidance strength $w_{\max} = 3.5$, setting the LoRA scale as 1.5 and applying 2 inference steps (denoted by *FlowMapSR-2*). The model adopts a UNet architecture following SDXL [Podell et al., 2024], initialized from the image restoration weights of Chadebec et al. [2025b]. Training is conducted on a single H100 GPU using HR targets of up to 1024 × 1024 resolution; additional training details and hyperparameter settings are provided in Appendix H.8. Notably, the same model is used for both ×4 and ×8 upscaling, without any scale-specific conditioning. At inference time, LR inputs are first resized in pixel space using bicubic interpolation according to the target scale, then processed in latent space by the Flow Map model, and finally decoded to obtain the HR output. When memory constraints arise due to large target resolutions, inference is performed after decoding on pixel-space tiles, followed by Gaussian blending.

Testing details. For synthetic evaluation with available ground truth, we generate LR–HR pairs from the DIV2K validation (DIV2K-Val) set [Agustsson and Timofte, 2017], RealSR [Cai et al., 2019], and DRealSR [Wei et al., 2020], using the degradation pipeline described in Appendix H.4.1. We recall that this pipeline is a lighter variant of the one proposed in Wang et al. [2021], with reduced additional noise, see Appendix H.8 for more details. Performance on synthetic datasets is assessed using both reference metrics—PSNR, SSIM [Wang et al., 2004], LPIPS [Zhang et al., 2018], DISTS [Ding et al., 2020], and FID [Heusel et al., 2017]—which measure fidelity to the HR ground truth, and non-reference metrics—NIQE [Zhang et al., 2015], MANIQA [Yang et al., 2022], MUSIQ [Ke et al., 2021], and CLIPIQA [Wang et al., 2023]—which better reflect perceptual quality. We additionally present extensive qualitative results on DIV2K-Val and RealSR. For real-world evaluation, we use the RealSet65 dataset [Yue et al., 2023] and report qualitative comparisons. FlowMapSR is compared against the GAN-based method Real-ESRGAN [Wang et al., 2021] and state-of-the-art single-step diffusion-based methods OSEDiff [Wu et al., 2024a] and S3Diff [Zhang et al., 2024a] (respectively denoted OSEDiff-1 and S3Diff-1 to emphasize their single-step nature). For fairness, we rely on publicly available code and pretrained checkpoints to generate HR outputs on the same test sets. While FlowMapSR differs substantially from prior distillation-based approaches in training budget (under 100M parameters for OSEDiff and S3Diff versus over 2B for FlowMapSR), its inference time remains competitive for the default FlowMapSR-2 model, as reported in Table H.1.

Table H.2: **Quantitative comparison between FlowMapSR and competing SR methods (×4 and ×8 upscaling).** The best and second best results are highlighted in **bold** and underlined, respectively.

Dataset (s_{up})	Method	Reference metrics					Non-reference metrics			
		PSNR↑	SSIM↑	LPIPS↓	DISTS↓	FID↓	NIQE↓	MANIQA↑	MUSIQ↑	CLIPIQA↑
DIV2K-Val (×4)	Real-ESRGAN	**23.03**	**0.6880**	0.2739	0.1276	21.97	3.758	0.3536	0.6112	0.4960
	OSEDiff-1	21.89	0.6500	0.2625	0.1107	19.23	3.373	**0.4363**	0.6777	**0.5974**
	S3Diff-1	20.87	0.6223	**0.2475**	0.1093	15.39	3.198	0.4223	**0.6780**	0.5893
	FlowMapSR-2	22.12	0.6412	0.2554	**0.0995**	**13.05**	**3.036**	0.4010	0.6431	0.5419
RealSR (×4)	Real-ESRGAN	**22.28**	**0.7047**	0.2743	0.1939	126.70	5.770	0.4253	0.6497	0.4575
	OSEDiff-1	21.15	0.6610	0.2752	0.1907	106.10	**5.180**	0.4936	0.6925	0.5288
	S3Diff-1	21.75	0.6762	**0.2585**	**0.1775**	**94.32**	5.661	0.5075	**0.6986**	0.5511
	FlowMapSR-2	20.42	0.5965	0.3011	0.1964	95.16	5.322	**0.5362**	0.6975	**0.5901**
DrealSR (×4)	Real-ESRGAN	**25.22**	**0.7771**	**0.2700**	0.1875	140.31	6.954	0.3876	0.6001	0.4581
	OSEDiff-1	23.54	0.7173	0.2884	0.1894	114.69	5.745	0.4758	0.6565	0.5634
	S3Diff-1	24.36	0.7200	0.2886	**0.1818**	**105.48**	**5.734**	0.4862	**0.6720**	0.5771
	FlowMapSR-2	23.15	0.6688	0.3056	0.1937	108.74	5.779	**0.4863**	0.6600	**0.5894**
DIV2K-Val (×8)	Real-ESRGAN	**20.55**	**0.6061**	0.3940	0.1882	40.98	5.209	0.2505	0.4612	0.3317
	OSEDiff-1	19.68	0.5567	0.3909	0.1667	43.41	3.394	**0.4080**	**0.6511**	**0.5759**
	S3Diff-1	19.30	0.5386	**0.3743**	0.1762	47.49	**3.069**	0.3468	0.6227	0.5712
	FlowMapSR-2	19.90	0.5671	0.3767	**0.1469**	**28.25**	3.197	0.3793	0.6135	0.5008
RealSR (×8)	Real-ESRGAN	**19.72**	**0.6014**	**0.3536**	**0.2327**	179.34	6.294	0.2594	0.4787	0.2865
	OSEDiff-1	18.86	0.5294	0.4174	0.2526	180.05	4.592	0.4396	0.6454	0.5011
	S3Diff-1	18.93	0.5224	0.3828	0.2335	172.07	**4.208**	0.4298	0.6546	0.5217
	FlowMapSR-2	17.96	0.4878	0.4190	0.2576	**163.35**	5.173	**0.5042**	**0.6773**	**0.5746**
DrealSR (×8)	Real-ESRGAN	**22.46**	**0.6898**	**0.3378**	**0.2244**	178.19	7.246	0.2612	0.4630	0.3252
	OSEDiff-1	21.11	0.5989	0.4190	0.2520	180.56	4.763	0.4216	0.6153	0.5201
	S3Diff-1	21.17	0.5631	0.4150	0.2405	185.93	**4.390**	0.4099	0.6321	0.5359
	FlowMapSR-2	20.70	0.5858	0.3944	0.2481	**170.81**	5.743	**0.4472**	**0.6365**	**0.5483**

H.5.2 Main results

We report quantitative results for ×4, ×8 upscaling in Table H.2, from which we draw the following observations.

(a) On DIV2K-Val, FlowMapSR achieves strong performance on reference-based metrics, particularly perceptual measures such as LPIPS, DISTS, and FID, compared to competing methods. This high fidelity to the DIV2K-Val ground-truth images is consistent with the qualitative results shown in Figure H.3, see respectively the rows 1-3 for ×4 upscaling and the rows 4-7 for ×8 upscaling. In particular, FlowMapSR exhibits faithful recovery of textures and details (e.g., human skin and attributes, animal fur), geometric structures, lighting, environmental context, and depth of field. While these qualitative examples demonstrate strong photorealism—often surpassing competing methods—this advantage is not always reflected in non-reference metrics, where OSEDiff-1 tends to achieve higher scores. Visually, this may be explained by the stronger sharpness and contrast produced by OSEDiff-1, which benefit such metrics but come at the expense of reference fidelity. We attribute the slightly lower non-reference scores of FlowMapSR in this setting to occasional boundary artefacts introduced by Gaussian tiling, see Appendix H.9, which is not required for processing RealSR and DRealSR images. Improving this tiling strategy is left for future work.

(b) On RealSR and DRealSR, the trend is reversed: FlowMapSR achieves the best average performance on non-reference metrics for ×4 and ×8 upscaling on both datasets, while still maintaining competitive DISTS and FID scores, especially at higher scaling factor. Although Real-ESRGAN attains strong reference-based scores on these datasets, this should be interpreted with caution. Visual inspection in Figure H.3 (row 4 for ×4, and row 8 for ×8) reveals that Real-ESRGAN often produces visibly low-quality outputs. Despite being counter-intuitive, high reference-based scores and poor visual quality are not contradictory here; both reflect the limited visual quality of the RealSR and DrealSR ground truth, thereby revealing the limitations of reference metrics in this case. In contrast, while FlowMapSR may appear less faithful based on reference metrics, it consistently produces higher-quality images, with more realistic textures and convincing depth of field areas, where S3Diff comparatively struggles.

Finally, we present qualitative comparisons on real-world inputs from RealSet65 in Figure H.4, where we show SR results for selected regions of the LR images. In this fully real-world setting, FlowMapSR consistently produces the most photorealistic and faithful outputs for both ×4 and ×8 upscaling, particularly in terms of texture realism and depth of field effects. Overall, FlowMapSR demonstrates strong and consistent performance across both synthetic and real-world scenarios, using a single model without any upscaling-specific conditioning. While this behavior is only partially captured by quantitative metrics, it is clearly reflected in qualitative results. Additional comparisons on DIV2K-Val, RealSR, and RealSet65 for both ×4 and ×8 upscaling are provided in Appendix H.9.

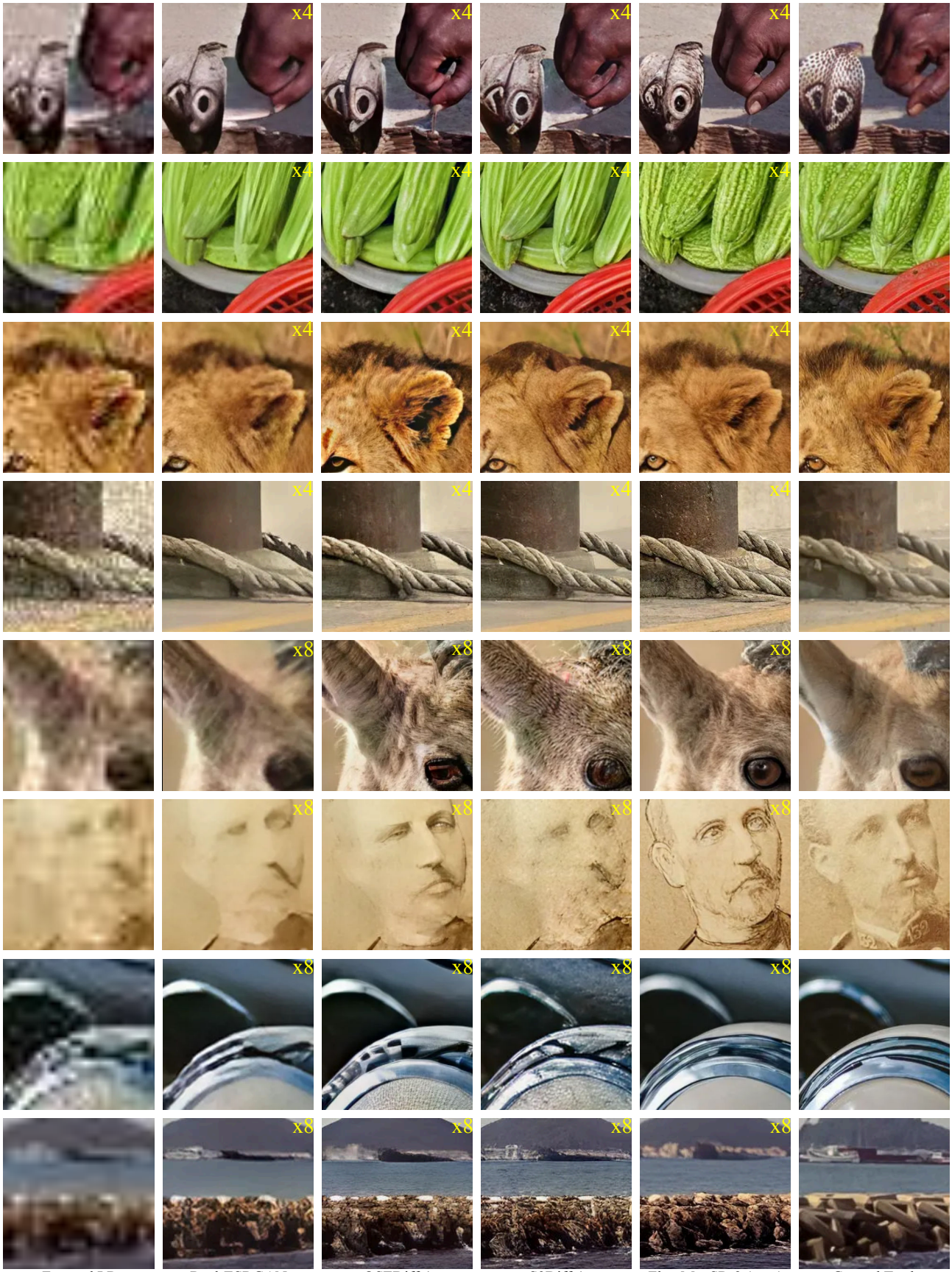


Figure H.3: **Qualitative comparison between FlowMapSR and competing SR methods for $\times 4$ (rows 1–4) and $\times 8$ (rows 5–8) upscaling.** Rows 1–3 and 5–7 show examples from DIV2K-Val, while rows 4 and 8 correspond to RealSR samples.

Figure H.4: **Qualitative comparison between FlowMapSR and competing diffusion-based SR methods on real-world LR inputs for $\times 4$ and $\times 8$ upscaling.** The two LR images are taken from the RealSet65 dataset. For each image, a green region is selected for $\times 4$ upscaling and a blue region for $\times 8$ upscaling. FlowMapSR-2 consistently achieves the best balance between reconstruction faithfulness and photorealism.

H.5.3 Ablation studies

We analyze the influence of three key hyperparameters of FlowMapSR: the number of inference steps, the LoRA scaling factor applied at inference, and the maximum CFG scale used during training. For each setting, we report quantitative results on DIV2K-Val and RealSR for $\times 4$ and $\times 8$ upscaling, complemented by qualitative comparisons for the first two ablations. Additional visual results are provided in Appendix H.9.

Impact of the number of inference steps. Our Flow Map formulation allows the number of inference steps (NFE) to be chosen as a power of two; we evaluate 1, 2, and 4 steps. As shown in Table H.3, using a single step consistently yields higher fidelity to the reference images, while increasing NFE improves model expressivity and perceptual quality, as reflected by non-reference metrics. These results highlight a clear trade-off: increasing NFE is not always beneficial. Qualitative examples in Figure H.5 illustrate that larger NFE can degrade depth-of-field consistency (rows 1 and 3) and introduce overly shiny or glassy textures (rows 2 and 4), whereas a single step, while more faithful, lacks visual refinement. Overall, using two inference steps, as in our default configuration, provides an effective balance between faithfulness and perceptual quality.

Impact of the LoRA scale at inference. A similar trade-off is observed with respect to the LoRA scaling factor, which controls the influence of the adversarially fine-tuned parameters at inference. Increasing the LoRA scale (e.g., from 1 to 2) enhances sharpness and contrast, as illustrated in Figure H.6. However, this comes at the cost of reduced reconstruction faithfulness and, in some cases, degraded photorealism, particularly in depth-of-field rendering (see rows 1 and 2). This effect becomes more pronounced for larger values (e.g., LoRA scale 3), which lead to clearly suboptimal results. The quantitative results in Table H.4 confirm this behavior, with higher reference-based scores at lower LoRA scales and improved non-reference metrics at larger scales. In practice, the default setting with a LoRA scale of 1.5 offers a good compromise between fidelity and perceptual enhancement.

Table H.3: **Quantitative comparison of FlowMapSR with varying number of inference steps (NFE) (×4 and ×8 upscaling).** All FlowMapSR variants use $w_{\max} = 3.5$ and a LoRA scale of 1.5, consistent with the FlowMapSR-2 configuration. The best and second best results are highlighted in **bold** and underlined, respectively. Red cells denote the default FlowMapSR-2 configuration used in Table H.2 and Figure H.3, while blue cells mark the best results when also considering competing methods reported in Table H.2.

Dataset (s_{up})	NFE	Reference metrics					Non-reference metrics			
		PSNR↑	SSIM↑	LPIPS↓	DISTS↓	FID↓	NIQE↓	MANIQA↑	MUSIQ↑	CLIPIQA↑
DIV2K-Val (x4)	1	**22.89**	**0.6703**	**0.2464**	**0.0986**	14.03	3.750	0.3810	0.6248	0.4995
	2	22.12	0.6412	0.2554	0.0995	**13.05**	3.036	0.4010	0.6431	0.5419
	4	20.94	0.5881	0.3033	0.1126	13.55	**2.775**	**0.4134**	**0.6575**	**0.5848**
RealSR (x4)	1	**21.74**	**0.6517**	**0.2636**	**0.1790**	**90.54**	5.630	0.4949	0.6764	0.5196
	2	20.42	0.5965	0.3011	0.1964	95.16	**5.322**	0.5362	0.6975	0.5901
	4	19.15	0.5322	0.3598	0.2174	103.86	5.342	**0.5607**	**0.7095**	**0.6204**
DIV2K-Val (x8)	1	**20.75**	**0.5981**	**0.3568**	0.1499	**27.97**	3.857	0.3537	0.5914	0.4648
	2	19.90	0.5671	0.3767	**0.1469**	28.25	3.197	0.3793	0.6135	0.5008
	4	18.85	0.5223	0.4249	0.1525	29.82	**3.077**	**0.3987**	**0.6311**	**0.5449**
RealSR (x8)	1	**19.28**	**0.5486**	**0.3886**	**0.2458**	166.73	5.758	0.4536	0.6495	0.5178
	2	17.96	0.4878	0.4190	0.2576	**163.35**	**5.173**	0.5042	0.6773	0.5746
	4	16.67	0.4179	0.4773	0.2755	164.99	5.535	**0.5309**	**0.6970**	**0.6127**

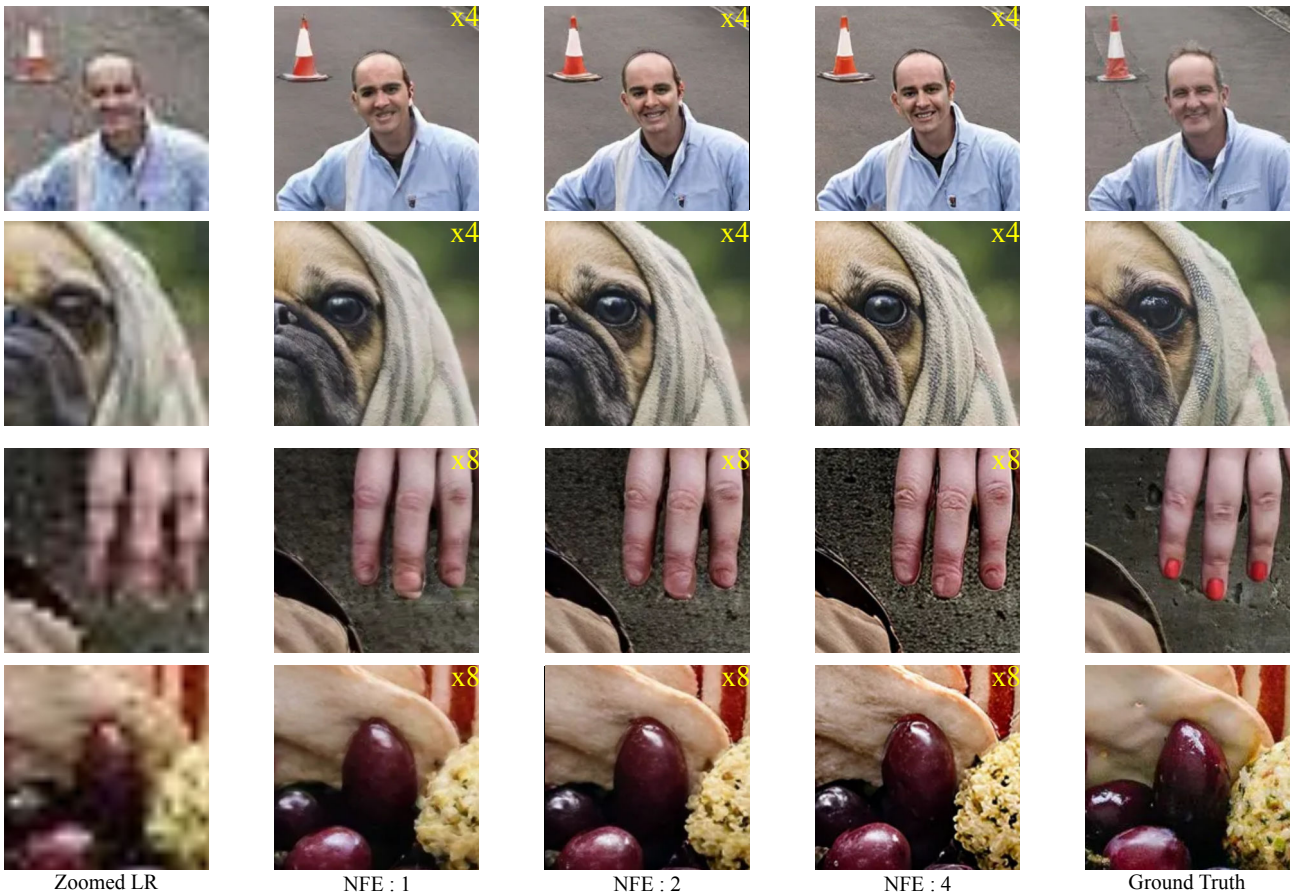


Figure H.5: **Qualitative comparison of FlowMapSR with varying numbers of inference steps (NFE) for ×4 (rows 1–2) and ×8 (rows 3–4) upscaling.** The LR inputs are drawn from the DIV2K-Val dataset. Increasing NFE improves perceptual quality, while fewer steps yield higher faithfulness to the reference, illustrating the trade-off quantified in Table H.3.

Table H.4: **Quantitative comparison of FlowMapSR with varying LoRA scale (×4 and ×8 upscaling).** All FlowMapSR variants use $w_{\max} = 3.5$ with 2 inference steps, consistent with the FlowMapSR-2 configuration. The best and second best results are highlighted in **bold** and underline, respectively. Red cells correspond to the default version of FlowMapSR-2 used in Table H.2 and Figure H.3.

Dataset (s_{up})	LoRA scale	Reference metrics					Non-reference metrics			
		PSNR↑	SSIM↑	LPIPS↓	DISTS↓	FID↓	NIQE↓	MANIQA↑	MUSIQ↑	CLIPIQA↑
DIV2K-Val (x4)	1	**22.29**	**0.6520**	0.2571	**0.0958**	**12.92**	3.170	0.3865	0.6309	0.5138
	1.5	22.12	0.6412	**0.2554**	0.0995	13.05	3.036	0.4010	0.6431	0.5419
	2	21.83	0.6224	0.2675	0.1093	13.43	**2.928**	0.4059	**0.6456**	0.5649
	3	20.21	0.5232	0.3928	0.1800	19.20	3.569	**0.4149**	0.6323	**0.6486**
RealSR (x4)	1	**20.50**	**0.6084**	**0.2891**	**0.1907**	**92.76**	**5.252**	0.5285	0.6961	0.5753
	1.5	20.42	0.5965	0.3011	0.1964	95.16	5.322	**0.5362**	**0.6975**	0.5901
	2	20.22	0.5753	0.3274	0.2109	103.50	5.659	0.5271	0.6860	**0.5967**
	3	19.39	0.5280	0.4140	0.2526	137.74	7.134	0.4991	0.6662	0.6009
DIV2K-Val (x8)	1	**20.08**	**0.5782**	0.3792	**0.1415**	**27.86**	3.313	0.3671	0.6021	0.4721
	1.5	19.90	0.5671	**0.3767**	0.1469	28.25	3.197	0.3793	0.6135	0.5008
	2	19.62	0.5457	0.3888	0.1645	30.45	**3.022**	0.3788	0.6115	0.5367
	3	18.71	0.4670	0.4987	0.2392	43.63	3.592	**0.3938**	**0.6136**	**0.6600**
RealSR (x8)	1	**18.07**	**0.5019**	**0.4041**	**0.2480**	**158.81**	**5.163**	0.4989	**0.6785**	0.5634
	1.5	17.96	0.4878	0.4190	0.2576	163.35	5.173	**0.5042**	0.6773	0.5746
	2	17.80	0.4638	0.4563	0.2797	182.77	5.537	0.4791	0.6580	0.5814
	3	17.76	0.4410	0.5492	0.3148	229.96	6.784	0.4491	0.6226	**0.5954**

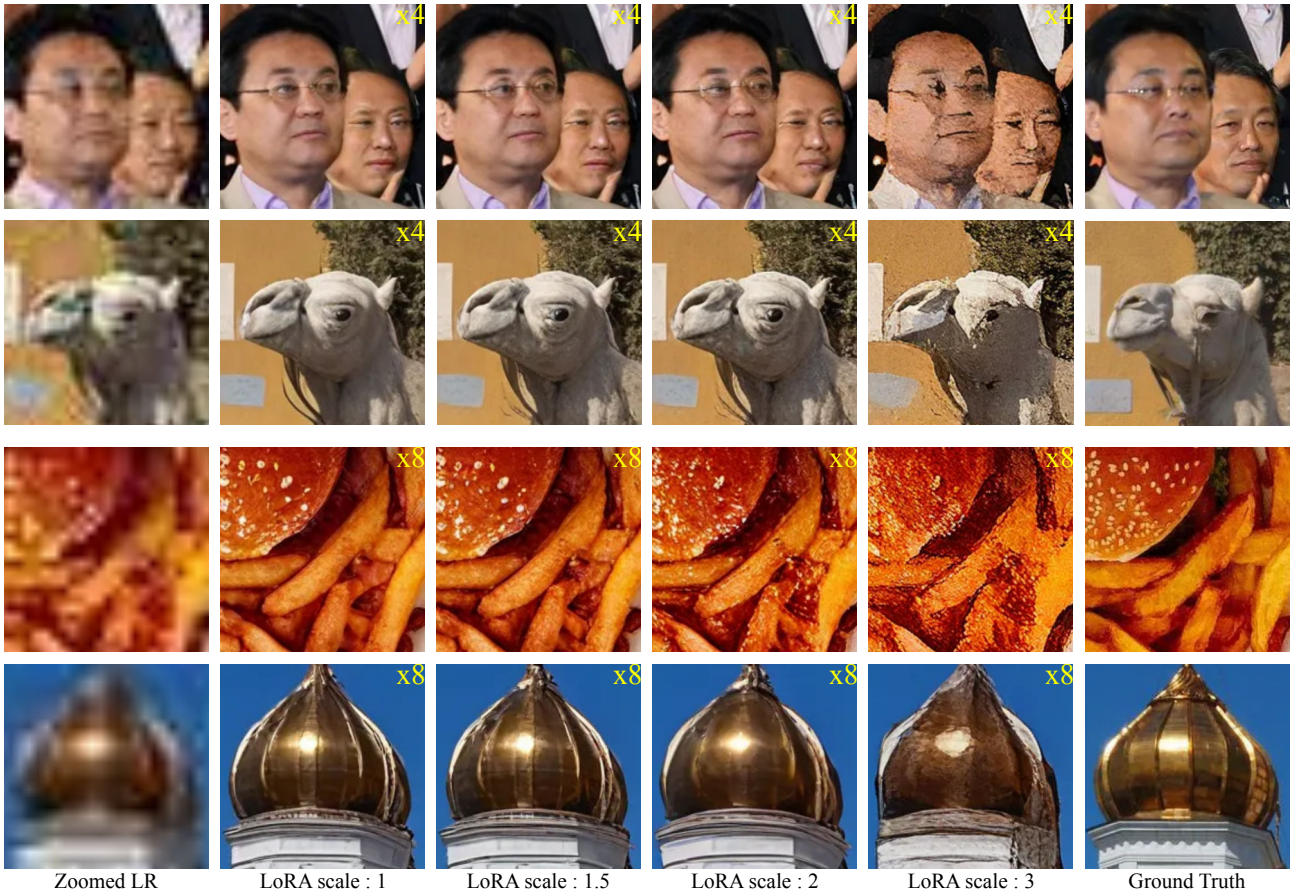


Figure H.6: **Qualitative comparison of FlowMapSR with varying LoRA scale for ×4 (rows 1–2) and ×8 (rows 3–4) upscaling.** Inputs are from the DIV2K-Val dataset. Smaller LoRA scales better preserve fidelity to the reference, while larger scales favor sharpness (eventually degrading quality with LoRA scale set to 3), consistent with the trade-off in Table H.4.

Table H.5: **Quantitative comparison of FlowMapSR with maximal positive-negative CFG scale $w_{\max}$ ($\times 4$ and $\times 8$ upscaling).** The best and second best results are highlighted in **bold** and underlined, respectively. Blue cells mark the best results when also considering competing methods reported in Table H.2. All FlowMapSR variants use 2 inference steps, consistent with the FlowMapSR-2 configuration, but are obtained without adversarial fine-tuning.

Dataset (s_{up})	Max. CFG scale	Reference metrics					Non-reference metrics			
		PSNR↑	SSIM↑	LPIPS↓	DISTS↓	FID↓	NIQE↓	MANIQA↑	MUSIQ↑	CLIPIQA↑
DIV2K-Val (x4)	2	23.15	0.6759	0.2568	0.1015	**11.93**	3.489	0.3484	0.6042	0.4605
	2.5	22.98	0.6675	0.2578	**0.0987**	12.08	3.314	0.3584	0.6068	0.4794
	3	**23.48**	**0.6913**	0.2658	0.1052	12.50	4.052	0.3344	0.5822	0.4415
	3.5	22.12	0.6413	**0.2554**	0.0995	13.05	**3.036**	0.4008	**0.6433**	0.5415
	4	20.37	0.5530	0.3602	0.1612	15.34	5.072	**0.4230**	0.6068	**0.6195**
RealSR (x4)	2	22.32	0.6627	0.2402	0.1692	88.92	4.911	0.4473	0.6630	0.4860
	2.5	22.08	0.6545	0.2450	0.1688	87.15	4.677	0.4634	0.6691	0.5106
	3	**22.77**	**0.6924**	**0.2180**	**0.1582**	**81.01**	**5.278**	0.4435	0.6602	0.4755
	3.5	20.43	0.5968	0.3009	0.1964	94.71	5.331	0.5368	**0.6976**	0.5910
	4	18.75	0.5003	0.4191	0.2469	107.96	8.727	**0.5513**	0.6777	**0.6371**
DIV2K-Val (x8)	2	20.27	0.5807	**0.3634**	**0.1420**	**27.59**	3.331	0.3601	0.6118	0.4737
	2.5	20.45	0.5818	0.3771	0.1494	28.57	3.270	0.3582	0.5926	0.4622
	3	**20.87**	**0.6091**	0.3726	0.1499	28.75	4.158	0.3237	0.5629	0.4183
	3.5	19.90	0.5671	0.3767	0.1469	28.31	**3.199**	0.3794	**0.6135**	0.5006
	4	19.53	0.5334	0.4639	0.1988	30.71	4.553	**0.3907**	0.5634	**0.5510**
RealSR (x8)	2	18.78	0.5241	0.3695	0.2350	**151.44**	4.524	0.4542	0.6673	0.5238
	2.5	18.98	0.5168	0.3835	0.2404	155.66	**4.474**	0.4540	0.6628	0.5383
	3	**19.40**	**0.5656**	**0.3465**	**0.2280**	154.32	5.131	0.4342	0.6590	0.4978
	3.5	17.96	0.4877	0.4188	0.2570	162.22	5.178	0.5034	**0.6776**	0.5749
	4	17.64	0.4316	0.5353	0.2996	175.57	8.143	**0.5140**	0.6408	**0.6261**

Impact of the positive–negative CFG strength. Finally, we observe a similar trade-off when analyzing the effect of the CFG scale used to train our FlowMapSR model, as reported in Table H.5. As expected, increasing the maximum guidance strength $w_{\max}$ consistently improves perceptual quality, as reflected by non-reference metrics. Notably, for $w_{\max} = 4$, FlowMapSR achieves CLIPIQA scores that surpass those of competing methods. However, this improvement comes at the expense of reconstruction faithfulness: reference-based metrics clearly degrade at higher guidance strengths, indicating reduced alignment with the HR ground truth. Conversely, lower CFG scales favor faithfulness, although the optimal setting is not trivial. While $w_{\max} = 2$ yields strong FID and DISTS scores, PSNR and SSIM peak at $w_{\max} = 3$. Overall, setting $w_{\max} = 3.5$, as we do by default, provides a good compromise between perceptual quality and reconstruction fidelity.

H.6 Conclusion

In this paper, we proposed FlowMapSR, a diffusion-based framework for image super-resolution (SR) built on recent Flow Map generative models and explicitly designed for fast inference. Unlike prior diffusion-based SR methods relying on teacher–student distillation—which may incur information loss—FlowMapSR directly trains a large, expressive model while retaining efficient inference. Our framework is applicable to all three Flow Map formulations (*Lagrangian*, *Eulerian*, and *Shortcut*). Importantly, we augment each with two SR-specific enhancements: a positive–negative prompting strategy that relies on a unified formulation of Classifier-Free Guidance (CFG) to Flow Map models, and a lightweight adversarial fine-tuning stage based on LoRA, enabling precise control of perceptual refinement at inference.

Empirically, the *Shortcut* formulation best accommodates these enhancements, while the *Lagrangian* and *Eulerian* variants suffer from unstable training. We conjecture that this limitation could be alleviated through finer-grained loss control [Sabour et al., 2025], which we leave for future work. Using the *Shortcut* variant as our default configuration, FlowMapSR achieves a strong balance between reconstruction faithfulness and photo-realism, outperforming competing diffusion-based SR methods in qualitative realism. It produces more lifelike textures (especially, for human subjects), improved depth-of-field rendering, and fewer overly sharp or glossy artefacts, all using a single model without scale- or degradation-specific conditioning. While training is more expensive than distillation-based approaches, inference time remains competitive. Finally, our ablations studies show that the fundamental SR trade-off is explicitly controlled by FlowMapSR hyperparameters: increasing the number of inference steps, CFG strength, or LoRA scale improves perceptual quality, while lower values favor fidelity.

A current limitation of our approach is the reliance on Gaussian tiling at very high resolutions, which can introduce mild blurring at the boundaries of generated images; such artefacts could likely be alleviated through more advanced tiling strategies. Additionally, our procedure may exhibit noticeable color shifts, a phenomenon commonly observed in diffusion models [Deck and Bischoff, 2023]. In practice, FlowMapSR could be combined with GPU-like post-processing techniques to correct these shifts, as explored for instance in Feydy et al. [2019]. We hope our results motivate further exploration of Flow Map models for other image-to-image translation tasks, such as object removal and relighting, depth and normal estimation.

Organization of the supplementary

The appendix is organized as follows. Appendix H.7 first presents complementary theoretical material related to the Flow Map models introduced in Appendix H.3. Appendix H.8 then provides additional implementation details regarding the data, model architecture, training procedure, and inference setup used in our experiments. Finally, Appendix H.9 reports further qualitative results for $\times 4$ and $\times 8$ super-resolution, including comparisons with competing methods on additional synthetic and real-world samples, as well as extended analyses of the ablation studies discussed in Appendix H.5.

H.7 Supplementary material : Theoretical details

H.7.1 Background on Flow Matching and Flow Map models

In the case where the interpolant I_t admits the general form $\mathrm{I}_t = \alpha_t X_0 + \beta_t X_1$, a velocity field candidate for the generative ODE (H.1) is given by $v_t(x) = \mathbb{E}[v_{\mathrm{FM}}^{\mathrm{target}} | \mathrm{I}_t = x]$ where $v_{\mathrm{FM}}^{\mathrm{target}} = \dot{\mathrm{I}}_t = \dot{\alpha}_t X_0 + \dot{\beta}_t X_1$ is the conditional target velocity. Following Boffi et al. [2025a], the loss target in the SD optimization problem (H.9) related to this general SI writes as follows

1. *Lagrangian* SD (LSD) loss

$$u_{\mathrm{LSD}}^{\mathrm{target}} = u_{s,s}^{\theta}\left(\mathrm{I}_t - (t-s)u_{s,t}^{\theta}(\mathrm{I}_t)\right) + (t-s)\partial_s u_{s,t}^{\theta}(\mathrm{I}_t)\ , \tag{H.18}$$

2. *Eulerian* SD (ESD) loss

$$u_{\mathrm{ESD}}^{\mathrm{target}} = u_{t,t}^{\theta}(\mathrm{I}_t) - (t-s)\left(\nabla u_{s,t}^{\theta}(\mathrm{I}_t)u_{t,t}^{\theta}(\mathrm{I}_t) + \partial_t u_{s,t}^{\theta}(\mathrm{I}_t)\right)\ , \tag{H.19}$$

3. *Shortcut* SD (SSD) loss

$$u_{\mathrm{SSD}}^{\mathrm{target}} = \frac{t-r}{t-s}u_{r,t}^{\theta}(\mathrm{I}_t) + \frac{r-s}{t-s}u_{s,r}^{\theta}\left(\mathrm{I}_t - (t-r)u_{r,t}^{\theta}(\mathrm{I}_t)\right), \quad r \sim q(r|s,t)\ , \tag{H.20}$$

where $q(r|t,s)$ is a time distribution on $\{r \in [0,1] : s < r < t\}$. These expressions are derived from straightforward computations on the Flow Map (H.8), summarized below as an informal counterpart to the proofs in Boffi et al. [2025a].

Proof. **Lagrangian setting.** By computing the derivative of the Flow Map (H.8) evaluated in the trajectory $(X_t)_{t\in[0,T]}$ with respect to time variable s, we have

$$\frac{\mathrm{d}X_{s,t}(X_t)}{\mathrm{d}s} = u_{s,t}(X_t) - (t-s)\partial_s u_{s,t}(X_t)\ ,$$

By definition of the Flow Map given in (H.7), we also have that $\mathrm{d}X_{s,t}(X_t)/\mathrm{d}s = v_s(X_{s,t}(X_t))$. By combining these two results, we obtain the *Lagrangian characterization*

$$\begin{aligned} u_{s,t}(X_t) &= v_s(X_{s,t}(X_t)) + (t-s)\partial_s u_{s,t}(X_t) \\ &= u_{s,s}(X_{s,t}(X_t)) + (t-s)\partial_s u_{s,t}(X_t)\ . \end{aligned}$$

After replacing u by u^{θ} in the previous equality, we obtain the LSD target (H.18) from the right term.

Eulerian setting. By computing the derivative of the Flow Map (H.8) evaluated in the trajectory $(X_t)_{t\in[0,T]}$ with respect to time variable t, we have

$$\begin{aligned} \frac{\mathrm{d}X_{s,t}(X_t)}{\mathrm{d}t} &= \frac{\mathrm{d}X_t}{\mathrm{d}t} - u_{s,t}(X_t) - (t-s)\frac{\mathrm{d}u_{s,t}(X_t)}{\mathrm{d}t} \\ &= v_t(X_t) - u_{s,t}(X_t) - (t-s)(\nabla u_{s,t}(X_t)v_t(X_t) + \partial_t u_{s,t}(X_t))\ . \end{aligned}$$

By definition of the Flow Map given in (H.7), we also have that $\mathrm{d}X_{s,t}(X_t)/\mathrm{d}t = v_t(X_t) - v_t(X_t) = 0$. By combining these two results, we obtain the *Eulerian characterization*

$$\begin{aligned} u_{s,t}(X_t) &= v_t(X_t) - (t-s)(\nabla u_{s,t}(X_t) v_t(X_t) + \partial_t u_{s,t}(X_t)) \\ &= u_{t,t}(X_t) - (t-s)(\nabla u_{s,t}(X_t) u_{t,t}(X_t) + \partial_t u_{s,t}(X_t)) \ . \end{aligned}$$

After replacing u by u^θ in the previous equality, we obtain the ESD target (H.19) from the right term.

Shortcut setting. By definition of the Flow Map given in (H.7), we have for any time $r \in (s,t)$

$$X_{s,t}(X_t) = X_{s,r}(X_{r,t}(X_t)) \ .$$

When substituting the parametrization (H.8) in the previous equality, we obtain the *Shortcut characterization*

$$\begin{aligned} & X_t - (t-s)u_{s,t}(X_t) = \{X_t - (t-r)u_{r,t}(X_t)\} - (r-s)u_{s,r}(X_{r,t}(X_t)) \\ \iff & u_{s,t}(X_t) = \frac{t-r}{t-s} u_{r,t}(X_t) + \frac{r-s}{t-s} u_{s,r}(X_{r,t}(X_t)) \ . \end{aligned}$$

After replacing u by u^θ in the previous equality, we obtain the SSD target (H.20) from the right term. □

Simplifications with the standard interpolant. Now assume that we have $\mathrm{I}_t = (1-t)X_0 + tX_1$. To obtain the objective functions presented in Appendix H.3.2, we proceed to the following simplifications: (a) taking inspiration from the conditional trick in FM, we replace each occurrence of the *marginal* velocity field ($u^\theta_{s,s}$ in LSD loss, $u^\theta_{t,t}$ in ESD loss) by the *conditional* velocity field, exactly defined as $v^{\text{target}}_{\text{FM}} = X_1 - X_0$, which has the computational advantage to be constant over the whole interpolant, (b) following Frans et al. [2025], we deterministically set $q(r|s,t) = \delta_{(s+t)/2}(r)$ in SSD formulation.

H.7.2 Generalization of CFG to the Flow Map setting

To incorporate CFG into Flow Map models, we suggest to replace the target velocity field v_t by its CFG counterpart (H.5), where the class-unconditional velocity $v_t(\cdot|\emptyset)$ is legitimately substituted by $u_{t,t}(\cdot|\emptyset)$. This naturally leads to the CFG-based loss for the FM setting detailed in (H.13), which is unchanged in the general SI setting (recalling that we now have $v^{\text{target}}_{\text{FM}} = \dot{\mathrm{I}}_t$). With this paradigm, the SD targets defined in Appendix H.7.1 naturally turn into the following CFG-based targets

1. CFG-based *Lagrangian* SD (cfg-LSD) loss

$$\begin{aligned} u^{\text{target}}_{\text{cfg-LSD}} &= v^{\text{target}}_{\text{cfg-FM}} + (t-s)\partial_s u^\theta_{s,t}(\mathrm{I}_t|\mathbf{c}) \ , \\ v^{\text{target}}_{\text{cfg-FM}} &= w\, u^\theta_{s,s}\left(\mathrm{I}_t - (t-s)u^\theta_{s,t}(\mathrm{I}_t|\mathbf{c}) \mid \mathbf{c}\right) + (1-w)u^\theta_{s,s}\left(\mathrm{I}_t - (t-s)u^\theta_{s,t}(\mathrm{I}_t|\emptyset) \mid \emptyset\right) \ , \end{aligned}$$

2. CFG-based *Eulerian* SD (cfg-ESD) loss

$$\begin{aligned} u^{\text{target}}_{\text{cfg-ESD}} &= v^{\text{target}}_{\text{cfg-FM}} - (t-s)\left(\nabla u^\theta_{s,t}(\mathrm{I}_t|\mathbf{c}) v^{\text{target}}_{\text{cfg-FM}} + \partial_t u^\theta_{s,t}(\mathrm{I}_t|\mathbf{c})\right) \ , \\ v^{\text{target}}_{\text{cfg-FM}} &= w\, u^\theta_{t,t}(\mathrm{I}_t|\mathbf{c}) + (1-w)u^\theta_{t,t}(\mathrm{I}_t|\emptyset) \ , \end{aligned}$$

3. CFG-based *Shortcut* SD (cfg-SSD) loss

$$u^{\text{target}}_{\text{cfg-SSD}} = \frac{t-r}{t-s} u^\theta_{r,t}(\mathrm{I}_t|\mathbf{c}) + \frac{r-s}{t-s} u^\theta_{s,r}\left(\mathrm{I}_t - (t-r)u^\theta_{r,t}(\mathrm{I}_t|\mathbf{c}) \mid \mathbf{c}\right) \ , \ r \sim q(r|t,s) \ .$$

Simplifications with the standard interpolant. Now assume that we have $\mathrm{I}_t = (1-t)X_0 + tX_1$. To obtain the CFG-based SD objectives presented in Appendix H.3.3, we apply the same simplifications as in the unconditional setting: in particular, we replace each occurrence of the *class-conditional marginal* velocity field in LSD and ESD settings by its conditional counterpart $v^{\text{target}}_{\text{FM}} = X_1 - X_0$, thus avoiding extra model evaluations, and set $q(r|s,t) = \delta_{(s+t)/2}(r)$ in SSD setting.

H.8 Supplementary material : Implementation details

H.8.1 General settings

Data. We assemble a collection of publicly available, free-to-use images to construct the LR–HR training dataset $\mathcal{D}$. To generate synthetic LR images from HR ground truth, we adapt the widely used degradation pipeline of Wang et al. [2021], applying the following operations in sequence:

1. *Moderate blurring*: applied with 80% probability;
2. *Downscaling* (core operation): applied systematically using a random scale $s_{\text{down}} \sim \mathrm{U}(0.1, 1)$ and a randomly chosen interpolation mode (nearest, area, bilinear, or bicubic);
3. *Addition of low-level noise*: Gaussian noise with 50% probability, otherwise Poisson noise;
4. *JPEG compression*: applied systematically with the default compression level from Wang et al. [2021].

To match the spatial resolution of the HR image and define a valid interpolant for the Flow Map formulation, the degraded output is resized back using a random interpolation mode, followed by additional blurring and pixel-value clamping. Both HR and LR images are then rescaled to the range $[-1, 1]$. Since we operate in latent space to leverage large pre-trained diffusion models, both HR and synthetically generated LR images are embedded using the pretrained variational encoder from [Podell et al., 2024].

To support training across multiple resolutions, we adopt a bucketing strategy [Podell et al., 2024] that accommodates varying aspect ratios and image sizes. The specific configuration depends on the scalability of each Flow Map variant:

• *SSD setting*: pixel budgets of $[512^2, 1024^2]$ sampled with probabilities $[0.2, 0.8]$ and batch sizes $[4, 2]$, respectively, to fit within a single H100 GPU; aspect ratios are sampled in $[1/4, 4]$.

• *LSD and ESD settings*: pixel budgets of $[256^2, 512^2]$ with the same sampling probabilities and batch sizes as in the SDD setting. Higher resolutions could not be accommodated due to memory constraints during training, which could not be alleviated by multi-GPU sharding owing to software limitations imposed by the use of the jvp package. Aspect ratios are again sampled in $[1/4, 4]$.

Architectures. In the FlowMapSR framework described in Appendix H.4, we employ three neural network components:
(A) *Flow Map model* $u^{\theta}_{s,t}$. This component is implemented as a UNet with an architecture inspired by the SDXL text-to-image diffusion model [Podell et al., 2024]. Conditioning is handled via projection-based embeddings. Specifically, to evaluate the model at a timestep pair (s, t), we apply positional encodings to each input, using 256 channels per timestep by default. In addition, conditioning on positive or negative text prompts is achieved through a frozen CLIP text encoder, producing embeddings of dimension 1280. Overall, the resulting model contains approximately 2.5B parameters.

(B) *Dynamic weighting function* $\lambda^{\psi}_{s,t}$. This component is implemented as a lightweight neural network. It consists of positional embeddings of s and t (256 channels each), whose outputs are concatenated into a 512-dimensional vector and passed through a two-layer MLP ($512 \to 1$ with SiLU activation, followed by $1 \to 1$). The final layer is initialized to zero using Kaiming uniform initialization, ensuring that the initial weighting is constant and equal to 1 across all timestep pairs. As a result, this network is extremely compact, with fewer than 1k parameters.

(C) *Discriminator* $\mathbf{D}^{\phi}$. It follows a patch-based architecture in the style of Pix2Pix [Isola et al., 2017]. It comprises four Conv2d layers with 64, 128, 256, and 512 channels, respectively. Group normalization is applied after each layer except the first, followed by a SiLU activation. A final Conv2d layer with a single output channel produces a patch-wise output of spatial size $(w/8 - 2, h/8 - 2)$ for an input of size (w, h). The discriminator contains approximately 2.8M parameters.

H.8.2 Training procedure

Initialization of unconditional FlowMapSR. For unconditional models, we initialize the network using the Latent Bridge Matching (LBM) image deblurring checkpoint from Chadebec et al. [2025b], which provides a strong initialization for image restoration. Although the LBM interpolant is defined as a Brownian bridge—i.e., a *stochastic* variant of the standard FM interpolant with added Gaussian noise scaled by $\sigma_t = \sigma\sqrt{t(1-t)}$—this initialization remains appropriate in our setting, since the noise level is negligible ($\sigma = 5 \times 10^{-3}$). As the

LBM model is unconditional, we load only the shared parameters, leaving class-embedding weights randomly initialized.

Initialization of CFG-based FlowMapSR. CFG-based models are initialized from their corresponding unconditional FlowMapSR counterparts, previously trained without guidance. As before, only shared parameters are loaded, while class-embedding weights are initialized randomly.

Time-dependent regularization with perceptual loss. We set the time-dependent perceptual weight as $\lambda_s^{\text{LPIPS}} = 5 \times \exp(-4s)$, which we found to balance effective pixel-level regularization with minimal bias toward the FM-SD objective. To reduce memory usage, we follow the random cropping strategy of Chadebec et al. [2025b], computing the LPIPS loss on randomly sampled patches from X_0 and $\hat{X}_0$. In all experiments, the maximum patch size is fixed to 512^2.

Discrete timestep sampling. Let $\{t_k\}_{k=0}^{K}$ denote a discretization of $[0, 1]$ into K uniform sub-intervals. We set $d_{\max} = 7$, corresponding to a finest discretization of $2^7 + 1 = 129$ timesteps. The joint sampling distribution $\bar{q}(s, t)$ is defined as follows, depending on the loss type:

- *FM loss.* With 75% probability, we set $d = d_{\max}$, sample k uniformly from $\{0, \dots, K_d\}$, and take $s = t = t_k$.
- *SD loss (LSD/ESD).* With 25% probability, we sample d uniformly from $\{0, \dots, d_{\max}\}$, then sample k uniformly from $\{1, \dots, K_d\}$ and k' uniformly from $\{0, \dots, k\}$, setting $(s, t) = (t_{k'}, t_k)$.
- *SD loss (SSD).* With 25% probability, we sample d uniformly from $\{0, \dots, d_{\max} - 1\}$, sample k uniformly from $\{1, \dots, K_d\}$, and set $(s, t) = (t_{k-1}, t_k)$. While more restrictive than the LSD/ESD setting, this choice simplifies the definition of the midpoint $r = (s + t)/2$ and matches the protocol of Frans et al. [2025].

Positive–negative prompting guidance & LoRA-based adversarial fine-tuning. Following standard CFG practice, we sample the positive prompt $\mathbf{c}^{\text{pos}}$ with 90% probability and the negative prompt $\mathbf{c}^{\text{neg}}$ with 10% probability. We set $\lambda^{\text{adv}} = 0.1$ and fix the LoRA rank to 64 across all experiments.

Optimization details. Training instances of FlowMapSR begins with a pure FM warm-start phase of 5k iterations, followed by 5k iterations of unconditional self-distillation and an additional 3k iterations with CFG. Adversarial fine-tuning then proceeds for 2.5k to 4k iterations, depending on RPGAN convergence. The effective batch size is fixed to 256 using gradient accumulation. We use AdamW with default hyperparameters, setting the learning rate to 4×10^{-5} for the Flow Map model and 10^{-4} for the discriminator. Since the initial LBM checkpoint is already well aligned with our framework, we do not employ EMA, which we found to degrade performance.

H.8.3 Unsuccessful enhancement strategies

In preliminary experiments, we explored several alternative strategies to further enhance SR performance. However, none of the approaches described below led to measurable improvements over our final configuration.
(a) *Conditioning on the upscaling factor s_{up}.* We investigated explicitly conditioning the Flow Map model on the target upscaling factor using positional encodings with appropriate scaling, but it did not yield any noticeable performance gains.

(b) *Fourier coefficient matching loss.* Inspired by [Wu et al., 2025b], we evaluated the use of Fourier coefficient matching as an additional pixel-level regularization term, using the same coefficient settings as for the LPIPS loss. In our experiments, this approach did not improve output sharpness or visual quality.

(c) *Pixel-space RPGAN fine-tuning.* To exhaustively explore adversarial training options, we also experimented with aligning pixel-space distributions after CFG training. To mitigate memory constraints, we adopted the same random cropping strategy as for LPIPS regularization, evaluating the discriminator on randomly sampled patches from $\hat{X}_0^{\text{adv}}$ and X_0, with a maximum patch size of 512^2. This approach consistently produced inferior results compared to our default latent-space adversarial fine-tuning. Representative qualitative examples are shown in Figure H.16.

H.9 Supplementary material : Additional results

Additional comparison with competing methods. In this section, we complement the main results of Appendix H.5. To demonstrate the consistency of FlowMapSR across upscaling factors, we report quantitative results on DIV2K-Val for $\times 2$ SR in Table H.6, with qualitative examples in Figure H.9. These results align with those observed for $\times 4$ and $\times 8$ upscaling in Appendix H.5. We further present qualitative comparisons for $\times 4$ upscaling on DIV2K-Val (Figure H.10) and RealSR (Figure H.11), as well as for $\times 8$ upscaling on DIV2K-Val (Figure H.12) and RealSR (Figure H.13). Real-world LR inputs from the RealSet65 dataset are also included in Figure H.14 for both $\times 4$ and $\times 8$ SR. Across all settings, FlowMapSR consistently recovers fine-grained structures aligned with HR references, produces lifelike textures, particularly for human and animal details (e.g., rows 5–8 in Figure H.10, rows 3–4 in Figure H.14, and rows 2, 3, and 7 in Figure H.12), and avoids visually disruptive artifacts. Moreover, FlowMapSR more faithfully preserves realistic depth of field, where competing methods often fail (e.g., row 7 in Figure H.10, rows 2 and 7 in Figure H.11, and row 8 in Figure H.12).

Ablation studies. We further provide complementary qualitative analyses to illustrate the effect of key design choices in FlowMapSR. In Figure H.7, we compare *deterministic* CFG training with the *stochastic* CFG strategy proposed in this work, observing reduced sharpness and increased artifacts under deterministic guidance. In Figure H.15, we report additional examples for the *Lagrangian* and *Eulerian* FlowMapSR variants, with and without CFG: the *Lagrangian* formulation deteriorates rapidly as CFG strength increases, while the *Eulerian* variant remains more stable but struggles to produce visually compelling results. In Figure H.16, we analyze the impact of adversarial fine-tuning in the *Shortcut* formulation, comparing latent-space and pixel-space RPGAN approaches under both unconditional and CFG-based training. Latent-space adversarial fine-tuning consistently yields superior visual quality, and its benefits are significantly amplified when combined with CFG. These observations motivate our final design choice, where adversarial fine-tuning is applied jointly with CFG rather than in isolation. Finally, we provide additional qualitative examples for the ablation studies on the number of inference steps (Figure H.17) and the LoRA scale (Figure H.18), which further confirm the trends discussed in Appendix H.5.

Figure H.7: **Qualitative comparison of CFG-enhanced Shortcut variants of FlowMapSR for $\times 4$ upscaling (deterministic vs stochastic).** We compare several deterministic choices of the guidance scale w with the default stochastic CFG configuration ($w_{\max} = 3.5$). Adversarial fine-tuning is not applied in these comparisons. The LR images are taken from the Div2K-Val dataset.

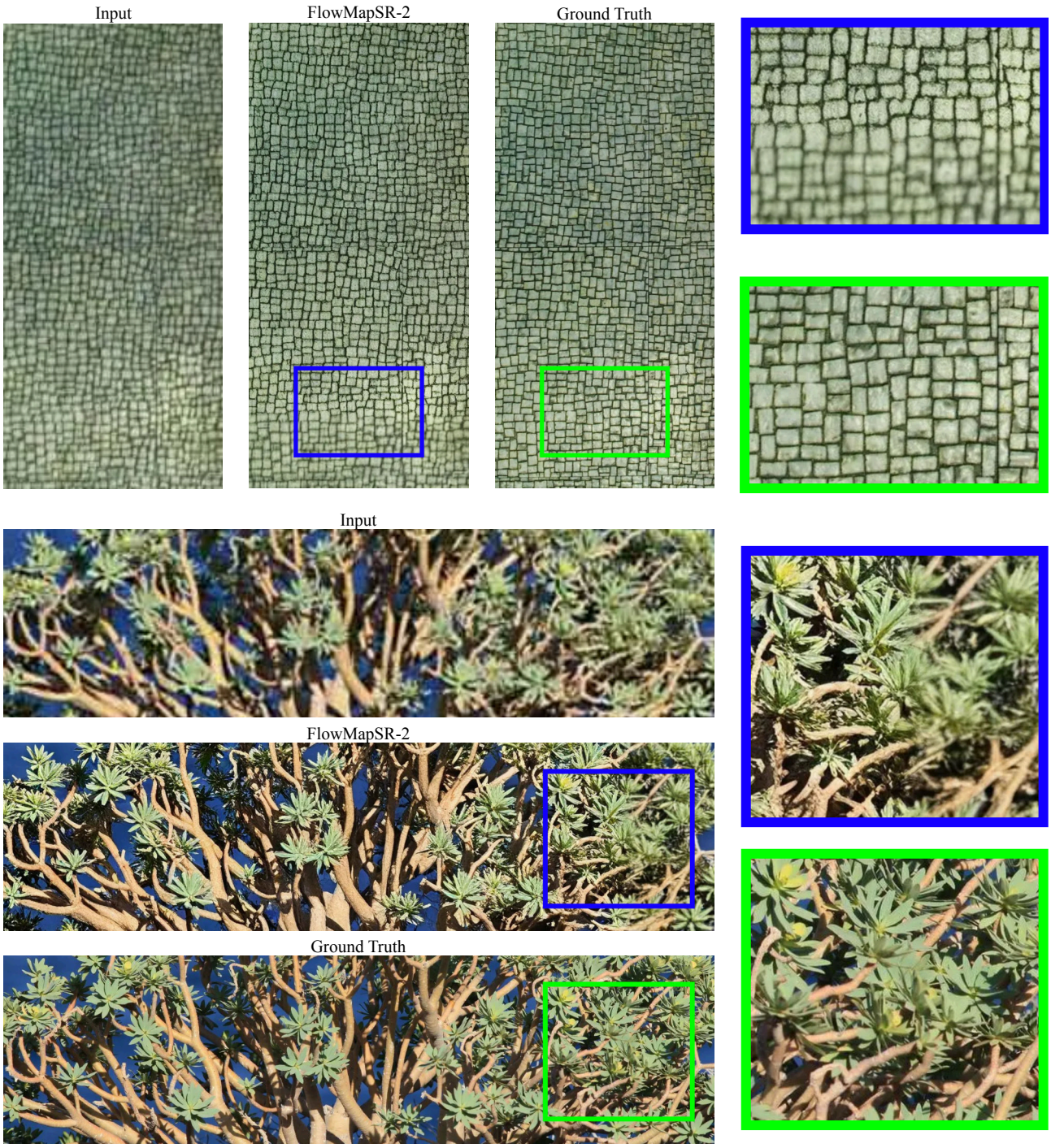


Figure H.8: **Examples of tiling failure in FlowMapSR for $\times 4$ upscaling.** The LR inputs are drawn from the DIV2K-Val dataset. For HR images, we display the same zoomed region (blue for FlowMapSR, green for ground truth).

Visual limitation of FlowMapSR. We observe in some cases that the Gaussian tiling strategy used in FlowMapSR to generate HR images with resolution exceeding 1024×1024 introduces visible artefacts, manifested as blurry rectangular regions near image boundaries, see Figure H.8. We conjecture that this limitation may contribute to the inferior performance observed on non-reference evaluation metrics in Table H.2, as such artefacts degrade image sharpness and perceptual realism. Addressing and improving the tiling procedure is left for future work.

Table H.6: **Quantitative comparison between FlowMapSR and competing SR methods (×2 upscaling).** The best and second best results are highlighted in **bold** and underlined, respectively.

Dataset (s_{up})	Method	Reference metrics					Non-reference metrics			
		PSNR↑	SSIM↑	LPIPS↓	DISTS↓	FID↓	NIQE↓	MANIQA↑	MUSIQ↑	CLIPIQA↑
DIV2K-Val (×2)	Real-ESRGAN	**25.16**	**0.7829**	0.2123	0.1050	14.75	3.855	0.4122	0.6450	0.5515
	OSEDiff-1	22.72	0.6931	0.2363	0.1063	17.78	3.386	**0.4533**	**0.6894**	**0.6183**
	S3Diff-1	22.39	0.6748	0.2316	0.1034	10.97	3.715	0.3700	0.6175	0.5045
	FlowMapSR-2	23.82	0.6989	**0.1999**	**0.0770**	**7.16**	**2.904**	0.3892	0.6390	0.5356

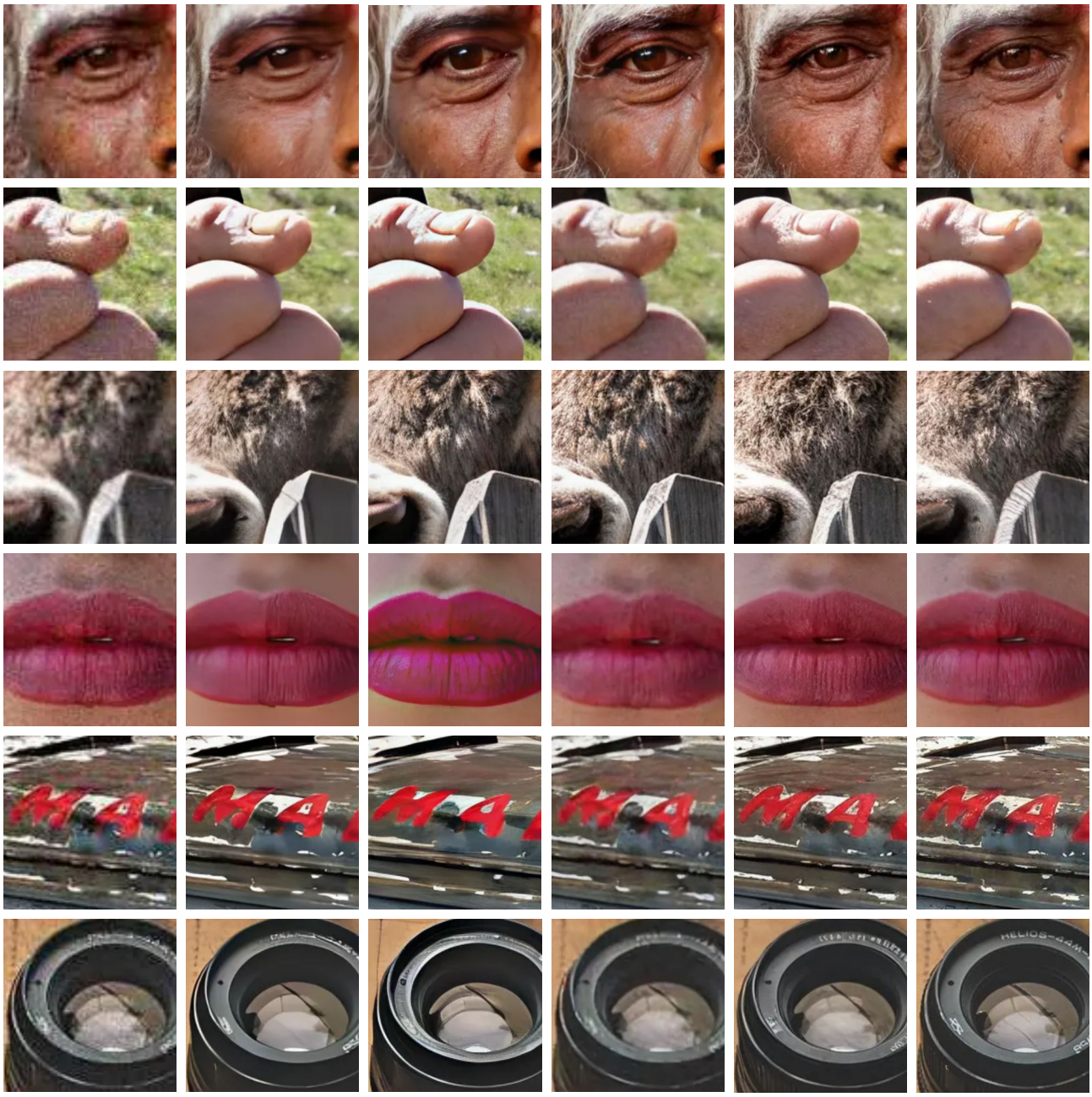


Figure H.9: **Qualitative comparison between FlowMapSR and competing SR methods for ×2 upscaling on DIV2K-Val.** Visually, FlowMapSR more faithfully recovers fine details, textures and depth of field while maintaining high overall image quality with no noticeable artifacts.

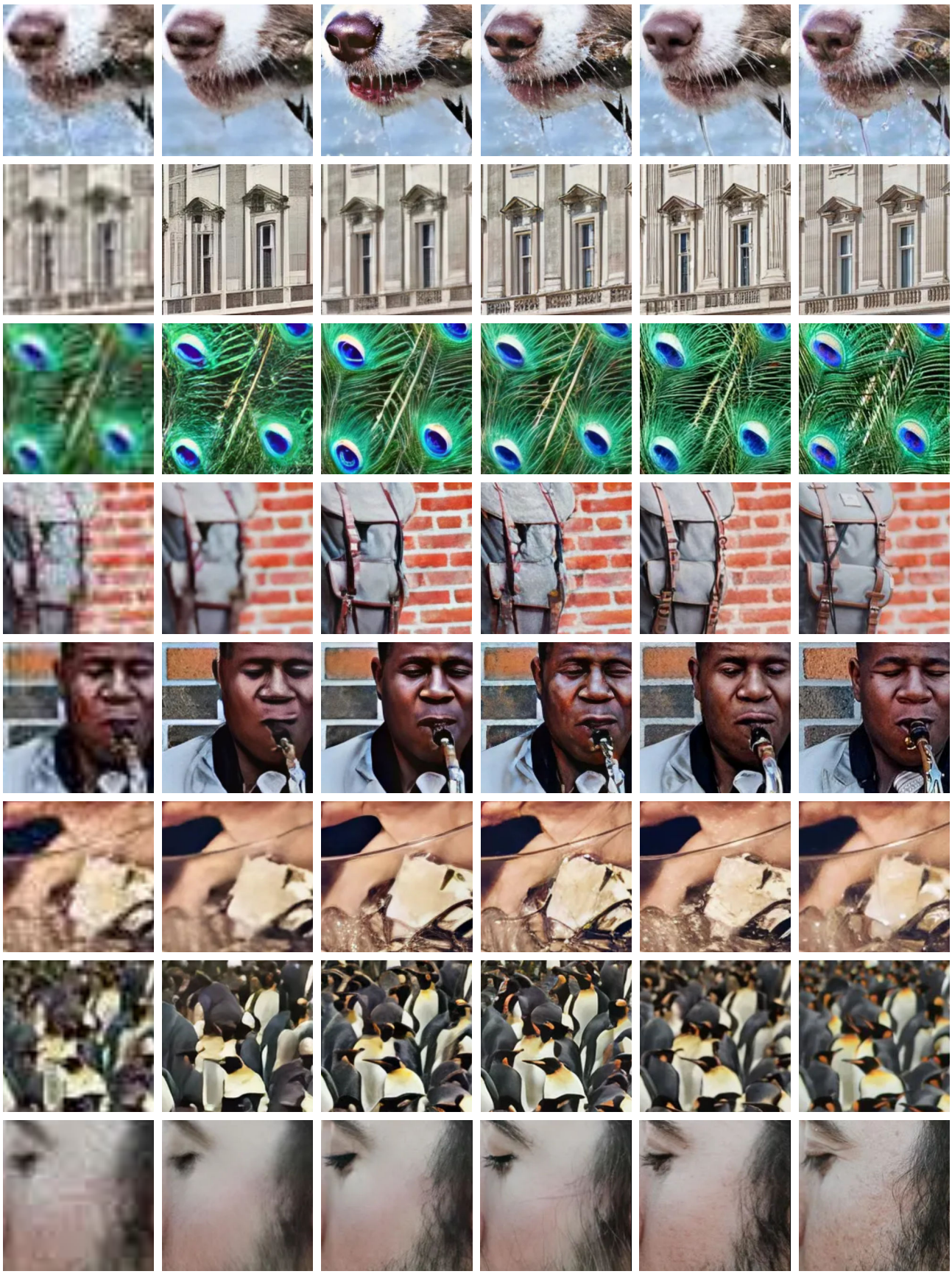


Figure H.10: **Qualitative comparison between FlowMapSR and competing SR methods for ×4 upscaling on DIV2K-Val.** This is complementary to Figure H.3.

Figure H.11: **Qualitative comparison between FlowMapSR and competing SR methods for ×4 upscaling on RealSR.** This is complementary to Figure H.3.

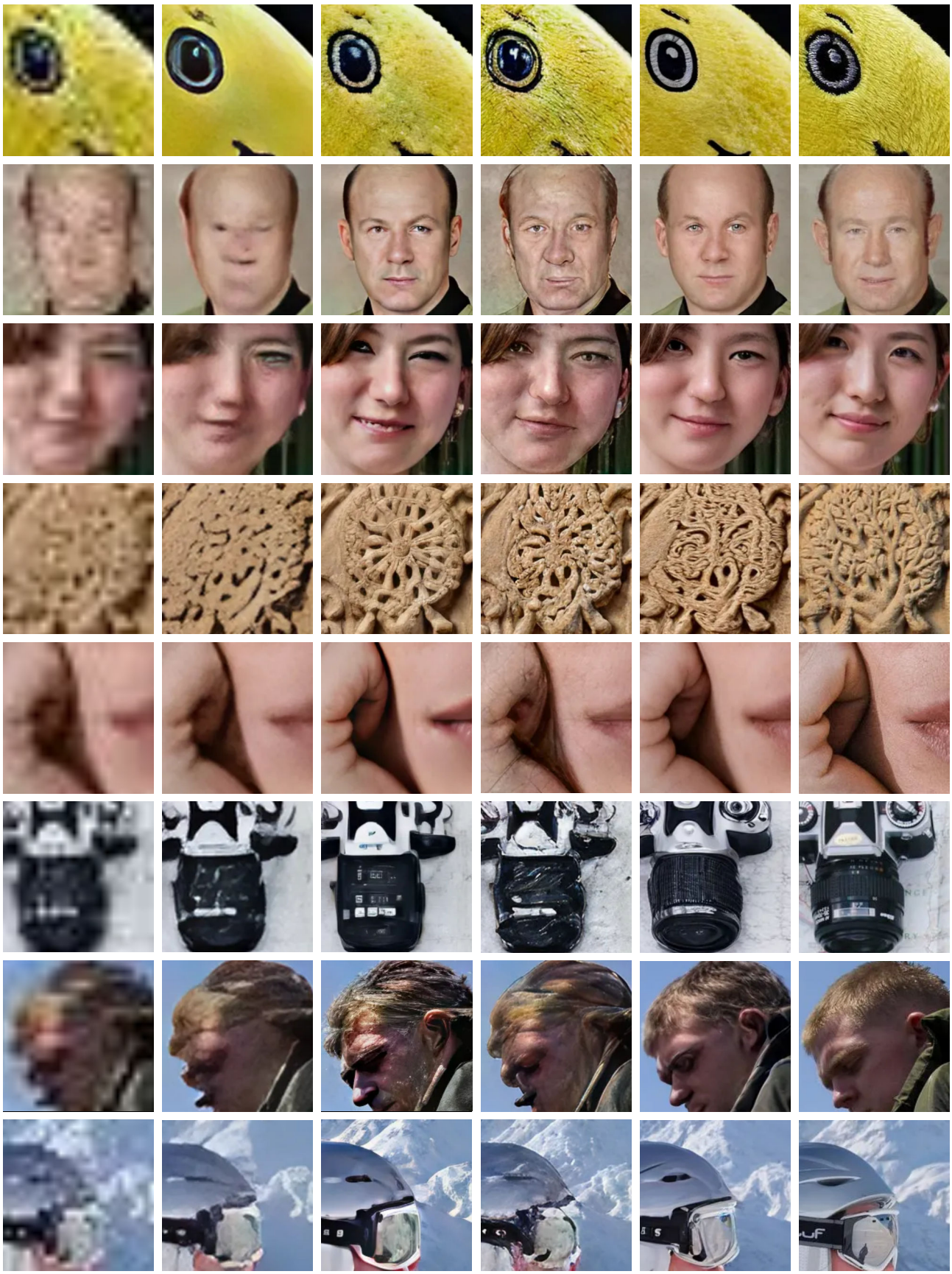


Figure H.12: **Qualitative comparison between FlowMapSR and competing SR methods for ×8 upscaling on DIV2K-Val.** This is complementary to Figure H.3.

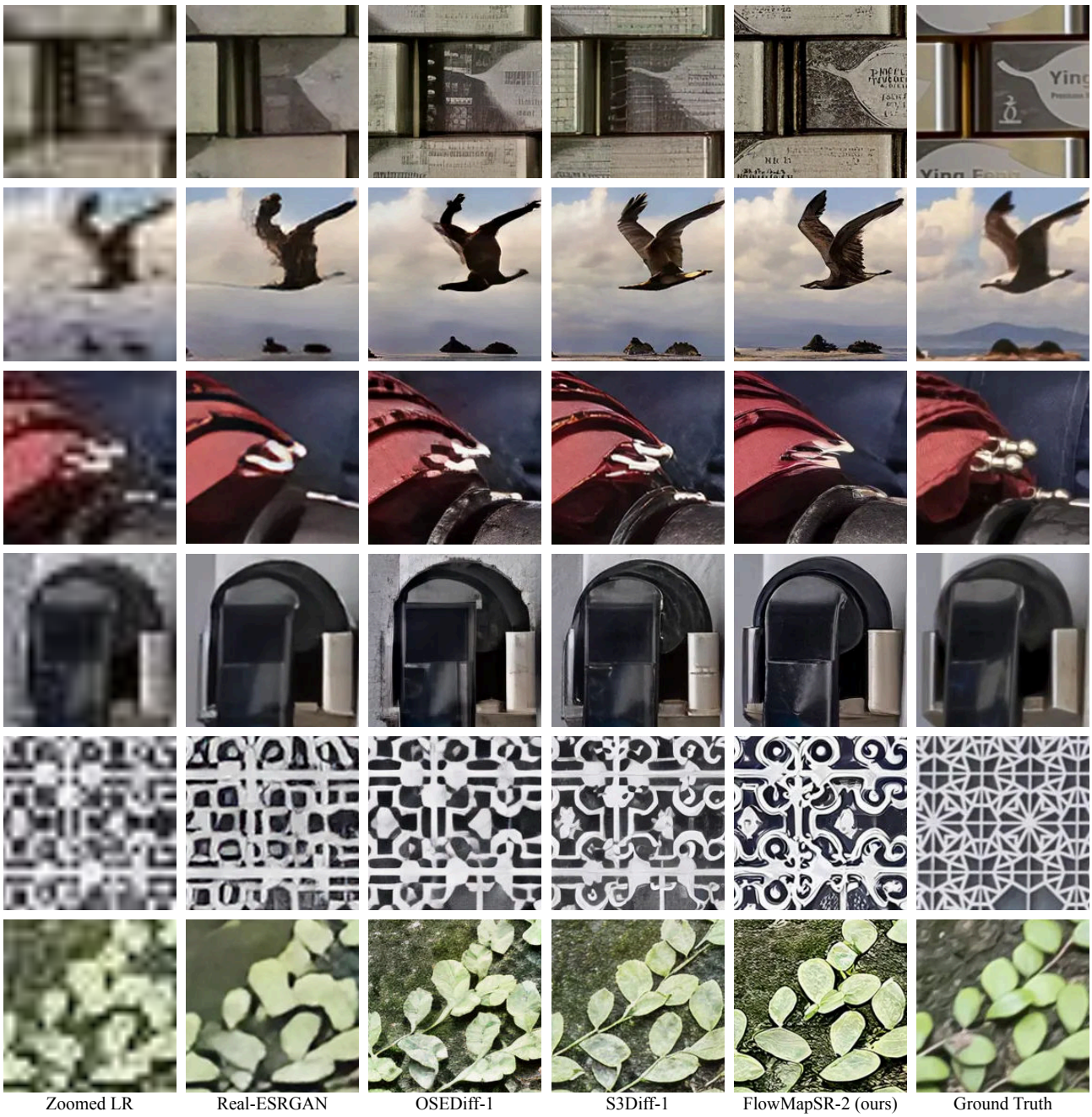


Figure H.13: **Qualitative comparison between FlowMapSR and competing SR methods for ×8 upscaling on RealSR.** This is complementary to Figure H.3.

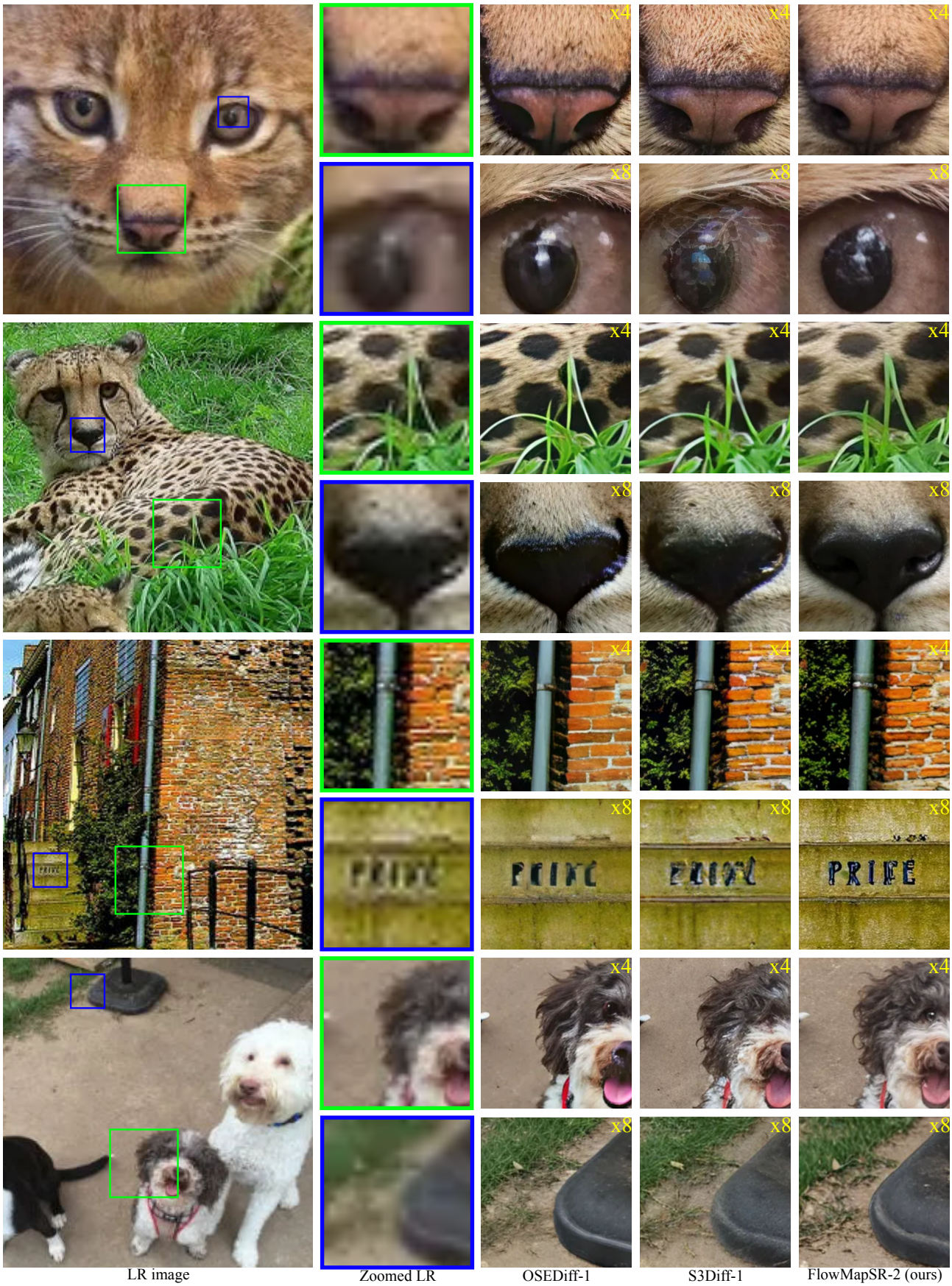


Figure H.14: **Qualitative comparison between FlowMapSR and competing diffusion-based SR methods on real-world LR inputs for $\times 4$ and $\times 8$ upscaling.** The LR images are taken from the RealSet65 dataset. This is complementary to Figure H.4.

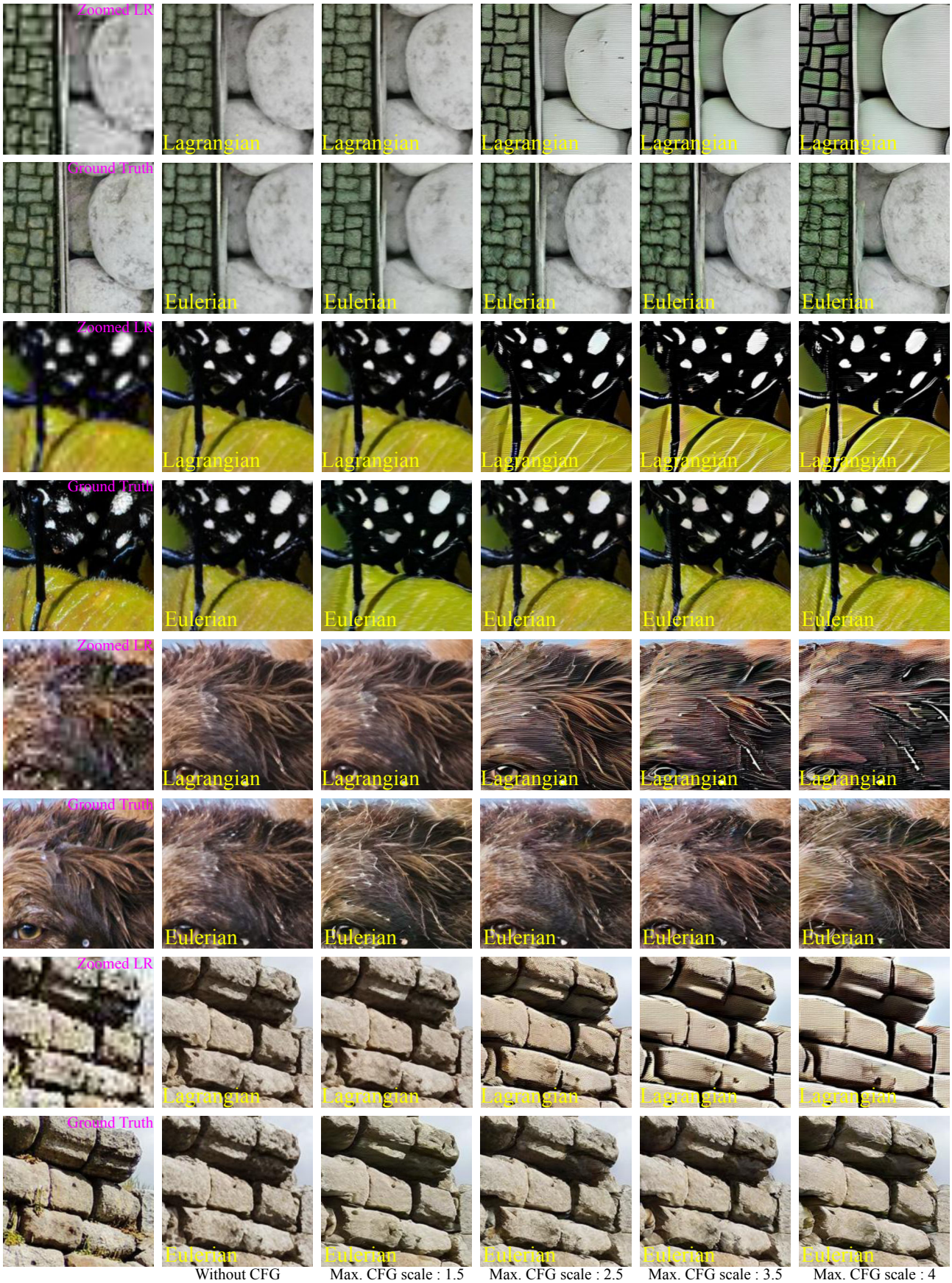


Figure H.15: **Qualitative comparison of positive-negative CFG enhancement applied to LSD- and ESD-based FlowMapSR for $\times 4$ and $\times 8$ upscaling.** The LR images are taken from the Div2K-Val dataset. This is complementary to Figure H.2.

Figure H.16: **Qualitative comparison of adversarially finetuned Shortcut variants of FlowMapSR for $\times 4$ upscaling.** The green setting ($w_{\max} = 3.5$ & latent RPGAN) is the default configuration used in our main experiments. The LR images are taken from the Div2K-Val dataset.

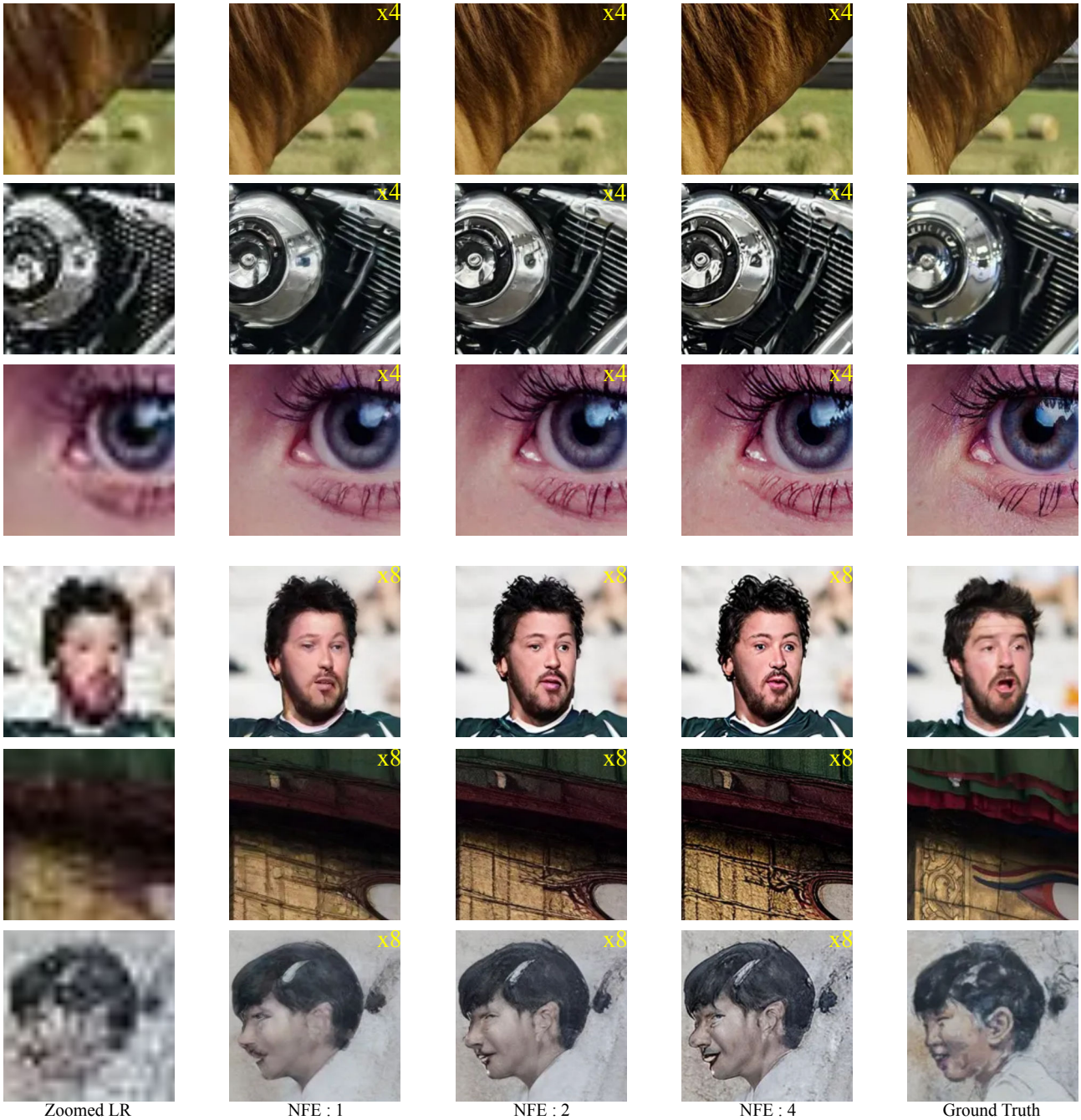


Figure H.17: **Qualitative comparison of FlowMapSR with varying numbers of inference steps (NFE) for $\times 4$ (rows 1–3) and $\times 8$ (rows 4–6) upscaling.** The LR inputs are drawn from the DIV2K-Val dataset. This is complementary to Figure H.5.

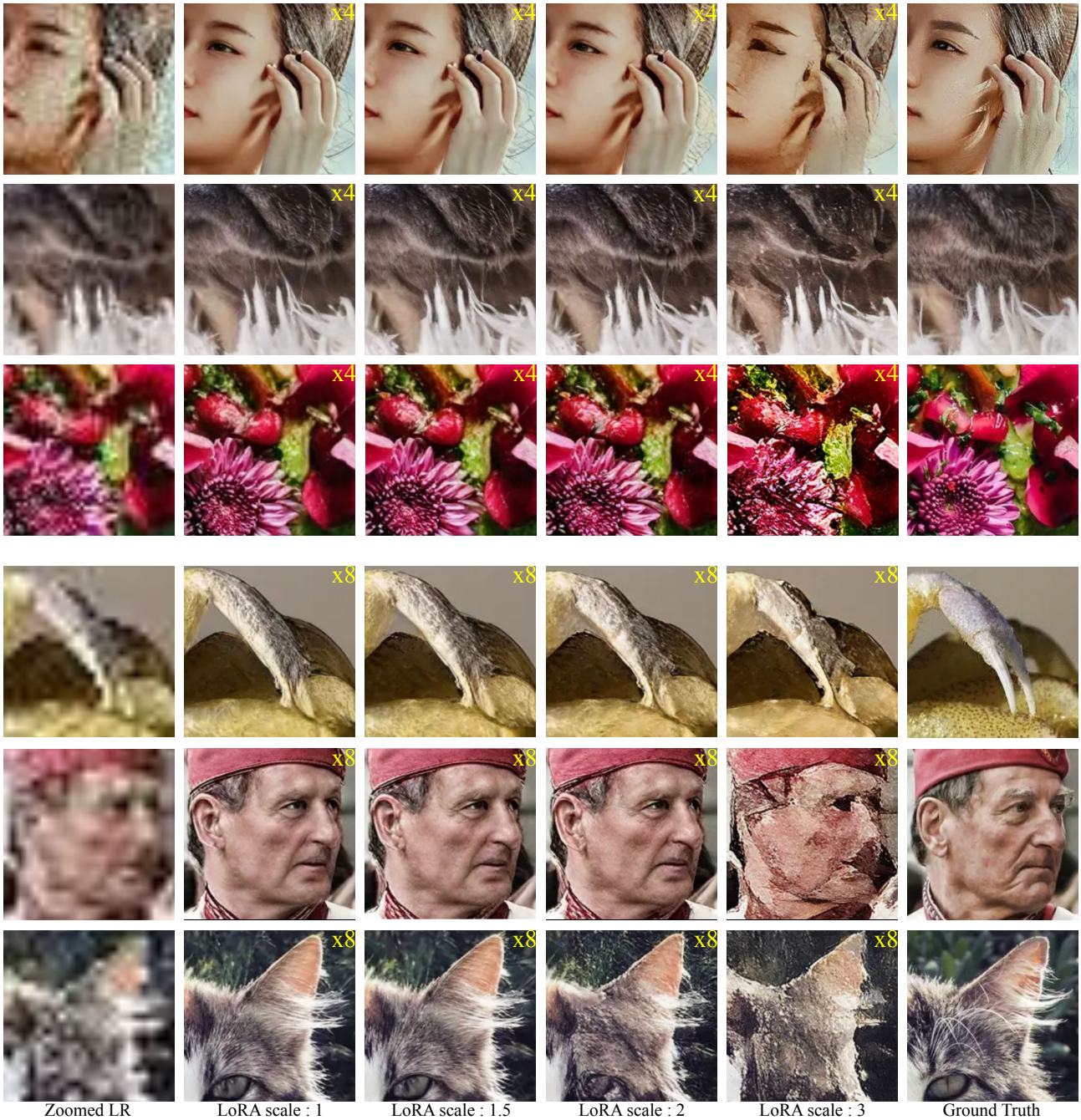


Figure H.18: **Qualitative comparison of FlowMapSR with varying LoRA scale for $\times 4$ (rows 1–3) and $\times 8$ (rows 4–6) upscaling.** The LR inputs are drawn from the DIV2K-Val dataset. This is complementary to Figure H.6.

ECOLE
DOCTORALE
DE MATHEMATIQUES
HADAMARD

Titre : Echantillonnage et Transport par modèles de diffusion basés sur le score : opportunités et avancées méthodologiques

Mots clés : Echantillonnage, transport optimal, modèles de diffusion, méthodes de Monte Carlo par chaîne de Markov, inférence variationnelle, apprentissage automatique

Résumé : Cette thèse explore la conception de nouvelles méthodologies pour l'échantillonnage et le transport optimal, en s'appuyant principalement sur des modèles génératifs récemment développés, appelés modèles de diffusion basés sur le score. La première partie aborde deux défis fondamentaux liés à l'échantillonnage. Nous examinons tout d'abord les espaces d'états soumis à des contraintes géométriques, pour lesquels nous proposons une nouvelle méthode non biaisée basée sur l'Hamiltonien Monte Carlo riemannien. Nous nous penchons ensuite sur le défi de la multimodalité, en soulignant les limites structurelles communes à de nombreuses approches existantes, y compris les méthodes Monte Carlo classiques. Pour surmonter la difficulté de la multimodalité, la deuxième partie introduit les modèles de diffusion dans des schémas d'échantillonnage et développe trois méthodologies complémentaires : un échantillonneur basé sur la diffusion sans apprentissage qui repose uniquement sur des estimateurs Monte Carlo ; un échantillonneur basé sur la diffusion par inférence variationnelle recuite qui utilise des indications informatives pour atténuer l'effondrement des modes ; et une nouvelle classe de générateurs de Boltzmann combinant des modèles de diffusion avec des techniques Monte Carlo recuites. Dans la dernière partie, nous exploitons les modèles de diffusion dans le cadre du pont de Schrödinger pour traiter des problèmes avancés de transport optimal. Tout d'abord, nous introduisons une méthode pour résoudre des problèmes de transport optimal multi-marginal avec des fonctions de coût structurées en arbre, s'appliquant principalement aux barycentres de Wasserstein. Ensuite, nous développons une approche de correspondance de ponts non biaisée pour résoudre numériquement des problèmes généralisés de pont de Schrödinger, motivés par l'inférence de la dynamique de population à partir de données longitudinales instantanées.

Title : Sampling and Transport with Score-based Diffusion Models : Opportunities and Methodological Advances

Keywords : Sampling, optimal transport, diffusion models, Markov chain Monte Carlo methods, variational inference, machine learning

Abstract : This thesis investigates the design of new methodologies for sampling and optimal transport, primarily leveraging recently developed generative models known as score-based diffusion models. The first part addresses two fundamental challenges in sampling. We first consider geometrically constrained state spaces, for which we propose a novel Riemannian Hamiltonian Monte Carlo-based method that is proven to be unbiased. We then turn to the challenge of multi-modality, highlighting structural limitations shared by broad classes of existing approaches, including classical Monte Carlo methods. To tackle multi-modality difficulty, the second part introduces diffusion models into sampling schemes and develops three complementary methodologies: a learning-free diffusion-based sampler relying solely on Monte Carlo estimators; an annealed variational inference diffusion-based sampler that uses informative guidance to mitigate mode collapse; and a new class of Boltzmann Generators combining diffusion models with annealed Monte Carlo techniques. In the final part, we leverage diffusion models within the Schrödinger Bridge framework to address advanced optimal transport problems. We first introduce a method for solving multi-marginal optimal transport problems with tree-structured cost functions, with applications to Wasserstein barycenters. Then we develop an unbiased bridge-matching approach for numerically solving generalized Schrödinger Bridge problems motivated by the inference of population dynamics from snapshot longitudinal data.

Institut Polytechnique de Paris
91120 Palaiseau, France