Generative models for ECG data : theory and application.

Gabriel Victorino Cardoso

INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2024IPPAX022

Thèse de doctorat

Modèles génératifs pour le traitement des données du type électrocardiogramme : théorie et application.

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à Ecole Polytechnique

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

Thèse présentée et soutenue à Palaiseau, le 29/04/2024, par

GABRIEL VICTORINO CARDOSO

Composition du Jury :

Philippe Moireau Directeur de recherche, INRIA Saclay et Ecole polytechnique	Président
Florence Forbes Directrice de recherche, INRIA Grenoble Rhone-Alpes	Rapporteuse
Thomas Schön Professeur, Uppsala University	Rapporteur
Marcelo Pereyra Professeur, Heriot-Watt University et Maxwell Insitute for Mathematical Sciences	Examinateur
Eric Moulines Professeur, Ecole polytechnique	Directeur de thèse
Rémi Dubois Professeur, Université de Bordeaux et Institut Liryc	Co-directeur de thèse
Geneviève Robin Chercheuse, Owkin	Invitée
Jean-Michel Haïssaguerre Professeur, Centre Hospitalier Universitaire(CHU) de Bordeaux.	Invité

Remerciements

Je veux d'abord remercier mes directeurs de thèse Eric Moulines et Rémi Dubois pour ces trois années de collaboration. Vous avez tous les deux contribué humainement et scientifiquement à ma thèse et a mon développement personnel pendant ces trois derniers années et je suis reconnaissant d'avoir eu la chance de travailler avec vous. Eric est pour moi l'équivalent de la fusion nucléaire car il "dégage une quantité d'énergie colossale par unité de masse, provenant de l'attraction entre les nucléons due à l'interaction forte". C'est impressionnant de le voir évoluer entre tous ces projets et la façon dont il arrive toujours à proposer des nouvelles pistes pour des projets et une fois ces projets lancés, de trouver les bonnes questions. En ce qui concerne des questions importantes de science, j'ai rarement vu quelqu'un d'aussi passionné par un projet que Rémi. Rémi possède non seulement une connaissance très profonde et vaste sur tout ce qui concerne la cardiologie, mais il a surtout l'envie d'apporter des solutions pertinentes à des problèmes concrets auxquels font face les cardiologues. Il sait très bien flairer les "bullshits" et discerner les pistes pertinentes. Il m'a apporté, même si je n'ai pas pu passer autant de temps à Bordeaux que j'aurais voulu pendant ma thèse, cette idée d'accorder plus d'importance au problème qu'on veut résoudre qu'au nombre de publications qui pourront en découler.

Je ne peux pas conclure la partie sur mes directeurs de thèse sans parler de Geneviève Robin et Michel Haïssaguerre. Geneviève a été vraiment quelqu'un de très important pour le début de ma thèse et qui m'a quelque part appris comment je devais la naviguer. J'ai de très bons souvenirs de ces 8 premiers mois de thèse, où j'ai interagi principalement avec Geneviève, et de ce séjour au CIRM. C'était toujours un plaisir d'aller discuter avec elle sur des questions scientifiques et autres, mon seul regret étant que nous n'ayons pas pu continuer cette collaboration plus longtemps car je suis sûr que j'aurais beaucoup appris sur le plan scientifique et humain. Michel Haïssaguerre est un nom que je connaissais déjà depuis mon expérience en tant qu'ingénieur dans le milieu de la cardiologie. C'est tout simplement une "star", dont les contributions scientifiques nous laissent tout simplement émerveillés. La façon passionnée avec laquelle Michel porte le projet HELP est très inspirante. Humainement, sa gentillesse et sa bienveillance ne sont que comparables à sa carrière en tant que médecin. Quand je discutais avec Michel, parfois j'étais embarrassé à chaque fois qu'il portait un regard admiratif sur le genre de connaissance que nous les ingénieurs /matheux pouvons avoir, lui qui a eu un impact fondamental sur la vie de milliards de personnes dans la planète.

Je souhaite remercier Florence Forbes et Thomas Schön d'avoir accepté de rapporter ma thèse, l'attention que vous avez porté a mes travaux est un honneur pour moi. Thank you, Pr. Schön, for accepting to report my thesis. Je remercie également Marcelo Pereyra et Philippe Moireau d'avoir accepté de faire partie du jury.

Je remercie aussi tous mes co-auteurs pour tous les très bons moments de partage scientifique passés ensemble : Inass Sekkat, Gabriel Stoltz, Tony Lelièvre, Achille Thin, Jimmy Olsson, Andony Arrieula, Mark Potse, Josselin Duchateau, Lisa Bedin, Julia Linhart, Alexandre Gramfort et Pedro L.C.Rodriguez. Non, je n'ai pas oublié le duo Janati-Le Corff ! Je commence par Sylvain. Je le remercie d'abord pour la générosité avec laquelle il m'a accueilli au LPSM. Sylvain est arrivé à un moment clé de ma thèse et m'a permis de mettre beaucoup de choses en perspective. Que ça soit sur les maths, le Machine Learning ou

la recherche en générale. Sylvain peut souvent dire qu'il est lent et qu'il ne comprend pas tout et je n'ai jamais su si c'était par modestie ou si vraiment il ne se rend pas compte de qui il est. C'est quelqu'un d'extrêmement brillant, généreux et avec une connaissance aussi vaste sur des sujets mathématiques différents, digne de son directeur de thèse.

Bon, si vous êtes ici c'est peut-être car vous me connaissez donc vous savez sûrement à quel point Dr. Janati est important pour moi. Pour ceux qui ne me connaissent pas, je vous explique. Je ne supporte pas Daft Punk. Je ne supporte pas une salle sans fenêtre avec des lumières blanches. Pourtant, un de mes meilleurs souvenirs de thèse a été d'être avec Yazid en train de bosser dans cette affreuse salle sans fenêtre à Jussieu. Je me souviens qu'on avait nos capuches tellement on avait mal à la tête d'avoir passé 10 heures dans cette salle avec ce monsieur qui nous racontait pour la 100000e fois dans la journée qu'il allait dans des "Discothèques" et qu'il s'appelait Giorgio. Pourtant c'est là que j'ai peut-être le plus appris. Scientifiquement, Yazid m'a rappelé à quel point c'est important d'avoir une compréhension profonde sur les problèmes qu'on essaie d'aborder. Lui et son plus grand allié, le mélange gaussien, sont toujours là pour nous éclairer sur des sujets que vous pensez connaître. Je le remercie énormément d'avoir voulu travailler avec moi et je peux garantir que sans lui cette thèse n'aurait pas lieu. J'espère pouvoir continuer à suivre, en tant que collaborateur ou juste admirateur, ses aventures scientifiques et personnelles. Et non, il y a toujours le "frown" dans notre papier, mais c'est bien de penser qu'on va réussir à l'enlever un jour. Et oui, c'était sympa Cambridge, voilà c'est dit. J'en profite aussi pour remercier Soukaina, dont la bonne humeur a toujours été au rendez-vous et nous a souvent aidé à rigoler des situations pas toujours très marrantes. Et j'ai entendu dire que vous embauchez souvent des super musiciens pour vos soirées, continuez comme ça.

Bien sûr, ces trois années auraient été beaucoup moins bien sans les co-doctorants. Je remercie tous les doctorants du CMAP/Eric Moulines et affiliés : Louis (la personne avec qui j'ai le plus mangé des noix de ma vie), Tom (on joue au foot ensemble), Achille, Maxence, Mehdi, Pablo, Pierre, Valentin, Vincent, Lisa, Mareike. Je remercie les doctorants/ingénieurs du Liryc a Bordeaux: Mariette, Georges et Nicolas. Je remercie aussi tous ceux de mon laboratoire d'adoption pendant quelques mois, le LPSM : Iqraa, Ariane, Antonio, Mathis, Camilla, Miguel, Grâce, Lucas, Ludovic, Pierre et Alexis. Je remercie aussi tous les chercheurs que j'ai eu la chance de croiser, dans des laboratoires ou dans des séminaires et workshops et qui m'ont beaucoup inspiré : Pierre Gloaguen, Marie Perrot, Antoine Godichon, Alain Durmus, Rémi Bardenet, Gabriel Lang, Arnaud Guyader, Stéphane Robin, Marie-Pierre Etienne, Sophie Donnet, Randal Douc.

Je tiens à remercier tout d'abord la musique, et bien sûr tous mes professeurs et amis musiciens, sans vous cela n'aurait pas été possible ni envisageable. D'abord Eric Lohrer avec qui j'ai l'opportunité d'avoir cours tous les jeudis. À chaque fois que je prends la guitare, je me dis d'essayer de faire comme Eric. Ce n'est toujours pas encore ça, mais bon, je sais où est le nord ! C'est un musicien incroyable et le maître du bon goût. En plus d'être quelqu'un de très sensible et qui m'a beaucoup aidé pendant des moments durs avec seulement quelques mots parfois. Je suis désolé si je n'ai pas été toujours un très bon élève, c'est qu'il y avait un autre Eric tu sais... En tout cas, c'est toujours un plaisir de jouer avec toi et je sais que tu sais, mais dès qu'il y a un concert à Paris j'y serai. Puis à mes professeurs d'ensemble William Carrossella, Guillaume Naud. Je voudrais remercier tous mes amis musiciens que j'admire et avec qui j'ai pu partager la scène : David, Charlie, Melissa, Thomas, Raphaël, Matéo, Germain, Victor, Hippolyte, Inès, Arturo, Nicolas et Axel.

Puis à mes amis qui ont assuré le "fluctuat nec mergitur" à divers moments: Antoine, Kahina, Thomas, Margot, Arturo, Daniel, Lucas, Gabriel (Rovina), César, Salvador, Péqui, Eduardo, Thomas Boudou, Dinara, Quentin, Philippe, Michel, Helenka. Antoine, Kahina, Thomas et Margot ont été mes premiers contacts avec les natifs. Je dis toujours que j'ai de la chance d'etre tombé sur eux, car je dois avouer qu'au début je ne comprenais pas vraiment ce qu'ils racontaient ! Mais je n'aurais pas pu mieux choisir. Arturo qui est aussi mon partenaire de musique, un vrai grand frère. Daniel, Lucas, Rovina, César,

Salvador, Péqui et Eduardo la troupe brésilienne avec qui on a découvert Paris et l'Europe. A Thomas, Dinara, Quentin et Philippe pour tous ces midis au Terminus ! Michel et Helenka pour être toujours des sources de franchement n'importe quoi inestimables. J'aimerais aussi remercier Ula et Sarah qui viennent d'arriver sur le bateau mais qui ont déjà aidé à boucher les trous sur la coque.

À minha família que, mesmo querendo estar mais perto, teve que acompanhar essa aventura de longe. Obrigado pelo apoio e pelo amor incondicional mesmo à distância.

Puis puisqu'on parle de bateau, Mme la capitaine Anna la prof, merci d'avoir embarqué et de me montrer que tout est encore possible.

Contents

Chapter 1

Introduction

This introduction describes the general context of the thesis and surveys the main results present in this thesis. We start by describing the problem of evaluating the risk of cardiac sudden death using non-invasive data which is the guiding problem behind all the theoretical contributions provided in this thesis and more directly addressed in Chapter 6. We then introduce the theoretical objects present in this thesis, namely importance sampling, Sequential Monte Carlo (SMC) methods, denoising diffusion generative models (DDGM) and inverse problems.

1.1 Notations:

Let $\mathbb{R}_+ := [0, \infty)$, $\mathbb{R}_+^* := (0, \infty)$, $\mathbb{N} := \{0, 1, 2, \ldots\}$, and $\mathbb{N}^* := \{1, 2, 3, \ldots\}$ denote the sets of nonnegative and positive real numbers and the same for integers, respectively. We denote by I_N the $N \times N$ identity matrix. For any quantities $\{a_\ell\}_{\ell=m}^t$, we denote vectors as $a_{m:t} := (a_m, \ldots, a_t)$, and for any $(m, t) \in \mathbb{N}^2$ such that $m \leq t$, we let $[\![m, t]\!] := \{m, m+1, \ldots, t\}$. For a given measurable space $(\mathbb{X}, \mathcal{X})$, where $\mathcal{X}$ is a countably generated σ-field, we denote by $\mathsf{F}(\mathcal{X})$ the set of bounded $\mathcal{X}/\mathcal{B}(\mathbb{R})$-measurable functions on $\mathbb{X}$. For any $h \in \mathsf{F}(\mathcal{X})$, we let $\|h\|_\infty := \sup_{x \in \mathbb{X}} |h(x)|$ and $\mathrm{osc}(h) := \sup_{(x,x') \in \mathbb{X}^2} |h(x) - h(x')|$ denote the supremum and oscillator norms, respectively, of h. Let $\mathsf{M}(\mathcal{X})$ be the set of σ-finite measures on $(\mathbb{X}, \mathcal{X})$, and $\mathsf{M}_1(\mathcal{X}) \subset \mathsf{M}(\mathcal{X})$ be the probability measures.

Let $(\mathbb{Y}, \mathcal{Y})$ be another measurable space. A possibly unnormalized transition kernel K on $\mathbb{X} \times \mathcal{Y}$ induces two integral operators, one acting on measurable functions, and the other on measures; specifically, for $h \in \mathsf{F}(\mathcal{X} \otimes \mathcal{Y})$ and $\nu \in \mathsf{M}_1(\mathcal{X})$, define the measurable function

$$Kh : \mathbb{X} \ni x \mapsto \int h(x, y)\, K(x, \mathrm{d}y)$$

and the measure

$$\nu K : \mathcal{Y} \ni A \mapsto \int K(x, A)\, \nu(\mathrm{d}x),$$

whenever these quantities are well defined. Now, let $(\mathbb{Z}, \mathcal{Z})$ be a third measurable space and L be another possibly unnormalized transition kernel on $\mathbb{Y} \times \mathcal{Z}$; we then define, with K as above, two different products of K and L, namely,

$$KL : \mathbb{X} \times \mathcal{Z} \ni (x, A) \mapsto \int L(y, A)\, K(x, \mathrm{d}y)$$

and

$$K \otimes L : \mathbb{X} \times (\mathcal{Y} \otimes \mathcal{Z}) \ni (x, A) \mapsto \iint \mathbb{1}_A(y, z)\, K(x, \mathrm{d}y)\, L(y, \mathrm{d}z),$$

whenever these are well defined. This also defines the $\otimes$ product of a kernel K on $\mathbb{X} \times \mathcal{Y}$ and a measure ν on $\mathcal{X}$, as well as of a kernel L on $\mathbb{Y} \times \mathcal{X}$ and a measure μ on $\mathcal{Y}$, as the measures

$$\nu \otimes K : \mathcal{X} \otimes \mathcal{Y} \ni A \mapsto \iint \mathbb{1}_A(x, y)\, K(x, \mathrm{d}y)\, \nu(\mathrm{d}x),$$
$$L \otimes \mu : \mathcal{X} \otimes \mathcal{Y} \ni A \mapsto \iint \mathbb{1}_A(x, y)\, L(y, \mathrm{d}x)\, \mu(\mathrm{d}y).$$

1.2 Résumé en Français :

Les sections qui suivent présentent une introduction de haut niveau aux objets mathématiques qui font partie des contributions apportées dans cette thèse. Nous présentons aussi brièvement les contributions apportées et faisons référence aux chapitres (en anglais) où elles sont présentés en détail.

1.2.1 Introduction du problème général

Environ 10% des décès chez les adultes en Europe et aux États-Unis sont dus à une mort subite d'origine cardiaque (MSOC), souvent incorrectement désignée comme un "arrêt cardiaque". La MSOC survient généralement à la suite d'arythmies ventriculaires extrêmement rapides, c'est-à-dire une fibrillation ventriculaire ou une tachycardie ventriculaire (FV/TV). Ces arythmies ventriculaires rapides sont souvent associées à des maladies cardiaques structurelles telles que les cardiomyopathies ou des zones d'hétérogénéité électrique cardiaque localisées Haïssaguerre et al. (2018). La détection et la quantification de ces rythmes cardiaques anormaux à l'aide de techniques non invasives telles que l'électrocardiogramme (ECG) constituent l'un des plus grands défis en cardiologie. Des traitements efficaces sont disponibles pour protéger les individus à risque, donc une évaluation précise est cruciale.

À ce jour, la cardiologie s'est appuyée sur les mesures de la fraction d'éjection du ventricule gauche (LVEF en anglais) pour évaluer le risque de MSOC. Bien que la LVEF soit utile, elle a une utilité limitée chez les patients plus jeunes sans cardiomyopathies. En effet, la LVEF est une mesure corrélée à la capacité de contraction de l'ensemble du ventricule gauche, donc moins sensible aux hétérogénéités cardiaques électriques non structurelles/localisées.

Étant donné que la MSOC nécessite une réponse exceptionnellement rapide pour prévenir les décès, il est extrêmement difficile de collecter des données non invasives directement auprès de cette population. Une approche alternative consiste à utiliser la distribution de signaux sains, car les bases de données contenant de telles données sont plus facilement disponibles Kang and Wen (2022); Wen and Kang (2021). Cette approche peut impliquer la détection d'anomalies ou de valeurs aberrantes dans les données et peut s'appuyer sur un modèle génératif capable d'approximer selon une certaine métrique statistique précis la distribution des signaux de patients sains. C'est la voie que nous avons choisie d'aborder le sujet dans cette thèse de doctorat.

Au cours de la dernière décennie, plusieurs techniques ont été développées pour concevoir et entraîner des modèles génératifs capables de générer des motifs hautement réalistes à partir des données originales, même pour des types de données complexes de grande dimension tels que les images et l'audio Kingma et al. (2019); Kobyzev et al. (2020); Gui et al. (2021). Un modèle génératif vise à construire une distribution p qui approche une distribution d'intérêt $\mathsf{q}_{\mathrm{data}}$ en ne s'appuyant que sur des échantillons i.i.d. de $\mathsf{q}_{\mathrm{data}}$. Il existe plusieurs façons de détecter des anomalies à l'aide d'un modèle génératif. Nous nous concentrons sur la tâche de détection d'incompatibilités dans les données.

Les sujets à risque de MSOC avec des mesures de LVEF normales auront probablement des anomalies localisées dans l'activité électrique du cœur. Étant localisées, nous nous attendons à ce que ces anomalies se manifestent plus nettement dans les dérivations ECG qui sont physiquement plus proches de la source de l'anomalie dans le cœur. Par conséquent, on peut utiliser un modèle génératif pour reconstruire un sous-ensemble de dérivations connaissant le sous-ensemble complémentaire de dérivations. Nous pouvons voir ce problème comme un problème d'"inpainting", mais pour des données de type ECG. Ce type de problème peut être formulé comme la résolution d'un problème inverse en utilisant la distribution issue du modèle génératif comme distribution a priori.

La question guide de cette thèse est la suivante :

Est-il possible de créer un modèle génératif capable de détecter des anomalies dans les données ECG qui ne s'appuie que sur un ensemble de données ECG saines et qui est fondé théoriquement ?

1.2.2 Problème inverse

Le terme problème inverse est utilisé lorsque l'on souhaite inférer à partir d'un vecteur d'observations indirectes $y \in \mathbb{R}^{\mathsf{d}_y}$ le vecteur sous-jacent d'inconnues $x \in \mathbb{R}^{\mathsf{d}_x}$. Nous supposons une connaissance d'un modèle (direct) reliant y et x défini par la fonction

$$f : (x, \varepsilon) \in \mathbb{R}^{\mathsf{d}_x} \times \mathbb{R}^{d_\varepsilon} \to f(x, \varepsilon) \in \mathbb{R}^{\mathsf{d}_y} ,$$

où ε est un vecteur de bruit inconnu, représentant l'aléatoire du modèle et/ou l'erreur de la mesure. L'un des modèles directs les plus courants est le modèle linéaire lorsque $f(x, \varepsilon)$ est de la forme

$$f(x, \varepsilon) = \mathrm{A}x + \sigma\varepsilon ,$$

avec $\mathrm{A} \in \mathbb{R}^{\mathsf{d}_y \times \mathsf{d}_x}$ l'opérateur direct linéaire. Ce modèle général est souvent utilisé dans le domaine de l'imagerie computationnelle, y compris diverses applications d'imagerie tomographique telles que les types courants d'imagerie par résonance magnétique Vlaardingerbroek and Boer (2013), la tomographie assistée par ordinateur aux rayons X Elbakri and Fessler (2002), l'imagerie radar Cheney and Borden (2009), et des tâches de restauration d'image de base telles que la super-résolution et le remplissage d'image González et al. (2009).

L'approche classique pour résoudre les problèmes inverses linéaires s'appuie sur des connaissances a priori sur x, telles que sa régularité, sa parcimonie dans un dictionnaire ou ses propriétés géométriques. Ces approches tentent d'estimer un $\widehat{x}$ en minimisant un problème inverse régularisé,

$$\hat{x} \in \operatorname{argmin}_x |y - \mathrm{A}x|^2 + \mathrm{Reg}(x) ,$$

où Reg est un terme de régularisation qui équilibre la fidélité aux données et le bruit tout en permettant des calculs efficaces. Cependant, une difficulté courante dans le problème inverse régularisé est la sélection d'un régularisateur approprié, qui a une influence décisive sur la qualité de la reconstruction. Bien que les problèmes inverses régularisés continuent de dominer le domaine, de nombreuses autres **formulations statistiques** ont été proposées ; voir Besag et al. (1991); Idier (2013); Marnissi et al. (2017) et les références qui y sont citées - voir également Stuart (2010) pour une perspective mathématique. Un avantage principal des **approches statistiques** est qu'elles permettent une **quantification de l'incertitude** dans la solution reconstruite ; voir Dashti and Stuart (2017).

La **formulation de Bayes** du problème inverse régularisé est basée sur la considération de l'état X et du bruit ε comme des variables aléatoires définies sur un espace d'états $(\mathbb{X}, \mathcal{X})$. Plus précisément, la formulation de Bayes consiste à considérer

$$Y = f(X, \varepsilon) ,$$

où $X \sim \rho$ et $\varepsilon \sim p_\varepsilon$. ρ est appelée la **distribution à priori** et p_ε la distribution du bruit. La densité de la distribution conditionnelle de Y étant donné X est appelée la **fonction de vraisemblance** et notée $g_0^y(x)$. En utilisant le théorème de Bayes, nous obtenons la distribution proxy non normalisée de la **distribution à posteriori**

$$\gamma(x) := g_0^y(x)\rho(x) .$$

La distribution d'intérêt, la distribution à posteriori elle-même, est définie comme

$$\pi(\mathrm{d}x) := \gamma(\mathrm{d}x)/\gamma(\mathbb{X}) .$$

En général, on s'intéresse à interroger la distribution a posteriori π avec une fonction mesurable $h : \mathbb{X} \to \mathbb{R}^m$, avec $m \in \mathbb{N}^*$ par

$$\pi h := \int h(x)\pi(\mathrm{d}x)\,.$$

C'est le cas si l'on souhaite calculer les moments de π ($h(x) = x^k$) ou la probabilité de X étant dans un certain ensemble $A \in \mathcal{X}$ ($h(x) = \mathbb{1}_A(x)$). Mais en général, π n'est pas disponible sous forme fermée et plusieurs estimateurs ont été proposés pour estimer πh en ne s'appuyant que sur γ.

Nous procédons maintenant à une brève introduction a certains éléments qui vont être utilisés dans les autres chapitres de cette thèse. Une introduction détaillée est donnée dans la section en anglais de ce chapitre.

1.2.3 Estimateurs autonormalisées

L'échantillonnage préférentiel (IS en anglais) est un algorithme qui produit une estimation de l'intégrale $\int h(x)\pi(\mathrm{d}x)$ en utilisant une distribution auxiliaire λ facile d'échantillonner. Cet algorithme est utile lorsque l'on ne connaît qu'une version non normalisée de π ou lorsque h prend des valeurs non nulles sur les queues de π.

Soit $\mathrm{d}\pi/\mathrm{d}\lambda$ la dérivée de Radom-Nikodym de π par rapport à λ. Pour tout h intégrable par rapport à π, $\pi h = \int h(x)\frac{\mathrm{d}\pi}{\mathrm{d}\lambda}(x)\lambda(\mathrm{d}x)$. En général, nous n'avons pas accès à la dérivée de Radom-Nikodym, mais à une fonction proxy $w := \mathrm{d}\gamma/\mathrm{d}\lambda = \gamma(\mathbb{X})\mathrm{d}\pi/\mathrm{d}\lambda$. Par conséquent, nous pouvons écrire

$$\pi h = \int h(x)\frac{\mathrm{d}\gamma}{\mathrm{d}\lambda}(x)\lambda(\mathrm{d}x) \Big/ \int \mathbb{1}_{\mathbb{X}}(x)\frac{\mathrm{d}\gamma}{\mathrm{d}\lambda}(x)\lambda(\mathrm{d}x)\,.$$

L'estimateur d'échantillonnage préférentiel autonormalisé (SNIS) consiste à calculer une approximation de Monte Carlo des deux intégrales avec le même ensemble d'échantillons, c'est-à-dire

$$\Pi_N h(X^{1:N}) := \sum_{i=1}^N w(X^i)h(X^i) \Big/ \sum_{j=1}^N w(X^j) = \sum_{i=1}^N \omega_N^i h(X^i)\,,$$

avec $N \in \mathbb{N}^*$, $\omega_N^i = w(X^i)/\sum_{j=1}^N w(X^j)$ et $X^{1:N} = (X^1, \cdots, X^N)$ échantillons i.i.d provenant de λ.

Bien que chaque estimation de Monte Carlo soit une estimation sans biais de chaque intégrale, l'estimateur SNIS est biaisé, c'est-à-dire $\mathbb{E}\left[\Pi_N h\right] \neq \pi h$. Sous réserve que $\lambda(w^2) < \infty$, le biais et l'erreur quadratique moyenne (MSE) de l'estimateur SNIS sur les fonctions de test bornées ff satisfaisant $\|f\|_\infty \leq 1$ sont donnés respectivement (voir (Agapiou et al., 2017, Théorème 2.1)) par

$$|\mathbb{E}[\Pi_N f(X^{1:N})] - \pi f| \leq (12/N)\kappa[\pi, \lambda], \quad \mathbb{E}[\{\Pi_N f(X^{1:N}) - \pi f\}^2] \leq (4/N)\kappa[\pi, \lambda]\,,$$

où $\kappa[\pi, \lambda] = \lambda(w^2)/\lambda^2(w)$.

Cette borne montre que le biais / MSE des estimations diminue en augmentant NN ou en réduisant $\kappa[\pi, \lambda]$. En effet, la conception de propositions plus adaptées est un domaine de recherche actif, avec plusieurs axes différents étant poursuivis tels que les algorithmes d'échantillonnage d'importance adaptatifs (voir Elvira and Martino (2021) et les références à l'intérieur) et les "Normalizing flows" (voir Papamakarios et al. (2021) et les références à l'intérieur) pour n'en citer que quelques-uns. Pour un λ donné, des estimateurs sans biais construits à partir des estimateurs SNIS ont été proposés par Middleton et al. (2019). L'un des problèmes avec de tels estimateurs est que le nombre d'échantillons de λ utilisés pour produire chaque estimation est aléatoire.

L'algorithme iterated sampling importance resampling (ISIR) est une méthode liée à l'algorithme SNIS qui permet de construire une chaîne de Markov qui converge vers la distribution cible π. Cette méthode peut être vue comme une version itérative de l'algorithme sampling importance resampling (SISR) proposé par Rubin (1987b).

Lorsque l'on utilise l'algorithme ISIR pour construire la chaîne de Markov, il est tentant de réutiliser tous les candidats des étapes intermédiaires pour construire un estimateur similaire à SNIS, étant donné que les poids normalisés sont disponibles. Ce type d'estimateur, souvent appelé ISIR recyclé, a été suggéré par Tjelmeland (2004b) et apparaît également dans Schwedes and Calderhead (2021) et Naesseth et al. (2020).

Une des questions auxquelles nous nous sommes intéressés lors de cette thèse est la suivante :

Quel est le biais des estimations de l'algorithme ISIR recyclé ? Quelle est la meilleure allocation de ressources ? Faut-il privilégier des chaînes plus longues avec des bassins de candidats plus petits a chaque itération d'ISIR ou l'inverse ?

Nous proposons dans le chapitre 2 une analyse théorique et numérique de l'estimateur suggéré par Tjelmeland (2004a). Nous proposons en suite un nouvel estimateur semblable au ISIR recyclé qui permet à la fois la diminution du biais par rapport à un estimateur SNIS tout en conservant la même ordre de grandeur de l'erreur quadratique. Ce chapitre correspond à l'article Cardoso et al. (2022c), accepté et publié à la conférence "Advances in Neural Information Processing Systems" 2022.

Cette question peut aussi être étendu aux problèmes de filtrage et lissage dans les modèles de Markov cachés (HMM). Les HMM sont des modèles statistiques couramment utilisés pour les données séquentielles.

Les HMM impliquent un processus d'état non observable, noté $\{X_t\}_{t\in\mathbb{N}}$, et des données observées, représentées par $\{Y_t\}_{t\in\mathbb{N}}$. Ces processus évoluent dans deux espaces mesurables distincts : $(\mathbb{X}, \mathcal{X})$ pour le processus d'état et $(\mathbb{Y}, \mathcal{Y})$ pour les observations. Les HMM sont définis par les deux propriétés suivantes :

- Le processus d'état, $\{X_t\}_{t\in\mathbb{N}}$, est une chaîne de Markov, caractérisée par des noyaux de transition $(M_{t+1})_{t\in\mathbb{N}}$ et une distribution initiale, η_0.
- Étant donné $\{X_t\}_{t\in\mathbb{N}}$, les observations $\{Y_t\}_{t\in\mathbb{N}}$ sont indépendantes, et nous notons la distribution de Y_t étant donné X_t comme $G_t(X_t, .)$ et sa densité par rapport à la mesure de Lebesgue comme $g_t(x_t, .)$.

Les HMM sont utilisés dans plusieurs domaines différents, tels que le climat Robertson et al. (2004), l'écologie Michelot et al. (2016) et la biologie Jarner et al. (2001); Shihab et al. (2012). Il y a deux principales distributions d'intérêt :

- la distribution de *filtrage*, c'est-à-dire la loi de X_t étant donné $Y_{0:t}$,
- la distribution de *lissage*, c'est-à-dire la loi de $X_{0:t}$ étant donné $Y_{0:t}$.

Comme la distribution de filtrage est la marginale de la distribution de lissage à l'instant t, le problème d'estimation de chaque distribution à partir d'une séquence de données $Y_{0:t}$ est étroitement lié. Dans plusieurs cas, comme l'apprentissage de paramètres dans le cas de l'algorithme EM par exemple, on peut s'intéresser à l'intégrale d'une fonction $\mathcal{X}$-mesurable h sur la distribution de lissage. Par exemple, l'énergie totale $\mathbb{E}_{X_{0:t}|Y_{0:t}}\left[\sum_{i=0}^{t} X_i^2\right]$, ou la corrélation croisée moyenne des états, à savoir $\mathbb{E}_{X_{0:t}|Y_{0:t}}\left[\sum_{i=1}^{t} X_i X_{i+1}^T\right]$.

Sauf dans des cas simples, les lois de filtrage et de lissage ne sont pas disponible de façon analytique. Ces distributions peuvent être estimées à l'aide de méthodes d'échantillonnage préférentiel ou des méthodes dites de Monte Carlo séquentiel (SMC).

À mesure que la longueur de la séquence augmente, la dimension de l'espace d'état résultant augmente également, ce qui rend finalement l'application de l'échantillonnage d'importance inapplicable. L'approche proposée dans Gordon et al. (1993), que nous présentons, offre une solution en faisant évoluer le "pool" d'échantillons de manière séquentielle. Plus précisément, cela implique de répliquer les échantillons qui possèdent des poids d'importance importants tout en éliminant ceux dont les poids sont négligeables.

De la même manière que nous pouvons voir le SMC comme une généralisation de l'idée d'échantillonnage préférentiel aux données séquentielles, les méthodes nommées filtre particulaire de Gibbs Andrieu et al. (2010a) peut être vu comme une extension de ISIR aux cas séquentiel. De la même manière, nous pouvons voir une analogie entre les estimateurs du type SNIS et des algorithmes de intégration sur les lois de lissage tel que l'algorithme dit PaRIS Olsson and Westerborn (2017)

Cela pose la question suivante :

Est-il possible de généraliser l'idée de recyclage des échantillons d'ISIR au recyclage des échantillons dans un algorithme telle que l'algorithme PaRIS?

Cette question est la question sous-jacente aux chapter 3. Dans le chapitre 4 nous faisons une analyse théorique et numérique de l'algorithme de descente de gradient lorsque les gradients sont estimées en utilisant l'estimateur "recyclé" proposé dans le chapitre 3. Ces deux chapitres correspondent aux articles Cardoso et al. (2022b) et Cardoso et al. (2023a) qui sont, respectivement, acceptés pour publication dans le journal Statistica Sinica et acceptés et publiés à la conférence "International Conference in Machine learning" 2023.

1.2.4 Modèles génératifs

La tâche de modélisation générative consiste à trouver, pour une distribution d'intérêt q_{data} définie sur $\mathbb{R}^d$, une fonction paramétrique

$$f_\theta : \mathbb{R}^{d_s} \to \mathbb{R}^d$$

capable de transformer une distribution de bruit λ définie sur $\mathbb{R}^{d_s}$ en une distribution qui se rapproche de q_{data}. Dans la plupart des applications, $\lambda = \mathcal{N}(0, \mathrm{I})$. Plus précisément, la tâche de modélisation générative consiste à trouver θ tel que

$$\mathsf{q}_{\text{data}} \approx p_\theta \,,$$

où $\approx$ signifie que les deux distributions sont proches en termes de mesure de dissimilarité statistique, telle que la distance de Wasserstein ou la divergence de Kullback-Leibler (KL).

Dans ce contexte, nous supposons avoir un ensemble de données d'échantillons i.i.d. de q_{data} qui peut être utilisé pour apprendre le paramètre θ. Cependant, certains modèles génératifs ne nécessitent qu'un accès à une approximation de la densité de q_{data} au lieu d'un ensemble de données. Le modèle génératif idéal permettrait un échantillonnage rapide de divers échantillons de haute qualité et une évaluation tractable de la fonction de densité sous-jacente.

Au cours des dernières années, plusieurs familles de modèles génératifs basés sur des réseaux de neurones profonds (DGM) ont été introduites. Chaque famille a ses inconvénients. Nous présentons maintenant une introduction de haut niveau aux modèles génératifs de diffusion de débruitage (DDGM), qui seront présentés dans plusieurs chapitres de cette thèse.

Avant d'aborder les modèles génératifs de diffusion dans leur forme moderne tels que présentés par Song et al. (2021c), il est crucial de comprendre les "Noise Conditional Score Networks" (NCSN) introduits par Song and Ermon (2019). Les NCSN représentent la première approche à surpasser les GANs (Generative adversarial networks Goodfellow et al. (2014)) dans les tâches de génération d'images sans utiliser d'entraînement adversarial.

Les NCSN s'appuient sur l'algorithme de Langevin non ajusté (ULA) Roberts and Tweedie (1996) et l'appariement du score Hyvärinen (2005), défini comme la dérive de la log densité d'une distribution par rapport à une certaine mesure de référence, normalement la mesure de Lebesgue. L'ULA génère des échantillons approximatifs d'une distribution d'intérêt q_{data}, en utilisant le gradient de la densité (ou score), défini comme $\nabla \log \mathsf{q}_{\text{data}}$. L'ULA construit une chaîne de Markov $X_{t t\in\mathbb{N}}$ avec des étapes de mise à jour utilisant le score et un terme de bruit ϵ_t. Notamment, la chaine est défini à partir d'un X_0 par l'équation

$$X_t{:=}X_{t-1} + \gamma \nabla \log \mathsf{q}_{\text{data}}(X_{t-1}) + (2\gamma)^{1/2}\epsilon_t \,,$$

où γ est un réel positif. Comme montré dans Durmus and Moulines (2017); Durmus et al. (2019), ULA produit des échantillons que sont arbitrairement proches (en KL) de la distribution cible q_{data} si γ est suffisamment petit et que la chaine de Markov a une longueur approprié.

L'appariement de score, comme défini par Hyvärinen (2005), consiste à approximer le score $\nabla \log \mathsf{q}_{\text{data}}$ à partir d'échantillons i.i.d. de q_{data}, sans estimer directement la densité. L'approximation est faite à partir d'un réseau de neurones s_θ. Ce réseau de neurones est utilisé pour minimiser une fonction de perte associée au score. L'approche propose par Hyvärinen (2005) introduit la fonction de perte

$$\mathbb{E}_{X\sim\mathsf{q}_{\text{data}}}\left[\text{tr}(\nabla \mathsf{s}_\theta(X)) + (1/2)\|\mathsf{s}_\theta(X)\|^2\right] \,.$$

Cette approche est connue pour être notamment difficile du point de vue numériquement, car elle fait intervenir la trace do score.

Song and Ermon (2019) propose de créer une séquence de distributions $\{\mathsf{q}_t\}_{t\in[\![0,n]\!]}$ en transformant la distribution des données par un noyau gaussien de variance croissante. Notamment, en introduisant les noyaux

$$q_{t|0}(x_t|x_0) = \mathcal{N}(x_t; x_0, \upsilon_t^2 \,\mathrm{I}_d) \,,$$

où $\{\upsilon_t^2\}_{t\in[\![0,n]\!]}$ est une suite positive croissante, nous définissons

$$\mathsf{q}_t(x_t){:=}\int \mathsf{q}_{\text{data}}(\mathrm{d}x_0) q_{t|0}(x_t|x_0) \,.$$

Cela a deux intérêts. Le premier étant que le score de chaque distribution intermédiaire q_t est alors appris via le Denoising Score Matching (DSM) Vincent (2011), qui engendre une fonction de perte équivalente mais plus abordable que celui de Hyvärinen (2005). En particulier, Song and Ermon (2019) propose d'utiliser le même réseau pour tous les niveaux de bruit υ_t en utilisant le niveau de bruit lui-même comme une entrée du réseau. Le deuxième est qu'en augmentant le niveau de bruit les distributions deviennent de plus en plus simples, donc plus facilement abordables à partir des méthodes du type ULA initialisé sur des distributions raisonnables.

En Song and Ermon (2019), l'ULA est utilisé pour générer des échantillons de façon séquentielle de chacune des lois q_t. Cela est fait en initialisant ULA pour q_n avec des échantillons de $\mathcal{N}(0, \upsilon_n \,\mathrm{I})$ et pour chaque niveau de bruit $t \in [\![0, n-1]\!]$, initialisant l'algorithme ULA visant q_t avec le dernier échantillon obtenu pour q_{t+1}.

Comme mentionné ci-dessus, les algorithmes de diffusion génératifs basés sur cette approche ont démontré des performances de pointe dans la génération d'images, notamment en battant les GANs sur des tâches comme la génération d'images CIFAR-10 Song and Ermon (2019). Cependant, leur inconvénient réside dans le temps d'inférence, nécessitant de nombreux pas d'ULA pour obtenir des échantillons de haute qualité.

Les modèles génératifs de diffusion par débruitage (DDGM) Song et al. (2021c) visent à enlever les étapes d'ULA et d'échantillonner de façon approximée directement q_t à partir d'un échantillon (approximé aussi) de q_{t+1}. Cela passe par une réformulation des lois intérmediaires $\{\mathsf{q}_t\}_{t\in[\![0,n]\!]}$ en utilisant des chaînes

de Markov ou des équations différentielles stochastiques/ordinaires. Une des formulations possibles, que nous appelons DDIM Song et al. (2021a), consiste à obtenir les lois $\{\mathsf{q}_t\}_{t\in[\![1,n]\!]}$ comme des lois marginales d'une loi étendue $\mathsf{q}_{0:n}$. Il s'avère que pour $t \in [\![0, n-1]\!]$ et que

$$X_{0:n} \sim \mathsf{q}_{0:n}\,,$$

la loi de $X_t|X_{t+1}, X_0$ est connue analytiquement. Nous appelons cette loi un "pont d'inférence". En se basant sur la remarque que l'apprentissage du score dans le cas de bruitage gaussien est équivalent a l'apprentissage d'un "débruiteur" optimale faite en Vincent (2011), Song et al. (2021a) propose une méthode de transition progressive entre les distributions en utilisant des "ponts d'inférence". Cela est obtenu en remplaçant X_0 par le débruitage de X_{t+1} dans la loi de $X_t|X_{t+1}, X_0$. Nous obtenons ainsi un noyau Gaussien $p_{t|t+1}$ et la chaine de Markov renversé ("backwards") dont la loi est notée $\mathsf{p}_{0:n}$. La formulation mathématique de ce qui a été décrit ci-dessus est faite en dans la section 1.5. Il est possible de montrer que la chaine de Markov ainsi obtenue correspond a l'optimum d'un problème d'inférence variationnelle sur une famille paramétrique des chaines de Markov dont les noyaux de transition sont des noyaux Gaussiens de variance prédéterminée et dont la moyenne est le paramétrique.

Cette méthode améliore l'efficacité des échantillons générés par rapport aux approches utilisant ULA, tout en maintenant la qualité des échantillons obtenus. Les modèles DDGM, en particulier les formulations DDIM, ont été validés empiriquement pour produire des échantillons d'une qualité remarquable dans la génération d'images.

L'utilisation des DDGM comme prior ouvre un champ de recherche riche, notamment pour résoudre des problèmes inverses bayésiens. Une des propriétés utiles des DDGMs dans ce cas est le fait que la génération dans les DDGM consiste de plusieurs étapes de simulation d'une chaine de Markov avec des noyaux Gaussiens. Il est donc possible d'intervenir à plusieurs étapes de la génération. Divers travaux de recherche ont proposé des méthodes pour échantillonner la distribution postérieure π lorsque la distribution a priori λ est un DDGM (comme décrit dans Song et al. (2021a); Kawar et al. (2022); Lugmayr et al. (2022); Chung et al. (2023)). La distribution postérieure est définie comme

$$\mathsf{p}_0^y(x_0) \propto g_0^y(x_0)\mathsf{p}_0(x_0)\,,$$

où g_0^y représente la fonction de vraisemblance du problème inverse associé.

La distribution postérieure étendue est définie comme suit :

$$\mathsf{p}_0^y(\mathrm{d}x_{0:n}) \propto g_0^y(x_0)\lambda_n(\mathrm{d}x_n)\prod_{t=1}^{n} p_{t-1|t}(\mathrm{d}x_{t-1}|x_t)\,.$$

Les marginales de p_0^y au temps t sont définies comme suit :

$$\mathsf{p}_t^y(A) := \int \mathbb{1}_A(x_t)\mathsf{p}_0^y(\mathrm{d}x_{0:n}) = \int \mathbb{1}_A(x_t)g_0^y(x_0)p_{0|t}(\mathrm{d}x_0|x_t)\mathsf{p}_t(\mathrm{d}x_t) = \int \mathbb{1}_A(x_t)g_t^y(x_t)\mathsf{p}_t(\mathrm{d}x_t)\,,$$

où

$$g_t^y(x_t) := \int g_0^y(x_0)p_{0|t}(\mathrm{d}x_0|x_t)\,.$$

Le score de la postérieure peut être écrit comme suit : $\nabla \log \mathsf{p}_t^y(x_t) = \nabla \log g_t^y(x_t) + \nabla \log p_{t|t+1}(x_t|x_{t+1})$. Notons qu'une estimation du score de la distribution postérieur permettrait la simulation du DDGM équivalent a un DDGM pour la distribution posterieur.

Les méthodes actuelles pour échantillonner p_0^y tentent soit d'approximer p_0^y en créant une version alternative plus facile à échantillonner (comme dans Song et al. (2021a); Kawar et al. (2022); Lugmayr et al. (2022)), soit d'approximer $\nabla \log g_t^y(x_t)$ (comme dans Chung et al. (2023)). Toutes ces méthodes

Figure 1.1: Illustration des échantillons obtenus avec `MCGdiff` *pour des problèmes inverses variés. Le texte sur les côtés indique respectivement le type de problème inverse l'écart type du bruit et le dataset sur lequel le modèle génératif a été entrainé. Pour plus de détails, voir la section 5.3.*

introduisent des erreurs d'approximation irréductibles, ce qui peut conduire à des échantillons qui, bien qu'attrayants dans certaines tâches, peuvent présenter des comportements inattendus dans d'autres. Cette absence de garanties théoriques pose problème, en particulier dans des applications sensibles comme le traitement des données médicales.

Cela introduit la question suivante :

Peut-on dériver un algorithme d'échantillonnage pour la postérieure d'un problème inverse bayésien utilisant un DDGM comme a priori, qui soit théoriquement fondé sous des hypothèses réalistes ?

Nous proposons dans le chapitre 5 un algorithme du type SMC pour échantillonner la distribution postérieur des problèmes inverses linéaires dont la prior vient d'un DDGM, que nous nommons `MCGdiff`. Nous montrons en figure 1.1 quelques exemples des échantillons produits par `MCGdiff` sur des données du type image pour plusieurs problèmes inverses, notamment le coloriage (Col), la super-résolution (SR), le "inpainting" (Inp) et le "Gaussian debluring". Nous fournissons des garanties théoriques qui montrent que notre algorithme est asymptotiquement exacte. Nous montrons aussi, que sur des problèmes où la distribution a posteriori est connue, l'algorithme proposé obtient des meilleures performances sur diverses métriques liées aux distances en distribution que l'état de l'art dans le domaine. Ce chapitre correspond à l'article Cardoso et al. (2023b), accepté pour présentation orale et publié à la conférence "International Conference in Representation Learning" 2024.

1.2.5 Application à des données du type ECG

Finalement, nous adressons la question posée au début de cette section, notamment, peut-on utiliser un modèle génératif pour détecter des signaux anormaux dans des données du type ECG. Dans le chapitre 6, nous montrons comment, en combinant `MCGdiff` du chapitre 5 avec un DDGM appris sur des données ECG, nous sommes en mesure de résoudre plusieurs tâches de reconstruction ECG différentes mieux que les méthodes actuelles sans aucun réglage fin nécessaire.

Nous montrons en particulier que cet outil peut être précieux pour résoudre la détection d'anomalies sur l'ECG et montrons qu'il distingue efficacement entre la population normale et celles qui ont subi un infarctus du myocarde. Nous adaptons également `MCGdiff` pour gérer le bruit de mesure inconnu en couplant `MCGdiff` avec un algorithme d'ascension de score.

Les résultats obtenus sont présentés dans les figures 1.3 et 1.2. Dans la figure 1.2, nous montrons en rouge les "vrais" signaux ECG d'un patient donné et en bleu les signaux obtenus avec `MCGdiff`. Pour obtenir ces signaux, nous considérons le problème inverse qui consiste à observer les trois premières

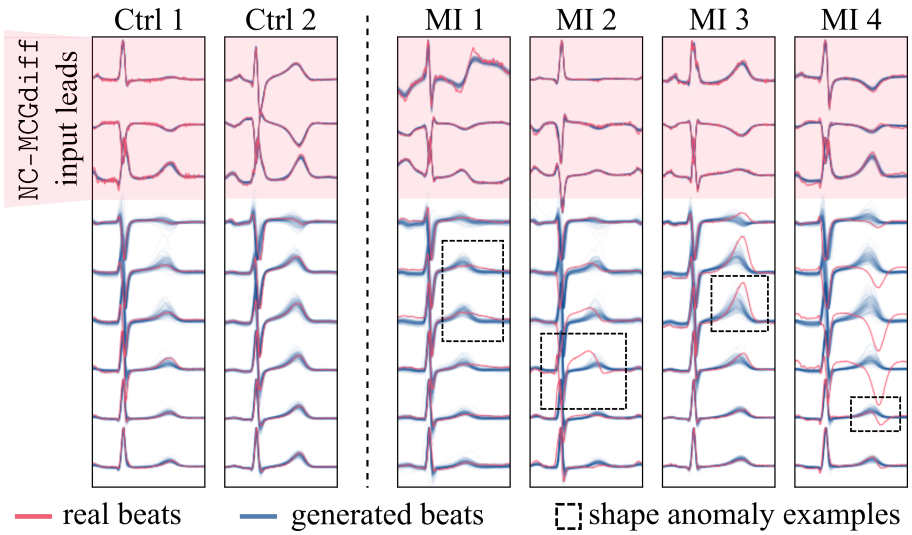


Figure 1.2: Illustration de l'utilisation de `MCGdiff` *pour la détection d'anomalies. "Ctrl" correspond à des patients dit "contrôle" (sans anomalie connue) et "MI" à des patients ayant subi un infarctus du myocarde.*

pistes de l'ECG de chaque patient. Ces pistes sont choisies, car elles représentent les pistes les plus éloignés physiquement du cœur, et donc moins susceptibles aux anomalies. Plus de détails sont donnés dans le chapitre 6. On peut voir que pour les patients du groupe de contrôle (sans anomalie connue), les signaux bleus et les signaux rouge coïncident alors que pour les patients ayant eu un infarctus du myocarde (MI), nous pouvons voir des différences significatives et localisées sur certaines pistes.

Pour quantifier à quel point cette différence est significative, pour chaque patient du groupe contrôle et du groupe MI nous avons calculé la distance de Mahanalobis entre une approximation gaussienne obtenue à partir des échantillons de la distribution à postériori obtenu avec `MCGdiff` et les vrais signaux des patients. Puis nous avons utilisé cette valeur comme score d'anomalie. Dans la figure 1.3 nous voyons la courbe ROC obtenu en faisant du seuillage sur le score d'anomalie ainsi obtenu. Nous pouvons voir que pour les deux sexes, la méthode proposée est capable de faire la distinction entre le groupe de contrôle et le groupe MI.

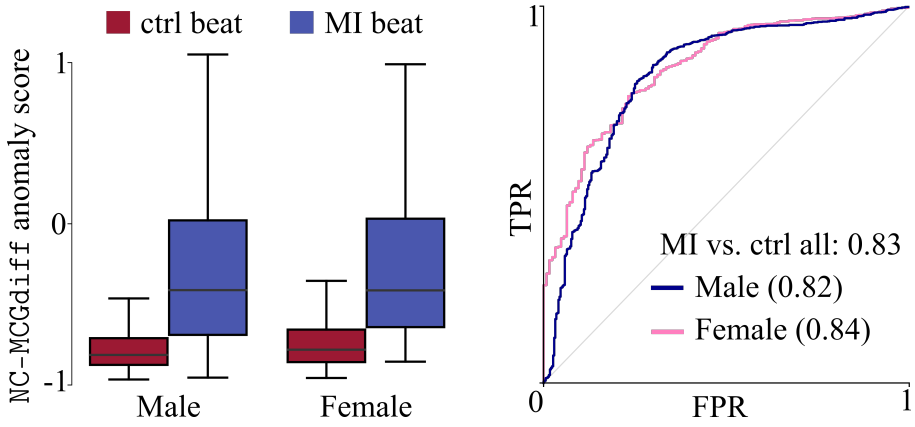


Figure 1.3: ***Gauche.*** *Distribution du score d'anomalie obtenu avec* `MCGdiff` *pour le groupe dit contrôle (rouge) et "MI" (infarctus du myocarde, bleu).* ***Droite.*** *Courbe ROC pour la classification entre contrôle et "MI" obtenue avec le score d'anomalie.*

1.3 General introduction:

1.3.1 The guiding problem: Assessing risk of sudden cardiac death using non-invasive data

Approximately 10% of adult deaths in Europe and the United States are due to sudden cardiac death (SCD), often incorrectly referred to as “cardiac arrest". SCD typically occurs due to extremely rapid ventricular arrhythmias, i.e., ventricular fibrillation or ventricular tachycardia (VF/VT). These rapid ventricular arrhythmias are often associated with structural heart disease such as cardiomyopathies or areas of cardiac electrical heterogeneity Haïssaguerre et al. (2018). Detecting and quantifying these abnormal heart rhythms with noninvasive techniques such as the electrocardiogram (ECG) is one of the greatest challenges in cardiology. Effective treatments are available to protect at-risk individuals, so accurate assessment is critical. To date, cardiology has relied on left ventricular ejection fraction (LVEF) measurements to assess SCD risk. LVEF, although valuable, has limited utility in younger patients without cardiomyopathies. Indeed, LVEF is a measure correlated to the capacity of contraction of the whole left ventricle, thus less sensitive to non-structural / localized cardiac electrical heterogeneity.

Because SCD requires an exceptionally rapid response to prevent deaths, it is extremely difficult to collect noninvasive data directly from this population. An alternative approach is to use the distribution of healthy signals, because databases containing such data are more readily available Kang and Wen (2022); Wen and Kang (2021). This approach may involve detecting outliers or anomalies in the data and can rely on a generative model capable of accurately approximating the distribution of signals of healthy patients. This is the path that we chose to approach the subject in this Ph.D. thesis.

Over the past decade, several techniques have been developed to design and train generative models capable of generating highly realistic patterns from the original data, even for complex high-dimensional data types such as images and audio Kingma et al. (2019); Kobyzev et al. (2020); Gui et al. (2021). A generative model aims to build a distribution p that approximates a distribution of interest q_{data} relying only on i.i.d samples from q_{data}. There are several ways anomalies detection can be done using a generative model. We focus on the task of detecting incompatibilities in the data. Subjects at risk of SCD with normal LVEF measurements will probably have localized abnormalities in the electrical activity of the heart. Being localized, we expect those abnormalities to manifest more prominently in ECG leads that are physically closer to the source of the abnormality in the heart. Therefore, one might use a generative model to reconstruct a subset of leads knowing the complementary subset of leads. We can see this problem as an “inpainting" problem, but for ECG type of data. This kind of problem can be formulated as solving an inverse problem using the distribution issued from the generative model as the prior distribution.

(**Q1**) Can we create a generative model that is able to identify anomalies in ECG data that relies only on a healthy ECG dataset and that is theoretically grounded?

1.3.2 Bayesian linear inverse problems

The term inverse problem is used whenever one wants to infer from a vector of indirect observations $y \in \mathbb{R}^{\mathsf{d}_y}$ the underlying vector of unknowns $x \in \mathbb{R}^{\mathsf{d}_x}$. We assume a knowledge of a (forward) model linking y and x defined by the function

$$f : (x, \varepsilon) \in \mathbb{R}^{\mathsf{d}_x} \times \mathbb{R}^{d_\varepsilon} \to f(x, \varepsilon) \in \mathbb{R}^{\mathsf{d}_y} ,$$

where ε is an unknown noise vector, representing model and/or measurement randomness. One of the most common forward models is the linear model when $f(x, \varepsilon)$ is of the form $\mathrm{A}x + \sigma\varepsilon$, with $\mathrm{A} \in \mathbb{R}^{\mathsf{d}_y \times \mathsf{d}_x}$ the linear forward operator. This general model is used throughout computational imaging, including various tomographic imaging applications such as common types of magnetic resonance

imaging Vlaardingerbroek and Boer (2013), X-ray computed tomography Elbakri and Fessler (2002), radar imaging Cheney and Borden (2009), and basic image restoration tasks such as deblurring, super-resolution, and image inpainting González et al. (2009).

The classical approach to solving linear inverse problems relies on prior knowledge about x, such as its smoothness, sparseness in a dictionary, or its geometric properties. These approaches attempt to estimate a $\widehat{x}$ by minimizing a regularized inverse problem, $\hat{x} \in \operatorname{argmin}_x \{\|y - \mathrm{A}x\|^2 + \mathrm{Reg}(x)\}$, where Reg is a regularization term that balances data fidelity and noise while enabling efficient computations. However, a common difficulty in the regularized inverse problem is the selection of an appropriate regularizer, which has a decisive influence on the quality of the reconstruction. Whereas regularized inverse problems continue to dominate the field, many alternative **statistical formulations** have been proposed; see Besag et al. (1991); Idier (2013); Marnissi et al. (2017) and the references therein - see also Stuart (2010) for a mathematical perspective. A main advantage of **statistical approaches** is that they allow for **uncertainty quantification** in the reconstructed solution; see Dashti and Stuart (2017).

The **Bayes' formulation** of the regularized inverse problem is based on considering the state X and the noise ε as random variables defined over some state space $(\mathbb{X}, \mathcal{X})$. More precisely, the Bayes' formulation consists in considering

$$Y = f(X, \varepsilon)\,,$$

where $X \sim \rho$ and $\varepsilon \sim p_\varepsilon$. ρ is called the **prior** distribution and p_ε the noise distribution. The density of the conditional distribution of Y given X is called the **likelihood** function and denoted as $g_0^y(x)$. We chose to represent it as a function over x since it is the variable that we are interested in. Using Bayes' theorem, we obtain the unnormalized proxy of the **posterior distribution**

$$\gamma(x) := g_0^y(x)\rho(x)\,. \tag{1.1}$$

The distribution of interest, the posterior distribution itself, is defined as $\pi(\mathrm{d}x) := \gamma(\mathrm{d}x)/\gamma(\mathbb{X})$.

In general, one is interested in querying the posterior distribution π with some measurable function $h : \mathbb{X} \to \mathbb{R}^m$, with $m \in \mathbb{N}^*$ through

$$\pi h := \int h(x)\pi(\mathrm{d}x)\,.$$

This is the case if one wants to compute the moments of π ($h(x) = x^k$) or the probability of X being in a certain set $A \in \mathcal{X}$ ($h(x) = \mathbb{1}_A(x)$). But in general, π is not available in closed form, and several estimators have been proposed to estimate πh relying only on γ.

The rest of this chapter is divided as follows. In Section 1.4, we describe two methods that produce so-called "self-normalized" estimations of πh from γ, namely **Importance Sampling** and its generalisation to sequential data, **Particle Smoothing**. In Section 1.5, we present generative models and namely the family of generative models called Denoising diffusion generative models (DDGM). We then highlight the capabilities of such model to serve as an *informative prior* to be used in inverse problems and the problem of sampling from π when ρ is the distribution defined by a DDGM. We conclude this chapter wtih section 1.6 where we introduce the contributions present in this thesis and the organization of the next chapters.

1.4 Self Normalized estimators

1.4.1 Self-Normalized Importance Sampling

Importance Sampling (IS) is an algorithm that produces an estimate of $\int h(x)\pi(\mathrm{d}x)$ through an auxiliary distribution λ from which sampling is easy and that dominates π, i.e., such that for all measurable A, $\pi(A) > 0$ implies $\lambda(A) > 0$. This is notably useful when knowing only an un-normalized proxy γ of π

but is also useful when h is a function that takes non-zeros values on the tails of π. In this case, vanilla Monte Carlo estimators generally yield high variance estimators.

Let $\mathrm{d}\pi/\mathrm{d}\lambda$ denote the Radom-Nikodym derivative of π with respect to λ. For any π-integrable h, $\pi h = \int h(x)\frac{\mathrm{d}\pi}{\mathrm{d}\lambda}(x)\lambda(\mathrm{d}x)$. In general, we do not have access to the Radom-Nikodym derivative, but to a proxy $w{:=}\mathrm{d}\gamma/\mathrm{d}\lambda = \gamma(\mathbb{X})\mathrm{d}\pi/\mathrm{d}\lambda$. Therefore, we can write

$$\pi h = \int h(x)\frac{\mathrm{d}\gamma}{\mathrm{d}\lambda}(x)\lambda(\mathrm{d}x) \Big/ \int \mathbb{1}_{\mathbb{X}}(x)\frac{\mathrm{d}\gamma}{\mathrm{d}\lambda}(x)\lambda(\mathrm{d}x)\,.$$

The Self Normalized importance sampling (SNIS) estimate consists in computing a Monte Carlo approximation of both integrals with the same set of samples, i.e.,

$$\Pi_N h(X^{1:N}){:=}\sum_{i=1}^{N} w(X^i)h(X^i) \Big/ \sum_{j=1}^{N} w(X^j) = \sum_{i=1}^{N} \omega_N^i h(X^i)\,,$$

with $N \in \mathbb{N}^*$, $\omega_N^i = w(X^i)/\sum_{j=1}^N w(X^j)$ and $X^{1:N} = (X^1, \cdots, X^N)$ i.i.d samples from λ.

Even though each Monte Carlo estimate is an unbiased estimate of each integral, the SNIS estimator is biased, i.e, $\mathbb{E}\left[\Pi_N h\right] \neq \pi h$. Provided that $\lambda(w^2) < \infty$, the bias and mean-squared error (MSE) of the SNIS estimator over bounded test functions f satisfying $\|f\|_\infty \leq 1$ are given respectively (see (Agapiou et al., 2017, Theorem 2.1)) by

$$|\mathbb{E}[\Pi_N f(X^{1:N})] - \pi f| \leq (12/N)\kappa[\pi,\lambda], \quad \mathbb{E}[\{\Pi_N f(X^{1:N}) - \pi f\}^2] \leq (4/N)\kappa[\pi,\lambda]\,, \tag{1.2}$$

where $\kappa[\pi,\lambda] = \lambda(w^2)/\lambda^2(w)$.

This bound shows that the bias / MSE of the estimates go down by either increasing N or by reducing $\kappa[\pi,\lambda]$. Indeed, the design of better suited proposals is an active research field, with several different axes being pursued such as Adaptative importance sampling algorithms (see Elvira and Martino (2021) and references within) and Normalizing Flows (see Papamakarios et al. (2021) and references within) to name a few. For a given λ, zero-bias estimators build upon the SNIS estimators have been proposed by Middleton et al. (2019). One of the problems with such estimators is that the number of samples of λ used to produce each estimate is random.

1.4.1.1 Iterated sampling importance resampling algorithm i-SIR

Another way of designing an estimator of πh is through Markov Chain Monte Carlo (MCMC) methods. A MCMC method relies on building an ergodic Markov Chain $\{X_k\}_{k\in\mathbb{N}}$ with invariant distribution π, i.e a chain that gets arbitrarily close to π as k increases. By discarding a burn-in period k_0, one can use the samples $\{X_k\}_{k>k_0}$ to produce a Monte Carlo estimate of πh, with bias decreasing with k_0. If the resulting Markov chain is geometrically ergodic, then the bias of the estimates decrease as κ^{k_0} where $\kappa \in (0,1)$.

There are several ways of building Markov chains that target π given the proxy γ, such as the Metropolis-Hastings algorithm Metropolis et al. (1953). A method that is closely related to SNIS is the *iterated sampling importance resampling* (i-SIR), proposed in Tjelmeland (2004a); see (Andrieu et al., 2010a; Lee et al., 2010; Lee, 2011; Andrieu et al., 2018). The i-SIR can be seen as an iterative application of the *sampling importance resampling* (SISR) algorithm proposed by Rubin (1987b); the k-th iteration is defined as follows. Given a state $Y_k \in \mathbb{X}$, (i) set $X_{k+1}^1 = Y_k$ and draw $X_{k+1}^{2:N}$ independently of the proposal distribution λ; (ii) compute, for $i \in \{1,\ldots,N\}$, the normalized importance weights $\omega_{N,k+1}^i = w(X_{k+1}^i)/\sum_{\ell=1}^N w(X_{k+1}^\ell)$; (iii) select Y_{k+1} from the set $X_{k+1}^{1:N}$ by choosing X_{k+1}^i with probability $\omega_{N,k+1}^i$. We refer to Y_{k+1} and $X_{k+1}^{1:N}$ as *state* and the *candidate pool*, respectively. Following

(Tjelmeland, 2004a), i-SIR may be viewed (up to an irrelevant permutation of the samples) as a two-stage Gibbs sampler targeting an extended probability distribution φ_N on an enlarged state space including the state as well as the candidate pool. As this extended distribution admits π as a marginal with respect to the state, one can expect the marginal distribution of the generated states $\{Y_k\}_{k\in\mathbb{N}}$, forming themselves a Markov chain, to approach the target π of interest as k tends to infinity. Furthermore, if $\|w\|_\infty/\lambda(w) < \infty$, the state and candidate-pool Markov chains $(Y_k)_{k\in\mathbb{N}}$ and $(X_k^{1:N})_{k\in\mathbb{N}}$ can be shown to be uniformly geometrically ergodic, suggesting that the resulting state chain can be used to form MCMC estimates.

But when using i-SIR as the underlying mechanism to build the Markov chain, one is tempted to recycle all the candidate pool $X_k^{1:N}$ to build a SNIS like estimator, since the normalized weights are available. This type of estimator, often called recycled i-SIR was suggested by Tjelmeland (2004b) and also appears in Schwedes and Calderhead (2021) and Naesseth et al. (2020).

(**Q2**) What is the bias of the recycled i-SIR estimates? What is the best allocation of ressources? Making longer chains with smaller candidate pools or the opposite?

1.4.2 Sequential Monte Carlo

Sequential importance sampling serves as a method tailored to address a particular set of challenges known as (non-linear) filtering, which involves sequential data. As the length of the sequence increases, the dimension of the resulting state space also increases, which eventually renders the application of importance sampling unfeasible. The approach proposed in Gordon et al. (1993), which we present, offers a remedy by evolving the sample pool sequentially. Specifically, this involves replicating samples that possess substantial importance weights while eliminating those with negligible weights. Before we delve into the details, we describe the most common statistical model used for sequential data.

Example 1 (Hidden Markov Models)**.** *Hidden Markov Models (HMMs) involve an unobservable state process denoted by $\{X_t\}_{t\in\mathbb{N}}$ and observed data represented by $\{Y_t\}_{t\in\mathbb{N}}$. These processes evolve within two distinct measurable spaces: $(\mathbb{X}, \mathcal{X})$ for the state process and $(\mathbb{Y}, \mathcal{Y})$ for observations. HMM are defined by the following two properties:*

- *The state process, $\{X_t\}_{t\in\mathbb{N}}$, is a Markov chain, characterized by transition kernels $(M_{t+1})_{t\in\mathbb{N}}$ and an initial distribution, η_0.*
- *Given $\{X_t\}_{t\in\mathbb{N}}$, the observations $\{Y_t\}_{t\in\mathbb{N}}$ are independent, and we denote the distribution of Y_t given X_t as $G_t(X_t, .)$ and its density with respect to the Lebesgue measure as $g_t(x_t, .)$.*

HMM are used in several different domains, such as climate Robertson et al. (2004), ecology Michelot et al. (2016) and biology Jarner et al. (2001); Shihab et al. (2012). There are two main distributions of interest:

- *the* filtering *distribution, i.e. the law of X_t given $Y_{0:t}$,*
- *the* smoothing *distribution, i.e. the law of $X_{0:t}$ given $Y_{0:t}$.*

As the filtering distribution is the t marginal of the smoothing distribution, the problem of estimating each distribution from a sequence of data $Y_{0:t}$ is closely related. In several cases, such as parameter learning in the case of the EM algorithm for example, one might be interested in the integral of some $\mathcal{X}$-measurable function h over the smoothing distribution. For example the overall energy $\mathbb{E}_{X_{0:t}|Y_{0:t}}\left[\sum_{i=0}^t X_i^2\right]$, or the averaged cross-correlation of the states, namely $\mathbb{E}_{X_{0:t}|Y_{0:t}}\left[\sum_{i=1}^t X_i X_{i+1}^T\right]$.

We now proceed to define *Feynman–Kac path measures* that provide a general framework for treating a sequence of distributions such as those defined by the states of an HMM. For a sequence $\{M_t\}_{t\in\mathbb{N}}$ of Markov kernels $M_t : \mathbb{X} \times \mathcal{X} \to [0, 1]$, an initial distribution $\eta_0 \in \mathsf{M}_1(\mathcal{X})$, and a sequence $\{g_t\}_{t\in\mathbb{N}}$

of bounded measurable potential functions $g_t : \mathbb{X} \to \mathbb{R}_+$, a sequence $\{\pi_{0:t}\}_{t\in\mathbb{N}}$ of *Feynman–Kac path measures* is defined by

$$\pi_{0:t} : \mathcal{X}^{\otimes t} \ni A \mapsto \frac{\gamma_{0:t}(A)}{\gamma_{0:t}(\mathbb{X}^t)}, \quad t \in \mathbb{N}, \tag{1.3}$$

where

$$\gamma_{0:t} : \mathcal{X}^{\otimes t} \ni A \mapsto \int \mathbb{1}_A(x_{0:t})\, \pi_0(\mathrm{d}x_0) \prod_{m=0}^{t-1} Q_m(x_m, \mathrm{d}x_{m+1}), \tag{1.4}$$

with

$$Q_m : \mathbb{X} \times \mathcal{X} \ni (x, A) \mapsto g_m(x) M_m(x, A) \tag{1.5}$$

being unnormalized kernels. By convention, $\pi_{0:0}{:=}\eta_0$. Note that each $\pi_{0:t}$ is a probability measure, whereas $\gamma_{0:t}$ is not normalized. For every $t \in \mathbb{N}^*$, we also define the marginal distribution $\pi_t : \mathcal{X} \ni A \mapsto \pi_{0:t}(\mathbb{X}^{\otimes t-1} \times A)$.

1.4.2.1 Particle Filtering

In most cases $\{\pi_m\}_{m\in\mathbb{N}}$ is intractable, but can be approximated by $\Pi_N(\boldsymbol{\xi}_m){:=}N^{-1}\sum_{i=1}\delta_{\xi_m^i}$ where for $m \in \mathbb{N}$, $\boldsymbol{\xi}_m = (\xi_m^1, \ldots, \xi_m^N)$, is a set of $N \in \mathbb{N}^*$ particles and each particle ξ_m^i is an $\mathbb{X}$-valued random variable. Such particle approximation is based on the recursion

$$\pi_{m+1} = \frac{\pi_m Q_m}{\pi_m g_m} = \frac{\int \pi_m(\mathrm{d}x) Q_m(x, \cdot)}{\int g_m(x)\pi_m(\mathrm{d}x)}\,. \tag{1.6}$$

By the recursion above, it is possible to obtain a new particle approximation of π_{m+1} from $\boldsymbol{\xi}_m$ by drawing new particles $\boldsymbol{\xi}_{m+1} = (\xi_{m+1}^1, \ldots, \xi_{m+1}^N)$ conditionally independently given $\boldsymbol{\xi}_m$ according to

$$\xi_{m+1}^i \sim \sum_{\ell=1}^{N} \frac{g_m(\xi_m^\ell)}{\sum_{\ell'=1}^N g_m(\xi_m^{\ell'})} M_m(\xi_m^\ell, \cdot), \quad i \in [\![1, N]\!].$$

Drawing ξ_{m+1}^i can be done by first *selecting* an ancestor according to the weights $g_m(\xi_m^\ell)/\sum_{\ell'=1}^N g_m(\xi_m^{\ell'})$ and then *updating* the selected ancestor through $M_m(\xi_m^\ell, \cdot)$. This procedure is called the *bootstrap particle filter* with multinomial resampling and it yields consistent approximations of π_m, in the sense that $\Pi_N(\boldsymbol{\xi}_m)h = N^{-1}\sum_{i=1}^N h(\xi_m^i)$ serves as a proxy for $\pi_m h$ for any π_m-integrable test function h. (Under general conditions, $N^{-1}\sum_{i=1}^N h(\xi_m^i)$ converges in probability to π_m as $N \to \infty$; see Del Moral (2004); Chopin and Papaspiliopoulos (2020), and the references therein.) We restrain our presentation to this resampling scheme, see Douc et al. (2005) for a comparison between different resampling schemes for the filtering problem.

Note that the particle filter builds an approximation of π_{m+1} using an approximation of π_m, which could lead to an accumulation of errors with m. A fundamental property of the particle filter is the *stability* w.r.t. the sequence length m. It can be shown, under general conditions, that the particle filter estimates converge to π_m *uniformly* w.r.t m, see Del Moral and Guionnet (2001); van Handel (2008); Whiteley (2013); Douc et al. (2014).

1.4.2.2 Particle Smoothing

We now focus on the problem of approximating the smoothing distribution $\pi_{0:m}$. It is possible to extend the procedure of the bootstrap particle filter to generate an approximation $\Pi_N(\boldsymbol{\xi}_{0:m}) = N^{-1}\sum_{i=1}^N \delta_{\xi_{0:m}^i}$ where for $m \in \mathbb{N}$, $\boldsymbol{\xi}_{0:m} = (\xi_{0:m}^1, \ldots, \xi_{0:m}^N)$, is a set of $N \in \mathbb{N}^*$ paths and each path $\xi_{0:m}^i$ is an $\mathbb{X}^{m+1}$-valued random variable. It is easy to see that (1.6) can be extended to $\pi_{0:m}$ therefore allowing the creation a new set of paths $\boldsymbol{\xi}_{0:m+1}$ that approximate $\pi_{0:m+1}$ by first selecting a *path* from $\boldsymbol{\xi}_{0:m}$ according to the

weights $g_m(\xi_m^\ell)/\sum_{\ell'=1}^N g_m(\xi_m^{\ell'})$ and then *updating* the path by concatenating a sample from $M_m(\xi_m^\ell, \cdot)$ to $\xi_{0:m}^\ell$ a new path to form $\boldsymbol{\xi}_{0:m+1}$.

The major drawback with the procedure above, known in the literature as the *poor man's smoother*, is that selecting a path from the previous paths leads to a collapse of the origins of the paths. Namely, since for $m > s$ the poor man's smoother keeps selecting a path that involves the particles ξ_s, the number of different elements at the sequence position s in $\boldsymbol{\xi}_{0:m}$ only decreases. It can actually be shown that the paths are expected to collapse after $m_N = \mathcal{O}(\log N)$, see Koskela et al. (2018). This renders the poor man's smoother unpractical when dealing with long sequences (large m).

Overcoming the path degeneracy: Backward decomposition based smoothers. To overcome the collapse of the poor man's smoother, current smoothing algorithms such as the *Forward Filtering Backward Simulation (FFBSi)* Godsill et al. (2004) and *particle-based, rapid incremental smoother (PaRIS)* Olsson and Westerborn (2017) rely on the backward decomposition of $\pi_{0:m}$. Let q_m denote the density of Q_m with respect to a given dominating measure λ. We define the backward kernel $\overleftarrow{Q}_{m,\lambda}$ as

$$\overleftarrow{Q}_{m,\lambda} : \mathbb{X} \times \mathcal{X} \ni (x_{m+1}, A) \mapsto \frac{\int \mathbb{1}_A(x_m) q_m(x_m, x_{m+1})\, \lambda(\mathrm{d}x_m)}{\int q_m(x'_m, x_{m+1})\, \lambda(\mathrm{d}x'_m)}\,. \tag{1.7}$$

and

$$B_m : \mathbb{X} \times \mathcal{X}^{\otimes m-1} \ni (x_m, A) \mapsto \int \cdots \int \mathbb{1}_A(x_{0:m-1}) \prod_{s=0}^{m-1} \overleftarrow{Q}_{s,\pi_s}(x_{s+1}, \mathrm{d}x_s), \tag{1.8}$$

Using $\{\overleftarrow{Q}_{s,\lambda}\}_{s\in[\![0,m]\!]}$ we obtain the *backward decomposition* Del Moral et al. (2010); Del Moral et al. (2016)

$$\pi_{0:m}(\mathrm{d}x_{0:m}) = \pi_m(\mathrm{d}x_m) \prod_{s=0}^{m-1} \overleftarrow{Q}_{s,\pi_s}(x_{s+1}, \mathrm{d}x_s)\,. \tag{1.9}$$

Namely one can obtain a path $\xi_{0:m} \sim \pi_{0:m}$ by starting from the filtering distribution $\xi_m \sim \pi_m$ and drawing backward $\xi_s|\xi_{s+1} \sim \overleftarrow{Q}_{s,\lambda}(\xi_{s+1}, \cdot)$ for $s \in [\![0, m-1]\!]$.

Using the backward decomposition, one can re-utilise the set of particle locations $\{\boldsymbol{\xi}_s\}_{s\in[\![0,m]\!]}$ produced by the particle filter in a subsequent (backward) sweep to sample N paths $\{\tilde{\xi}_{0:m}^i\}_{i=1}^N$. More precisely, given the forward particles $\{\boldsymbol{\xi}_s\}_{s=0}^m$, each path $\tilde{\xi}_{0:m}^i$ is generated by first drawing $\tilde{\xi}_m^i \sim \Pi_N(\boldsymbol{\xi}_m)$ and then drawing, recursively,

$$\tilde{\xi}_s^i \sim \overleftarrow{Q}_{s,\Pi_N(\boldsymbol{\xi}_s)}(\tilde{\xi}_{s+1}^i, \cdot) = \sum_{j=1}^N \frac{q_s(\xi_s^j, \tilde{\xi}_{s+1}^i)}{\sum_{\ell=1}^N q_s(\xi_s^\ell, \tilde{\xi}_{s+1}^i)} \delta_{\xi_s^j}\,; \tag{1.10}$$

that is, given $\tilde{\xi}_{s+1}^i$, $\tilde{\xi}_s^i$ is picked at random among $\boldsymbol{\xi}_s$ based on weights proportional to $\{q_s(\xi_s^j, \tilde{\xi}_{s+1}^i)\}_{j=1}^N$. This procedure constitutes the FFBSi smoother, with distribution $\pi_{0:m}^{\texttt{FFBSi}}$, which is no longer supported on the ancestor paths drawn during the particle filtering algorithm and avoids the trajectory degeneracy issue from the poor man's smoother. In this formulation, each backward-sampling operation (3.9) requires the computation of the normalising constant $\sum_{\ell=1}^N q_m(\xi_m^\ell, \tilde{\xi}_{m+1}^i)$, leading to an overall quadratic complexity of the algorithm. This can be eased by using an effective accept–reject technique, as proposed in Douc et al. (2011). One major drawback of the FFBSi is that the algorithm is essentially offline. In principle, if we want to compute an estimate of $\pi_{0:m}h$ for some test function h we need to draw N paths backward and then compute for each path $h(\xi_{0:m})$ to form the FFBSi estimate $\pi_{0:m}^{\texttt{FFBSi}}h = N^{-1}\sum_{i=1}^N h(\xi_{0:m}^i)$.

In the case of additive functionals $h_m(x_{0:m}) = \sum_{i=1}^m \tilde{h}_{i-1}(x_{i-1}, x_i)$, it is actually possible to render FFBSi online by establishing a recursion over the smoothing estimates $\pi_{0:m}^{\texttt{FFBSi}}h$ themselves (Del Moral et al. (2010)). Additive test functions are ubiquitous in smoothing in HMM, they appear notably during

the *E-step* of the *EM* algorithm (Cappé et al. (2005a)), and will also play an important role in Chapter 4. More specifically, using the forward decomposition $h_{m+1}(x_{0:m+1}) = h_m(x_{0:m}) + \tilde{h}_m(x_m, x_{m+1})$ and the backward kernel B_{m+1} defined in (1.8), we may write, for $x_{m+1} \in \mathbb{X}$,

$$\begin{aligned} B_{m+1}h_{m+1}(x_{m+1}) &= \int \overleftarrow{Q}_{m,\pi_m}(x_{m+1}, \mathrm{d}x_m) \int \left(h_m(x_{0:m}) + \tilde{h}_m(x_m, x_{m+1})\right) B_m(x_m, \mathrm{d}x_{0:m-1}) \\ &= \overleftarrow{Q}_{m,\pi_m}(B_m h_m + \tilde{h}_m)(x_{m+1}), \end{aligned} \tag{1.11}$$

which, by (1.9), implies that

$$\pi_{0:m+1}h_{m+1} = \pi_{m+1}\overleftarrow{Q}_{m,\pi_m}(B_m h_m + \tilde{h}_m). \tag{1.12}$$

The recursion above makes use of the filtering distributions $\{\pi_m\}_{m\in\mathbb{N}}$. Because they are generally intractable, we plug particle approximations $\Pi_N(\boldsymbol{\xi}_{m+1})$ and $\overleftarrow{Q}_{m,\Pi_N(\boldsymbol{\xi}_m)}$ of π_{m+1} and $\overleftarrow{Q}_{m,\pi_m}$, respectively, into recursion (1.12). More precisely, we proceed recursively, and assume that at time m, we have a sample $\{(\xi_m^i, \beta_m^i)\}_{i=1}^N$ of particles with associated statistics, where each statistic β_m^i serves as an approximation of $B_m h_m(\xi_m^i)$. Then, evolving the particle cloud according to the particle filtering algorithm and updating the statistics using (1.11), with $\overleftarrow{Q}_{m,\pi_m}$ replaced by $\overleftarrow{Q}_{m,\Pi_N(\boldsymbol{\xi}_m)}$, yields the particle-wise recursion

$$\beta_{m+1}^i = \sum_{\ell=1}^N \frac{q_m(\xi_m^\ell, \xi_{m+1}^i)}{\sum_{\ell'=1}^N q_m(\xi_m^{\ell'}, \xi_{m+1}^i)} \left(\beta_m^\ell + \tilde{h}_m(\xi_m^\ell, \xi_{m+1}^i)\right), \quad i \in [\![1, N]\!], \tag{1.13}$$

and, finally, the estimator $N^{-1}\sum_{i=1}^N \beta_m^i$ of $\pi_{0:m}h_m$, where we set $\boldsymbol{\beta}_m := (\beta_m^1, \ldots, \beta_m^N)$, for $i \in [\![1, N]\!]$. The procedure is initialized by simply letting $\beta_0^i = 0$, for all $i \in [\![1, N]\!]$. This algorithm is a special case of the *forward-filtering backward-smoothing* (`FFBSm`) *algorithm* (see Andrieu and Doucet (2003); Godsill et al. (2004); Douc et al. (2011); Särkkä (2013)) for additive functionals. It allows for online processing of the sequence $\{\pi_{0:m}h_m\}_{m\in\mathbb{N}}$, but also has the appealing property that only the current particles $\boldsymbol{\xi}_m$ and statistics $\boldsymbol{\beta}_m$ need to be stored in memory. However, because each update requires a summation of N terms, the scheme has an overall *quadratic* complexity in the number of particles, leading to a computational bottleneck in applications to complex models that require large particle sample sizes N.

To avoid the computational burden of this forward-only implementation of `FFBSm`, the `PARIS` algorithm Olsson and Westerborn (2017) updates the statistics $\boldsymbol{\beta}_m$ by replacing each sum (1.13) with the Monte Carlo estimate

$$\beta_{m+1}^i = \frac{1}{M}\sum_{j=1}^M \left(\tilde{\beta}_t^{i,j} + \tilde{h}_t(\tilde{\xi}_t^{i,j}, \xi_{t+1}^i)\right), \quad i \in [\![1, N]\!], \tag{1.14}$$

where

$$\{(\tilde{\xi}_m^{i,j}, \tilde{\beta}_m^{i,j})\}_{j=1}^M \sim \left(\sum_{\ell=1}^N \frac{q_m(\xi_m^\ell, \xi_{m+1}^i)}{\sum_{\ell'=1}^N q_m(\xi_m^{\ell'}, \xi_{m+1}^i)} \delta_{(\xi_m^\ell, \beta_m^\ell)}\right)^{\otimes M}, \quad i \in [\![1, N]\!].$$

Moreover, when the Markov transition densities of the model can be uniformly bounded, that is, there exists, for every $m \in \mathbb{N}$, an upper bound $\bar{\sigma}_m > 0$ such that for all $(x_m, x_{m+1}) \in \mathbb{X}^2$, $m_t(x_t, x_{t+1}) \leq \bar{\sigma}_t$ (a weak assumption satisfied for most models of interest), then we can generate a sample $(\tilde{\xi}_m^{i,j}, \beta_m^{i,j})$ by drawing, with replacement and until acceptance, candidates $(\tilde{\xi}_m^{i,*}, \tilde{\beta}_m^{i,*})$ from $\{(\xi_m^i, \beta_m^i)\}_{i=1}^N$ based on the normalized particle weights $\{g_m(\xi_m^\ell)/\sum_{\ell'=1}^N g_m(\xi_m^{\ell'})\}_{\ell=1}^N$ (obtained as a by-product in the generation of $\boldsymbol{\xi}_{m+1}$), and accepting the same with probability $m_m(\tilde{\xi}_m^{i,*}, \xi_{t+1}^i)/\bar{\sigma}_m$. Because this sampling procedure bypasses the calculation of the normalizing constant $\sum_{\ell'=1}^N q_m(\xi_m^{\ell'}, \xi_{m+1}^i)$ of the targeted categorical distribution, it yields an overall $\mathcal{O}(MN)$ complexity of the algorithm; see Douc et al. (2011) for details.

Increasing M improves the accuracy of the algorithm at the cost of additional computational complexity. As shown in Olsson and Westerborn (2017), there is a qualitative difference between the cases $M = 1$ and $M \geq 2$, and the latter is required to keep the PARIS numerically stable. More precisely, in the latter case, it can be shown that the PARIS estimator $N^{-1} \sum_{i=1}^{N} \beta_m^i$ satisfies, as N tends to infinity while M is held fixed, a central limit theorem (CLT) at the rate $\sqrt{N}$, with a t-normalized asymptotic variance of order $\mathcal{O}(1 - 1/(M-1))$. As it is clear from this bound, using a large M only wastes computational work, and setting M to two or three typically works well in practice.

Particle Gibbs. As for the importance sampling case, it is possible to define an ergodic sampler relying on the set of particle locations $\boldsymbol{\xi}_{0:m}$ that targets $\pi_{0:m}$. This procedure is known by the names *Particle Gibbs* (PG) or *conditional particle filter* (CPF) and is the sequential version of the iSIR algorithm Section 1.4.1.1. The general idea is to, at each step k, select a trajectory $(\zeta_0, \ldots, \zeta_t)$ from the set of particle locations $\boldsymbol{\xi}_{0:m}$ produced through an initial particle filtering algorithm and then insert (frozen) this path in the next particle filter iteration (i.e, $\zeta_i \in \boldsymbol{\xi}_i$ for all $i \in [\![0, m]\!]$). Formally, at iteration k with frozen path $\zeta_{0:m}[k] := (\zeta_0[k], \ldots, \zeta_t[k])$ we build a set of particle locations by defining, for $s \in [\![0, m]\!]$, $\boldsymbol{\xi}_s = (\zeta_s[k], \xi_1, \cdots, \xi_{N-1})$ where $(\xi_1, \cdots, \xi_{N-1})$ are drawn according to the particle filter update step on $\boldsymbol{\xi}_{s-1}$. From this new set of particle locations $\boldsymbol{\xi}_{0:m}$ and using the backward decomposition (1.9), we can draw a new path $\zeta_{0:m}[k+1]$. The procedure just described is called Particle Gibbs with backward sampling (PGBS) (Andrieu et al. (2010b)) and defines a Markov chain that converges geometrically fast to $\pi_{0:m}$ under standard strong mixing assumptions. We note though that there are other available options for defining a Particle Gibbs algorithm, such as the Particle Gibbs with Ancestor sampling (Lindsten et al. (2014b)), which for the bootstrap filter can be shown to be statistically equivalent to PGBS, see Lee et al. (2020).

(**Q3**) Can we generalize the results from Q 2? As the PGBS suffers from the same drawback as the iSIR procedure, computational waste, is it possible to recycle the particle clouds generated at each step of the PGBS while still achiving bias reduction?

1.5 Generative models

The task of generative modelling is to find for a given distribution of interest q_{data} defined over $\mathbb{R}^d$, a parametric function $f_\theta : \mathbb{R}^{d_s} \to \mathbb{R}^d$ that is able to push a noise distribution λ defined over $\mathbb{R}^{d_s}$ into a distribution that is "close" to q_{data}. In most applications, $\lambda = \mathcal{N}(0, \mathrm{I})$. More precisely, if we define for every Borelian set $A \subset \mathbb{R}^{d_s}$, the distribution $p_\theta(A) := \int \mathbb{1}_A(f_\theta(\epsilon))\lambda(\mathrm{d}\epsilon)$, the task of generative modelling is finding θ such that

$$\mathsf{q}_{\text{data}} \approx p_\theta \tag{1.15}$$

where $\approx$ means that the two distributions are close in terms of some statistical measure of dissimilarity, such as the Wasserstein distance or the Kullback-Leibler divergence. Here we focus on the case where we suppose that we have a dataset $\mathcal{D}$ of i.i.d samples from q_{data} that can be used to learn the parameter θ. We note, however, that some generative models require only access to a proxy of the density of q_{data} instead of a dataset $\mathcal{D}$. The ideal generative model would enable rapid sampling of diverse high quality samples and also tractable evaluation of the underlying density function. In recent years, several families of generative models have been introduced that rely on deep neural networks to construct f_θ. These generative models are called Deep generative models (DGM). Each family has its drawbacks. We present now a brief high-level introduction to the most well known families of DDGM and their known drawbacks, before focusing on a more detailed introduction to the so called Denoising Diffusion generative models (DDGM), which will appear in several chapters of this thesis.

1.5.1 Normalizing Flows

Normalizing flows are somehow the straighforward generative models. They rely on the fact that when we push forward a known distribution λ through a diffeomorphism $T : \mathbb{R}^{d_s} \to \mathbb{R}^{d_s}$, then the resulting distribution has a density with respect to the Lebesgue measure that is given by

$$p_T(x) = \lambda(T^{-1}(x))|J_{T^{-1}}(x)|\,, \tag{1.16}$$

where J_T is the Jacobian matrix of T. Therefore, if J_T is known, it is therefore easy to both sample p_T and to evaluate its density. Normalizing flows consist in stacking such diffeomorphisms $n \in \mathbb{N}$ times to form $f_\theta = T_{1,\theta_1} \circ \cdots \circ T_{n,\theta_n}$, where for each $i \in [\![1, n]\!]$, $T_{i,\theta_i} : \mathbb{R}^{d_s} \to \mathbb{R}^{d_s}$ is a diffeomorphism with easy to calculate inverse Jacobians $J_{T^{-1}_{i,\theta_i}}$. The density of p_θ with respect to the Lebesgue measure is

$$p_\theta(x) = \lambda(f_\theta^{-1}(x))|J_{f_\theta}^{-1}(x)|\,, \tag{1.17}$$

where $|J_{f_\theta}^{-1}(x)| = \prod_{i=1}^{n} |J_{T_{i,\theta}^{-1}}((T_{n,\theta}{}^{-1} \circ \cdots \circ T_{i+1,\theta}{}^{-1})(x))|$.

This fact makes also the training of normalizing flows particularly simple. Indeed, we can write

$$\mathsf{KL}(\mathsf{q}_{\text{data}} \parallel p_\theta) = -\int \log p_\theta(x)\mathsf{q}_{\text{data}}(\mathrm{d}x) + C\,, \tag{1.18}$$

which leads to the following optimization objective

$$\theta = \operatorname{argmin} -\mathbb{E}_{\mathsf{q}_{\text{data}}}\left[\log p_\theta(x)\right]\,. \tag{1.19}$$

It is then possible to use first-order gradient optimization objectives by calculating a Monte Carlo estimate of the gradient of the objective defined above using samples from q_{data}.

As previously said, the two main advantages of Normalizing Flows is that it is both easy to sample from and also easy to evaluate its log density. One of the main issues is that since f_θ is itself a diffeomorphism, the resulting distribution is defined over $f_\theta(\mathbb{R}^{d_s})$ which inherits from all the topological properties of $\mathbb{R}^d_s$, namely, being a connected d_s manifold. Therefore, if q_{data} represents a distribution with several non-connected modes, the resulting distribution p_θ would inevitably draw a linking path between the modes figure 1.4. Another problem is that, if we suppose that the distribution q_{data} is defined over a manifold of dimension $d < d_s$, then by the same reason we know that p_θ would not be able to accurately fit such distribution.

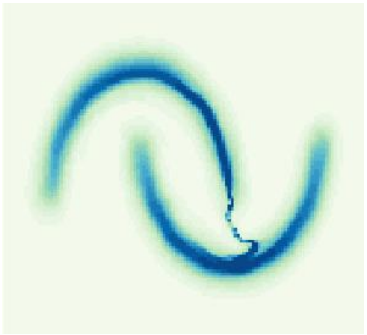


Figure 1.4: Example of Normalizing flow learned distribution for the Banana shaped dataset, taken from Figure 8 of Grenioux et al. (2023) .

1.5.2 Generative Adversarial Networks (GAN)

Generative Adversarial Networks (Goodfellow et al. (2014)) alleviate the diffeomorphism constraint by allowing $d_s \neq d$ and a much more flexible network architecture. This renders the evaluation of the density and log density of the ensuing distribution intricate. For $d_s < d$, the density w.r.t the Lebesgue measure is not defined.

To circumvent this issue, in Goodfellow et al. (2014) the generative models are trained through an adversarial training procedure, which involves the introduction of a second deep neural network $D_\phi : \mathbb{R}^{d_s} \to \{0, 1\}$, the discriminator network. The discriminator is trained to differentiate between samples from p_θ and q_{data}. At the same time, the generative model is trained to reduce the performance of the discriminator network. More precisely, GANs are trained according to the following objective

$$(\theta, \phi) := \operatorname{argmin}_\theta \operatorname{argmax}_\phi V(\theta, \phi) := \mathbb{E}_{\mathsf{q}_{\text{data}}} [\log D_\phi] + \mathbb{E}_{p_\theta} [\log(1 - D_\phi)] \, . \tag{1.20}$$

At first, it might seem challenging to compute gradients of $V(\theta, \phi)$ with respect to θ, but note that $\mathbb{E}_{p_\theta} [\log(1 - D_\phi)] = \mathbb{E}_{\epsilon \sim \lambda} [\log(1 - D_\phi(f_\theta(\epsilon)))]$. While GANs excel in quality of the generated samples and runtime speed and are the golden standard in generative model for most tasks, they suffer from some well known drawbacks, such as mode collapse and unstable training. Several strategies have been proposed (see Jabbar et al. (2021) and references therein) to mitigate both issues, but they are still a challenge to practitioners today.

1.5.3 Noise Conditional Score Networks (NCSN) inference by annealed Langevin dynamics

Before introducing the first version of DDGM in its modern form, introduced by Song et al. (2021c), we start by first describing the algorithm for Noise Conditional Score Networks (NCSN) inference by annealed Langevin dynamics, introduced in Song and Ermon (2019). NCSN lay the foundation of what would then become DDGM and is the first model without adversarial training to beat GANs in image generation tasks Song and Ermon (2019).

Before diving into the generative model defined in Song and Ermon (2019), we briefly introduce Unadjusted Langevin Algorithm (ULA) (Roberts and Tweedie, 1996) and Score matching Hyvärinen (2005). ULA consists of an algorithm that provides approximate samples of a distribution of interest q_{data} which admits a density with respect to the Lebesgue measure by exploiting the score of the distribution. The score is defined as the gradient of the density, i.e. $\nabla \log \mathsf{q}_{\text{data}}$. ULA defines a Markov chain $\{X_t\}_{t \in \mathbb{N}}$ by first sampling X_0 according to some initial distribution μ_0 and then defining for $t \in \mathbb{N}^*$

$$X_t := X_{t-1} + \gamma \nabla \log \mathsf{q}_{\text{data}}(X_{t-1}) + (2\gamma)^{1/2} \epsilon_t \, , \tag{1.21}$$

where $\epsilon_t \sim \mathcal{N}(0, \mathrm{I}_d)$ and γ is a positive constant called the step size. As shown in Durmus and Moulines (2017); Durmus et al. (2019), ULA provides samples that are arbitrarily close (in KL) to the q_{data} if one chooses γ small enough and runs the chain long enough. Notably, the amount of iterations of $\{X_t\}_{t \in \mathbb{N}}$ needed to obtain a given precision depends on how "far" μ_0 and q_{data} are.

Score matching Hyvärinen (2005) learns the score $\nabla \log \mathsf{q}_{\text{data}}$ by using i.i.d samples of q_{data} and without training a model to estimate the density of q_{data} first. To do so, it relies on a neural network s_θ that is trained to minimize

$$\theta \in \mathbb{R}^{\mathrm{d}_\theta} \to \mathbb{E}_{X \sim \mathsf{q}_{\text{data}}} \left[\operatorname{tr}(\nabla \mathsf{s}_\theta(X)) + (1/2) \|\mathsf{s}_\theta(X)\|^2 \right] \, , \tag{1.22}$$

which is shown in Hyvärinen (2005) to be equivalent to minimizing the Score matching loss

$$\theta \in \mathbb{R}^{\mathrm{d}_\theta} \to \mathbb{E}_{X \sim \mathsf{q}_{\text{data}}} \left[\|\mathsf{s}_\theta(X) - \nabla \log \mathsf{q}_{\text{data}}(X)\|^2 \right] \, . \tag{1.23}$$

There are two main drawbacks when learning the score via (1.22). The first being the cost of computing the terms of (1.22) at each iteration of the training procedure. Furthermore, Song and Ermon (2019) shows evidence that score matching with (1.22) fails to provide reliable estimates of the score in zones of low density, which might lead to the generation of spurious samples that do not reflect the underlying data density. We refer to (Song and Ermon, 2019, Section 3) for a detailed discussion.

To account for both problems, Song and Ermon (2019) proposes to first build a sequence of easier to sample laws $\{\mathsf{q}_t\}_{t\in[\![0,n]\!]}$, from which the respective scores $\mathsf{s}_{\theta,t}$ can be learned through Denoising score matching (DSN) and then sample from them using ULA sequentially. The sequence of laws is defined by convoluting the data distribution with a Gaussian kernel with increasing variance, namely $q_{t|0}(x_t|x_0) = \mathcal{N}(x_t; x_0, \upsilon_t^2 \mathrm{I}_d)$, where $\{\upsilon_t^2\}_{t\in[\![0,n]\!]}$ is an increasing positive sequence, i.e. $\mathsf{q}_t(x_t) := \int \mathsf{q}_{\mathrm{data}}(\mathrm{d}x_0) q_{t|0}(x_t|x_0)$. We denote the joint law $\mathsf{q}_{t,0}(\mathrm{d}x_t, \mathrm{d}x_0) := q_{t|0}(\mathrm{d}x_t|x_0)\mathsf{q}_{\mathrm{data}}(\mathrm{d}x_0)$. The score of q_t can be calculated through Fisher's identity

$$\nabla \log \mathsf{q}_t(x_t) = \mathbb{E}_{X_0\sim\mathsf{q}_{\mathrm{data}}}\left[\nabla \log q_{t|0}(X_0|x_t)\frac{q_{t|0}(x_t|X_0)}{\mathsf{q}_t(x_t)}\right]. \tag{1.24}$$

One can learn the score of q_t via a Neural network $\mathsf{s}_{\theta,t}$ by minimising

$$\mathbb{E}_{X_t\sim\mathsf{q}_t}\left[\|\mathsf{s}_{\theta,t}(X_t) - \nabla \log \mathsf{q}_t(X_t)\|^2\right],$$

which can be written

$$\begin{aligned}
&\mathbb{E}_{X_t\sim\mathsf{q}_t}\left[\left\|\mathsf{s}_\theta(X_t) - \mathbb{E}_{X_0\sim\mathsf{q}_{\mathrm{data}}}\left[\nabla \log q_{t|0}(x_t|X_0)\frac{q_{t|0}(X_t|X_0)}{\mathsf{q}_t(X_t)}\right]\right\|^2\right] \\
&\quad = \mathbb{E}_{X_t\sim\mathsf{q}_t}\left[\mathbb{E}_{X_0\sim\mathsf{q}_{\mathrm{data}}}\left[\left(\frac{q_{t|0}(X_t|X_0)}{\mathsf{q}_t(X_t)}\right)\left(\|\mathsf{s}_\theta(X_t)\|^2 - 2\mathsf{s}_\theta(X_t)^T\nabla \log q_{t|0}(X_t|X_0)\right)\right]\right] + C \\
&\quad = \mathbb{E}_{(X_t,X_0)\sim\mathsf{q}_{t,0}}\left[\|\mathsf{s}_\theta(X_t)\|^2 - 2\mathsf{s}_\theta(X_t)^T\nabla \log q_{t|0}(X_t|X_0)\right] + C \\
&\quad = \mathbb{E}_{(X_t,X_0)\sim\mathsf{q}_{t,0}}\left[\|\mathsf{s}_\theta(X_t) - \nabla \log q_{t|0}(X_t|X_0)\|^2\right] + \tilde{C},
\end{aligned}$$

where C and $\tilde{C}$ are constants that do not depend on θ. This procedure is called Denoising score matching (DSN) and has been introduced in Vincent (2011). Therefore, by noting that $\nabla \log q_{t|0}(x_t|x_0) = -\upsilon_t^{-2}(x_t - x_0)$ the score matching problem can be written as

$$\theta^* = \mathrm{argmin}_\theta\, \mathcal{L}_t(\theta) = \mathbb{E}_{(X_t,X_0)\sim\mathsf{q}_{t,0}}\left[\|\mathsf{s}_\theta(X_t) + \upsilon_t^{-2}(X_t - X_0)\|^2\right], \tag{1.25}$$

which corresponds to (Song and Ermon, 2019, equation 5). Instead of using one neural network for each t, NCSN consists in using a single network to learn all the scores by accepting as an input the level of noise, thus, the objective becomes

$$\mathcal{L}_{1:n}(\theta, \varrho_{1:n}) = \sum_{t=1}^n \varrho_t^2 \mathcal{L}_t(\theta) = \sum_{t=1}^n \varrho_t^2 \mathbb{E}_{(X_t,X_0)\sim\mathsf{q}_{t,0}}\left[\|\mathsf{s}_\theta(X_t, \upsilon_t) + \upsilon_t^{-2}(X_t - X_0)\|^2\right], \tag{1.26}$$

where $\{\varrho_t\}_{t\in[\![1,n]\!]} \in \mathbb{R}$ are a sequence of weights. In Song and Ermon (2019), they chose $\varrho_t = \upsilon_t$.

The sequence $\{\upsilon_t^2\}_{t\in[\![0,n]\!]}$ is designed to increase progressively from a small υ_1^2 to attain a relatively large υ_t^2. The reason is that υ_1^2 small would ensure the samples to be close to samples from $\mathsf{q}_{\mathrm{data}}$. Large υ_n^2 would increase the density in parts of the space of low density for $\mathsf{q}_{\mathrm{data}}$, potentially linking between two zones of high density for $\mathsf{q}_{\mathrm{data}}$ that previously were separated by low density regions and rendering exploration during ULA more efficient. Furthermore, the number of iterations needed for

ULA to provide samples that are close to the target distribution depends on the distance between q_t and q_{t+1}, which serves as the starting distribution for ULA targeting q_t. Therefore, one would ideally have $\upsilon_{t+1}^2 - \upsilon_t^2$ small to require less ULA iterations.

The full annealed Langevin algorithm using the scores $\{\mathsf{s}_\theta(\cdot, \upsilon_t)\}_{t\in[\![1,n]\!]}$ is given in Algorithm 1, and takes as input a starting sample X_n^0, the number of Langevin steps k, a multiplicative constant for the stepsize r. At time of its publication Song and Ermon (2019) achieved state of the art sample quality on unconditional CIFAR 10 generation, beating several different generative models, such as GANs and Normalizing flows. However, one of the drawbacks of such generative model is that the inference time is quite long. Indeed, for the algorithm to produce high quality samples, several Langevin steps k ($k = 100$) are needed with a small r ($r \approx 10^{-5}$) and $n = 10$, leading to 1000 Neural network evaluations (NNE) versus 1 NNE for GAN models.

Algorithm 1 NCSN algorithm

Data: X_n^0, k, r, θ
Result: X_0^0
for $t \leftarrow n$ **to** 1 **do**
  for $\ell \leftarrow 1$ **to** k **do**
    set $\gamma = r\upsilon_t^2/\upsilon_n^2$.
    draw $\epsilon_{t,\ell} \sim \mathcal{N}(0, \mathrm{I}_d)$.
    set $X_t^\ell = X_t^{\ell-1} + (\gamma/2)\mathsf{s}_\theta(X_t^{\ell-1}, \upsilon_t) + \gamma^{1/2}\epsilon_{t,\ell}$
  set $X_{t-1}^0 = X_t^\ell$.

1.5.4 Denoising Diffusion generative models (DDGM)

Denoising diffusion generative models (DDGM) target the same sequence of laws $\{\mathsf{q}_t\}_{t\in[\![1,n]\!]}$, but instead of relying on ULA to produce samples from q_t it builds a way of sampling of q_t *directly* from q_{t+1}. There are several formulations and variations of DDGM, relying on stochastic differential equations Song et al. (2021c), ordinary differential equations Karras et al. (2022) or Markov chains Song et al. (2021a). We follow the presentation of Song et al. (2021a) that yields the so called DDIM (denoising diffusion implicit model) sampler.

The building block for DDIM are the inference bridges $\{q_{t-1|t,0}^{\eta_{t-1}}(x_{t-1}|x_0, x_t)\}_{t=2}^n$, depending on a sequence $\{\eta_t\}_{t\in[\![1,n-1]\!]}$ of hyperparameters, defined as

$$q_{t-1|t,0}^{\eta_{t-1}}(x_{t-1}|x_t, x_0) := \mathcal{N}\left(x_{t-1}; \mu_{k-1}(x_0, x_t), \eta_{t-1}^2\,\mathrm{I}_d\right) \tag{1.27}$$

$$\mu_{t-1}(x_0, x_t) := x_0 + (\upsilon_{t-1}^2/\upsilon_t^2 - \eta_{t-1}^2/\upsilon_t^2)^{1/2}(x_t - x_0)\,. \tag{1.28}$$

The definitions above may at first seem artificial, but they are motivated by the following lemma.

Lemma 1 (Adapted from (Song et al., 2021a, Lemma 1, Appendix B)). *Let $t \in [\![2, n-1]\!]$ and $\eta^2 \in (0, \upsilon_{t-1}^2)$. Then,*

$$q_{t-1|0}^{\eta}(x_{t-1}|x_0) := \int q_{t|0}(\mathrm{d}x_t|x_0) q_{t-1|t,0}^{\eta}(x_{t-1}|x_t, x_0) = q_{t-1|0}(x_{t-1}|x_0)\,. \tag{1.29}$$

Define, for a given $\eta = \{\eta_t\}_{t\in[\![0,n]\!]}$ satisfying $\eta_t \in (0, \upsilon_t)$ for $t \in [\![1, n]\!]$ and inference process

$$q_{1:n|0}^{\eta}(\mathrm{d}x_{1:n}|x_0) = q_{n|0}(\mathrm{d}x_n|x_0) \prod_{t=2}^{n} q_{t-1|t,0}^{\eta_{t-1}}(\mathrm{d}x_{t-1}|x_t x_0) \tag{1.30}$$

$$q_{0:n}^{\eta}(\mathrm{d}x_{0:n}) = q_{1:n|0}^{\eta}(\mathrm{d}x_{1:n}|x_0)\mathsf{q}_{\mathrm{data}}(\mathrm{d}x_0)\,. \tag{1.31}$$

Lemma 1 implies that $q^{\eta}_{0:n}$ admits q_t as t marginals. Furthermore, Lemma 1 allows us to define

$$\mathsf{q}^{\eta_{t-1}}_{t-1|t} : \mathbb{R}^d \times \mathcal{B}(\mathbb{R}^d) \ni (x_t, A) \to \int \mathbb{1}_A(x_{t-1}) q^{\eta_{t-1}}_{t-1|t,0}(\mathrm{d}x_{t-1}|x_t, x_0) q_{0|t}(\mathrm{d}x_0|x_t)\,, \tag{1.32}$$

which satisfies

$$\mathsf{q}_{t-1}(\mathrm{d}x_{t-1}) = \int \mathsf{q}^{\eta_{t-1}}_{t-1|t}(\mathrm{d}x_{t-1}|x_t)\mathsf{q}_t(\mathrm{d}x_t)\,. \tag{1.33}$$

Even though (1.33) provides a way of passing from q_t to q_{t-1}, it involves an intractable kernel $\mathsf{q}^{\eta_{t-1}}_{t-1|t}$. By noting that $\mathbb{E}_{X_0 \sim q^{\eta}_{0|t}(\cdot|x_t)}\left[\nabla \log q_{t|0}(x_t|X_0)\right] = -(\mathbb{E}_{x_0 \sim q^{\eta}_{0|t}(\cdot|x_t)}[X_0] - x_t)/\upsilon_t^2$, one can obtain an estimate of $\mathbb{E}_{X_0 \sim q^{\eta}_{0|t}(\cdot|x_t)}[X_0]$ by

$$\mu_{t,\theta}(x_t) := x_t + \upsilon_t^2 \mathsf{s}_\theta(x_t, \upsilon_t)\,. \tag{1.34}$$

We use $\mu_{t,\theta}()$ as a replacement of the integral in (1.33) to define, for a given η_{t-1}

$$p^{\theta,\eta_{t-1}}_{t-1|t}(\mathrm{d}x_{t-1}|x_t) = q^{\eta_{t-1}}_{t-1|t,0}(\mathrm{d}x_{t-1}|x_t, \mu_{\theta,t}(x_t))\,. \tag{1.35}$$

We finally define the backward distribution

$$\mathsf{p}^{\theta,\eta}_{0:n}(\mathrm{d}x_{1:n}) := \lambda_n(\mathrm{d}x_n) \prod_{t=1}^{n} p^{\theta,\eta_{t-1}}_{t-1|t}(\mathrm{d}x_{t-1}|x_t)\,. \tag{1.36}$$

where $p^{\theta,\eta_0}_{0|1}(\cdot|x_1) = \mathcal{N}(\mu_{1,\theta}(x_1), \eta_0^2\,\mathrm{I})$ and $\lambda_n = \mathcal{N}(0, \upsilon_n^2\,\mathrm{I})$.

While we have motivated the backward distribution by replacing $\mathsf{q}^{\theta,\eta}_{t-1|t}$ by $p^{\theta,\eta}_{t-1|t}$, it can also be viewed as minimizing the Kullback-Leibler (KL) between the inference process (1.31) and the variational family defined by (1.36), which we denote $\mathcal{F}_{\mathrm{DDIM}}(\eta)$.

Defining $\mathsf{q}_{t,0}(\mathrm{d}(x_t, x_0)) := \mathsf{q}_{\mathrm{data}}(\mathrm{d}x_0) q_{t|0}(\mathrm{d}x_t|x_0)$, we can write

$$\begin{aligned}
\mathsf{KL}(q^{\eta}_{0:n} \parallel \mathsf{p}^{\theta}_{0:n}) &= \int \log\left(\frac{\mathsf{q}_{\mathrm{data}}(x_0) q_{n|0}(x_n|x_0) \prod_{t=2}^{n} q^{\eta}_{t-1|t,0}(x_{t-1}|x_t, x_0)}{\lambda_n(x_n) \prod_{t=1}^{n} p^{\theta}_{t-1|t}(x_{t-1}|x_t)}\right) q^{\eta}_{0:n}(\mathrm{d}x_{0:n}) \\
&= \int \log\left(\frac{\mathsf{q}_{\mathrm{data}}(x_0)}{p^{\theta}_{0|1}(x_0|x_1)}\right) \mathsf{q}_{0,1}(\mathrm{d}x_{0:1}) + \int \log\left(\frac{q_{n|0}(x_n|x_0)}{\lambda_n(x_n)}\right) \mathsf{q}_{n,0}(\mathrm{d}x_n, \mathrm{d}x_0) \\
&+ \sum_{t=2}^{n} \int \log\left(\frac{q^{\eta}_{t-1|t,0}(x_{t-1}|x_0, x_t)}{p^{\theta}_{t-1|t}(x_{t-1}|x_t)}\right) q^{\eta}_{t-1|t,0}(\mathrm{d}x_{t-1}|x_t, x_0) \mathsf{q}_{t,0}(\mathrm{d}x_t, \mathrm{d}x_0) \\
&= -\int \log p^{\theta}_{0|1}(x_0|x_1) \mathsf{q}_{1,0}(\mathrm{d}x_{0:1}) + \upsilon_n^{-2} \mathbb{E}_{\mathsf{q}_{\mathrm{data}}}\left[\|X_0\|^2\right] + \int \log \mathsf{q}_{\mathrm{data}}(x_0) \mathsf{q}_{\mathrm{data}}(\mathrm{d}x_0) \\
&+ \sum_{t=2}^{n} \mathbb{E}_{(X_t, X_0) \sim \mathsf{q}_{t,0}}\left[\mathsf{KL}(q^{\eta}_{t-1|t,0}(\cdot|X_t, X_0) \parallel p^{\theta}_{t-1|t}(\cdot|X_t))\right] \\
&= \frac{1}{2} \sum_{t=0}^{n-1} \tilde{\varrho}_t^2 \mathbb{E}_{(X_t, X_0) \sim \mathsf{q}_{t,0}}\left[\|\mu_{t,\theta}(X_t) - X_0\|^2\right] + \upsilon_n^{-2} \mathbb{E}_{\mathsf{q}_{\mathrm{data}}}\left[\|X_0\|^2\right] \\
&+ \int \log \mathsf{q}_{\mathrm{data}}(x_0) \mathsf{q}_{\mathrm{data}}(\mathrm{d}x_0) + d \log \eta_0 + \frac{d}{2} \log(2\pi)\,,
\end{aligned}$$

with $\tilde{\varrho}_{t-1} := \left[\upsilon_t - (\upsilon_{t-1}^2 - \eta_{t-1}^2)^{1/2}\right] (\eta_{t-1}\upsilon_t)^{-1}$ for $t \in [\![2, n]\!]$ and $\varrho_0 = \eta_0^{-1}$. Note that since $\|\mu_{t,\theta}(x_t) - x_0\|^2 = \upsilon_t^2 \|\mathsf{s}_\theta(x_t, \upsilon_t) - \nabla \log q_{t|0}(x_t|x_0)\|^2$, using (1.26) we can write

$$\mathsf{KL}(q^{\eta}_{0:n} \parallel \mathsf{p}^{\theta}_{0:n}) = \mathbb{E}_{X_1 \sim \mathsf{q}_1}\left[\mathsf{KL}(\mathsf{q}_{\mathrm{data}} \parallel p^{\theta}_{0|1}(\cdot|X_1))\right] + \mathcal{L}_{1:n}(\theta, \varrho_{1:n}) + \mathsf{KL}(\mathsf{q}_n \parallel \lambda_n)\,, \tag{1.37}$$

with $\varrho_{t-1} = \tilde{\varrho}_{t-1}\upsilon_t = \eta_{t-1}^{-1}\left[\upsilon_t - (\upsilon_{t-1}^2 - \eta_{t-1}^2)^{1/2}\right]$ for $t \in [\![2, n]\!]$ and $\varrho_0 = \eta_0^{-1}\upsilon_1$. This links the minimization of $\mathsf{KL}(q_{0:n}^{\eta} \parallel \mathsf{p}_{0:n}^{\theta})$ and the score matching objective defined in eq. (1.26) with this particular choice of $\varrho_{1:n}$. Note that to further minimize the $\mathsf{KL}(q_{0:n}^{\eta} \parallel \mathsf{p}_{0:n}^{\theta})$, we must choose $\upsilon_n \gg \mathbb{E}_{\mathsf{q}_{\text{data}}}\left[\|X_0\|^2\right]^{1/2}$. Note as well that one expects that by choosing υ_1 small, the loss term defined by $\mathbb{E}_{X_1 \sim \mathsf{q}_1}\left[\mathsf{KL}(\mathsf{q}_{\text{data}} \parallel p_{0|1}^{\theta}(\cdot|X_0))\right]$ should be easier to learn, as $\mathsf{q}_1 \approx \mathsf{q}_{\text{data}}$.

As shown in Song et al. (2021c) and Song et al. (2021a), the corresponding generative model is capable of generating high-quality samples. Note as well that the same minimum is shared over $\mathcal{F}_{\text{DDIM}}(s\eta)$ for every $s \in (0, 1)$, showing that once a model is trained with a fixed η, it is possible to reduce the variance of the backward kernels to $s\eta$ while still being sure of attaining the minimizer over $\mathcal{F}_{\text{DDIM}}(s\eta)$. In Song et al. (2021a), they show that this provides a tradeoff between sample quality and inference time. Namely by reducing η and skipping some of the backward kernels, one is able to obtain higher perceptual scores for image generation than one would by only skipping some of the backward kernels. This leads to reducing considerably the number of NNE to 10 with only a slight degradation of sample quality.

Convolutional Neural Networks and Denoising

The success of DDGM for image generation tasks relies in parts on the fact that Convolutional Neural networks, and especially the UNet Ronneberger et al. (2015) architecture, are extremely good in denoising tasks. In particular, in DDGM each backward kernel $p_{t-1|t}$ relies on the denoising of the state at iteration t to render $q_{t-1|t,0}^{\eta_{t-1}}$ Markovian. This section is heavily inspired by the work of Ulyanov et al. (2018), where they show that the structure of the UNet itself is a good prior for some inverse problems, such as denoising, super resolution and inpainting. We provide some numerical insights into why it is a good idea to train a neural network $\mu_\theta(\cdot)$ to minimize (1.37). Let x_0 be a natural image, x_s a realisation of $X_s = x_0 + (1/4)\varepsilon_1$ and x_t a realisation of $X_t = x_s + (1/4)\varepsilon_2$ where $\varepsilon_1, \varepsilon_2$ are i.i.d $\mathcal{N}(0, \mathrm{I}_d)$. In Ulyanov et al. (2018), they propose to use a UNet network $\mu_\theta(\cdot)$ to denoise x_t by solving $\operatorname{argmin}_\theta \|\mu_\theta(z) - x_t\|^2$, where z is a *fixed* white noise seed. Indeed, Ulyanov et al. (2018) show that by early stopping on this objective it is possible to generate realistic denoized images of x_t.

With this in mind, in this section we adapt this method to answer the following question What is easier for a UNet to predict, x_s from x_t or x_0 from x_t?

To answer this question, let again $\mu_\theta(\cdot)$ be a UNet. Consider the following losses:

$$L_{t|s}(\theta) := \|\mu_\theta(x_t) - x_s\|^2 \quad \text{and} \quad L_{t|0}(\theta) := \|\mu_\theta(x_t) - x_0\|^2 .$$

In Figure 1.5 we train two networks with the same initialization to minimize $L_{t|s}(\theta)$ (indicated by the circles) and $L_{t|0}(\theta)$ (indicated by stars) and track the values of each loss through the optimization. We see that in the beginning of the optimization, the output from the network is actually closer to x_0 than to x_s, even when trained over $L_{t|s}(\theta)$ which is coherent with Ulyanov et al. (2018). This shows that the task of predicting x_0 from x_t is actually simpler than predicting x_s from x_t.

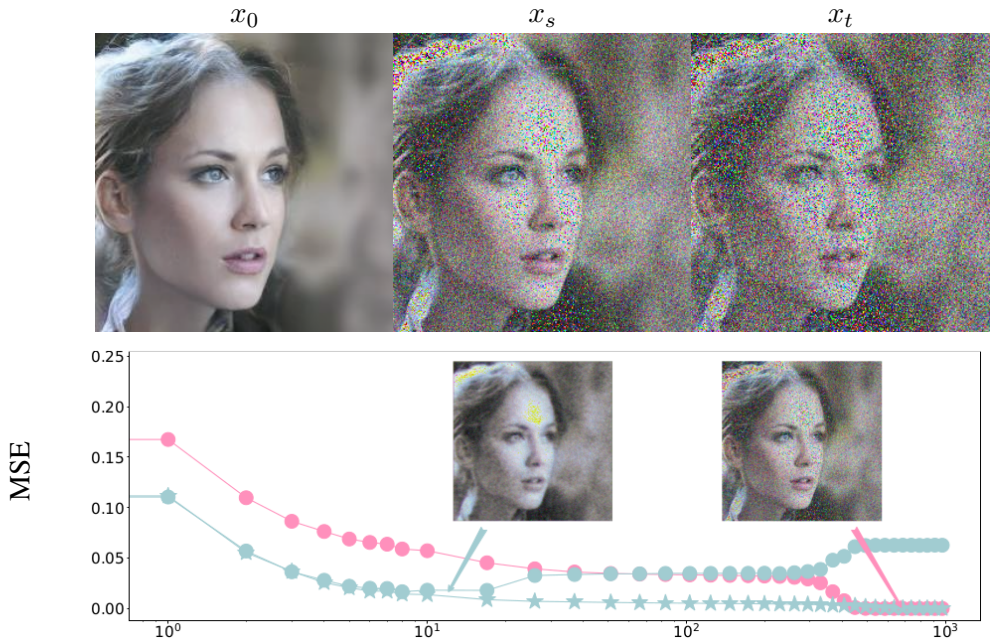


Figure 1.5: In the top row, we show the images used in the experiment (x_0, x_s, x_t) and in bottom row we show the evolution of $L_{t|s}(\theta)$ and $L_{t|0}(\theta)$ where the circle curves are obtained by minimizing $L_{t|s}(\theta)$ and the star curve by minimizing $L_{t|0}(\theta)$.

Another interesting consideration is that the last term in the optimization objective of a DDGM(1.25) is closely connected to the denoising objective defined above. Indeed, X_n is close to $\mathcal{N}(0, \upsilon_n^2 \mathrm{I})$ and therefore the term $\mathbb{E}_{X_0 \sim \mathsf{q}_{\mathrm{data}}, \epsilon \sim \mathcal{N}(0,\mathrm{I})} \left[\|\mu_{n,\theta}(X_0 + \upsilon_n \epsilon) - X_0\|^2 \right]$, is effectively trying to predict the image X_0 from $\upsilon_n \epsilon$. We show in Figure 1.6 examples of $\mu_{n,\theta}(\cdot)$ applied to Gaussian noise with υ_n standard deviation.

Figure 1.6: Example of the denoising from Gaussian noise using a DDGM. We use here the google/ddpm-ema-celebahq-256 *from the HuggingFaces diffusers library.*

DDGM as a prior for inverse problems

An interesting property of DDGM that distiguishes it from other generative models such as GANs is the iterative nature of the process of generating a sample. While the denoiser network is a complicated object, conditionally on x_t, $p_{t-1|t}$ is a well known distribution. This makes conditioning of such models easier, since it is possible to act over each $p_{t-1|t}$ separately in order to obtain a conditional sample through $p_{0|1}$. We drop the dependence on θ from the notation, since in this section we consider that we are given

a pre-trained DDGM.

This possibility opens a considerably rich field of research, particularly when using DDGM as priors for solving Bayesian inverse problems defined in Section 1.3.2. Several research works propose methods for sampling from the posterior distribution π when the prior distribution λ is a DDGM, such as Song et al. (2021a); Kawar et al. (2022); Lugmayr et al. (2022); Chung et al. (2023). The posterior distribution, as in (1.1), is $\mathsf{p}_0^y(x_0) \propto g_0^y(x_0)\mathsf{p}_0(x_0)$,where g_0^y is the likelihood function of the associated inverse problem. The posterior extended distribution is defined as

$$\mathsf{p}_{0:n}^y(\mathrm{d}x_{0:n}) \propto g_0^y(x_0)\lambda_n(\mathrm{d}x_n)\prod_{t=1}^{n} p_{t-1|t}(\mathrm{d}x_{t-1}|x_t)\,. \tag{1.38}$$

The t marginals of $\mathsf{p}_{0:n}^y$ are

$$\mathsf{p}_t^y(A) := \int \mathbb{1}_A(x_t)\mathsf{p}_{0:n}^y(\mathrm{d}x_{0:n}) = \int \mathbb{1}_A(x_t)g_0^y(x_0)p_{0|t}(\mathrm{d}x_0|x_t)\mathsf{p}_t(\mathrm{d}x_t) = \int \mathbb{1}_A(x_t)g_t^y(x_t)\mathsf{p}_t(\mathrm{d}x_t)\,, \tag{1.39}$$

where $g_t^y(x_t) := \int g_0^y(x_0)p_{0|t}(\mathrm{d}x_0|x_t)$. The score of the posterior can be written as $\nabla \log \mathsf{p}_t^y(x_t) = \nabla \log g_t^y(x_t) + \nabla \log p_{t|t+1}(x_t|x_{t+1})$. The current available methods to sample from p_0^y either try to approximate $\mathsf{p}_{0:n}^y$ by creating an alternative easier to sample version of $\mathsf{p}_{0:n}^y$, such as Song et al. (2021a); Kawar et al. (2022); Lugmayr et al. (2022) or try to approximate $\nabla \log g_t^y(x_t)$ such as Chung et al. (2023). All of those algorithms introduce irreducible approximation errors, leading to samplers that even though generate samples that are qualitatively appealing in some tasks, might have unexpected behaviours in other tasks. This lack of theoretical guarantees is specifically a problem when considering sensitive applications of such algorithms, as for example would be the case in applications to medical data.

(**Q4**) Is it possible to derive an algorithm for sampling from the posterior of a Bayesian inverse problem when using a DDGM as a prior that is theoretically grounded under realistic assumptions?

1.6 Contributions

The content of the present thesis is motivated by the research questions Q 1, 3 and 4 studied in the following papers which constitute the five remaining chapters of this document.

1. BR-SNIS: bias reduced self-normalized importance sampling (Cardoso et al., 2022c)
   Gabriel V Cardoso, Sergey Samsonov, Achille Thin, Eric Moulines, Jimmy Olsson.
   Advances in Neural Information Processing Systems 35 (NeurIPS) 2022.
2. Particle-based, rapid incremental smoother meets particle Gibbs (Cardoso et al., 2022a)
   Gabriel V. Cardoso, Jimmy Olsson, Eric Moulines
   Statistica Sinica.
3. State and parameter learning with the PaRIS particle Gibbs (Cardoso et al., 2023a).
   Gabriel V. Cardoso, Yazid Janati El Idrissi, Sylvain Le Corff, Eric Moulines, Jimmy Olsson.
   International Conference in Machine Learning 40 (ICML) (2023).
4. Monte Carlo guided Diffusion for Bayesian linear inverse problems (Cardoso et al., 2023b).
   Gabriel V. Cardoso, Yazid Janati El Idrissi, Sylvain Le Corff, Eric Moulines.
   Accepter for oral presentation ICLR 2024.
5. ECG Inpainting with denoising diffusion prior (Bedin et al., 2023).
   Lisa Bedin, **Gabriel V. Cardoso**, Remi Dubois, Eric Moulines
   Deep Generative Models for Health Workshop NeurIPS 2023.

6. Bayesian ECG reconstruction using denoising diffusion generative models (Cardoso et al., 2023c)
   Gabriel V. Cardoso, Lisa Bedin, Josselin Duchateau, Rémi Dubois, Eric Moulines.
   Under review.

While not present in this thesis, I have also co-authored the following conference papers:

- A Patient-Specific Single Equivalent Dipole Model. (Cardoso et al., 2022b)
  Gabriel V. Cardoso, Geneviève Robin, Andony Arrieula, Mark Potse, Michel Haïssaguerre, Eric Moulines, Rémi Dubois.
  2022 Computing in Cardiology (CinC). Vol. 498. IEEE, 2022.
- Generative methods for sampling transition paths in molecular dynamics. (Lelièvre et al., 2023)
  Tony Lelièvre, Geneviève Robin, Innas Sekkat, Gabriel Stoltz, **Gabriel V. Cardoso**.
  ESAIM: Proceedings and Surveys 73 (2023): 238-256.

Below, we provide a summary of the contributions made in each chapter. Please note that we introduce notations in each chapter, although there may be some overlap. These notations are always defined at the beginning of each chapter.

Chapter 2 / Q 2 - Bias Reduced Self Normalizing importance sampling

In this chapter, we analyse the so-called recycled i-SIR estimator described in Section 1.4.1.1 and show that under the same hypothesis that ensures geometric ergodicity of the chain of states $\{Y_k\}_{k\in\mathbb{N}}$, the SNIS estimations associated with the candidate pool chain $\{X_k^{1:N}\}_{k\in\mathbb{N}}$ have exponentially fast decaying bias. We derive MSE and concentration bounds for this estimator.

We propose a rollout estimator, that we furnish with bias, MSE and concentration bounds. Those bounds suggest a bias-variance trade off with respect to the number of burn-in steps k_0. We then propose a bootstrap procedure that allows to recover the variance loss with respect to the equivalent SNIS algorithm (SNIS with the same number of samples used in the whole procedure).

We show empirically in different datasets and applications the effect of bias-reduction without significantly increasing the variance of the proposed estimator. Furthermore, we show that in settings of limited budget, the proposed estimator yields estimations with smaller empirical bias then the zero-bias estimators proposed by Middleton et al. (2019).

Chapter 3 / Q 3 -PPG: Particle-based, Rapid Incremental Smoother Meets Particle Gibbs.

In this chapter, we extend the results concerning iSIR and importance sampling obtained in Chapter 2 to the case of the Particle Gibbs with backward sampling by merging the Particle Gibbs with Backward sampling algorithm with the `PARIS` algorithm. The proposed algorithm can be seen as a small modification over the `PARIS` algorithm and is able to generate a conditioning path $\zeta_{0:m}[k]$ (as in PGBS) and a sequence of $\boldsymbol{\beta}_m$ that approaches $\pi_{0:m}h_{0:m}$ for additive functionals $h_{0:m}$.

We show that the sequence of paths $\zeta_{0:m}[k]$ still have the same theoretical guarantees as the PGBS while achieving an exponential reduction of the bias of the estimator $N^{-1}\sum_{i=1}^{N}\beta_m^i$. We provide the resulting algorithm with an upper bound on the bias that decreases inversely proportional to the number N of particles and exponentially fast with the particle Gibbs iteration index k (under the assumption that the particle Gibbs sampler is uniformly ergodic). This is achieved while keeping the MSE comparable to that of the `PARIS` smoother. We provide numerical illustrations of our bounds in a Linear Gaussian state space model and in the non-linear stochastic volatility model.

Chapter 4 / Q 3 - Parameter learning with PPG.

Once we obtained the PPG algorithm in Chapter 3, we employ it in the context of score ascent, where we adapt the strategy of Karimi et al. (2019) to provide a non-asymptotic bound for the expectation of the squared gradient in terms of bias and MSE of the PPG. This bound establishes a $\mathcal{O}(\log(n)/\sqrt{n})$ convergence rate of the learning procedure which is explicit in the bias and MSE of the PPG estimator.

We show that the issuing optimization scheme is competitive in several numerical examples and performs better than approaches purely based on Particle Gibbs such as Lindholm and Lindsten (2018) in a same budget setting.

Chapter 5 / Q 4 - Monte Carlo guided Diffusion for Bayesian linear inverse problems.

In this chapter we consider the problem of sampling from the posterior of a DDGM model. We focus on the linear Gaussian inverse problem. Current methods Song et al. (2021a); Kawar et al. (2022); Lugmayr et al. (2022); Chung et al. (2023) aiming to sample from π, introduce an irreducible bias rendering them unreliable for critical applications. We propose a sequential Monte Carlo sampler that returns a consistent particle approximation of π, ensuring that asymptotically we sample from the target posterior. For this purpose we introduce a sequence of guiding potentials $\{g_s^y\}_{s=1}^n$ to the posterior distribution (1.38) that guide each marginal p_t to form $\mathsf{p}_t g_t^y$ while still admitting p_0^y as the 0-th marginal.

We construct the sequence of potentials first in the "noiseless" setting, i.e. $\sigma = 0$. We show that the general case ($\sigma > 0$) can be seen as a noiseless inverse problem on the extended states with prior $\mathsf{p}_{0:n}^\theta$. The derived SMC sampler targets the posterior and is provided with a non-asymptotic bound on the KL divergence between the target posterior and the *expected* particle approximation.

We show several examples (in high-dimension) for which the target posterior distribution is known evidence of our theoretical results, i.e. that the empirical distribution of samples from our algorithms converge to the target posterior distributions. By doing so, we also show that current "posterior sampling algorithms" do **not** sample from the target posterior, by generating a significant number of samples outside the support of the target posterior.

Chapter 6 / Q 1 - Bayesian ECG Reconstruction using MCG-DIFF.

In this chapter, we show how, by combining `MCGdiff` from Chapter 5 with a learned DDGM on ECG data, we are able to solve several different ECG reconstruction tasks better than the current methods without any fine-tuning required.

We show in particular that this tool can be valuable for solving anomaly detection on the ECG and show that it effectively distinguishes between the normal population and those that suffered a Miocardial Infarction. We also adapt `MCGdiff` to handle unknown measurement noise by coupling `MCGdiff` with a score ascent algorithm.

Bibliography

Adib, E., Afghah, F., and Prevost, J. J. (2022). Arrhythmia classification using cgan-augmented ecg signals*. *2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1865–1872.

Adib, E., Fernandez, A. S., Afghah, F., and Prevost, J. J. (2023). Synthetic ecg signal generation using probabilistic diffusion models. *IEEE Access*, 11:75818–75828.

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

Alcaraz, J. M. L. and Strodthoff, N. (2023). Diffusion-based conditional ecg generation with structured state space models. *Computers in Biology and Medicine*, 163:107115.

Aldous, D., Lovász, L., and Winkler, P. (1997). Mixing times for uniformly ergodic markov chains. *Stochastic Processes and their Applications*, 71(2):165–185.

Anderson, G. D. and Qiu, S.-L. (1997). A monotonicity property of the gamma function. *Proc. Amer. Math. Soc.*, 125(11):3355–3362.

Andrieu, C. (2016). On random-and systematic-scan samplers. *Biometrika*, 103(3):719–726.

Andrieu, C. and Doucet, A. (2002). Particle filtering for partially observed Gaussian state space models. *J. Roy. Statist. Soc. B*, 64(4):827–836.

Andrieu, C. and Doucet, A. (2003). Online Expectation–Maximization type algorithms for parameter estimation in general state space models. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, volume 6, pages 69–72.

Andrieu, C., Doucet, A., and Holenstein, R. (2010a). Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B*, 72(3):269–342.

Andrieu, C., Doucet, A., and Holenstein, R. (2010b). Particle Markov chain Monte Carlo methods (with discussion). *J. Roy. Statist. Soc. B*, 72:269–342.

Andrieu, C., Lee, A., and Vihola, M. (2018). Uniform ergodicity of the iterated conditional SMC and geometric ergodicity of particle Gibbs samplers. *Bernoulli*, 24(2):842–872.

Arjomand Bigdeli, S., Zwicker, M., Favaro, P., and Jin, M. (2017). Deep mean-shift priors for image restoration. *Advances in Neural Information Processing Systems*, 30.

Ball, R. L., Feiveson, A. H., Schlegel, T. T., Stare, V., and Dabney, A. R. (2014). Predicting "heart age" using electrocardiography. *Journal of personalized medicine*, 4(1):65–78.

Bazett, H. (1997). An analysis of the time-relations of electrocardiograms. *Annals of noninvasive electrocardiology*, 2(2):177–194.

Bedin, L., Cardoso, G., Dubois, R., and Moulines, E. (2023). ECG inpainting with denoising diffusion prior. In *Deep Generative Models for Health Workshop NeurIPS 2023*.

Benton, J., Shi, Y., De Bortoli, V., Deligiannidis, G., and Doucet, A. (2022). From denoising diffusions to denoising markov models. *arXiv preprint arXiv:2211.03595*.

Besag, J., York, J., and Mollié, A. (1991). Bayesian image restoration, with two applications in spatial statistics. *Annals of the institute of statistical mathematics*, 43:1–20.

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

Bonneel, N., Rabin, J., Peyré, G., and Pfister, H. (2015). Sliced and Radon Wasserstein Barycenters of Measures. *Journal of Mathematical Imaging and Vision*, 1(51):22–45.

Brammer, J. C. (2020). biopeaks: a graphical user interface for feature extraction from heart- and breathing biosignals. *Journal of Open Source Software*, 5(54):2621.

Burda, Y., Grosse, R., and Salakhutdinov, R. (2015). Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*.

Cappé, O. (2001). Recursive computation of smoothed functionals of hidden Markovian processes using a particle approximation. *Monte Carlo Methods Appl.*, 7(1–2):81–92.

Cappé, O. (2011). Online EM algorithm for hidden Markov models. *J. Comput. Graph. Statist.*, 20(3):728–749.

Cappé, O., Godsill, S. J., and Moulines, E. (2007). An overview of existing methods and recent advances in sequential Monte Carlo. *IEEE Proceedings*, 95(5):899–924.

Cappé, O., Moulines, E., and Rydén, T. (2005a). *Inference in Hidden Markov Models*. Springer.

Cappé, O., Moulines, E., and Ryden, T. (2005b). *Inference in Hidden Markov Models (Springer Series in Statistics)*. Springer-Verlag, Berlin, Heidelberg.

Cardoso, G., El Idrissi, Y. J., Le Corff, S., Moulines, É., and Olsson, J. (2023a). State and parameter learning with paris particle gibbs. In *International Conference on Machine Learning*, pages 3625–3675. PMLR.

Cardoso, G., Idrissi, Y. J. E., Corff, S. L., and Moulines, E. (2023b). Monte carlo guided diffusion for bayesian linear inverse problems.

Cardoso, G., Moulines, E., and Olsson, J. (2022a). Particle-based, rapid incremental smoother meets particle gibbs. *arXiv preprint arXiv:2209.10351*.

Cardoso, G., Robin, G., Arrieula, A., Potse, M., Haïssaguerre, M., Moulines, E., and Dubois, R. (2022b). A patient-specific single equivalent dipole model. In *2022 Computing in Cardiology (CinC)*, volume 498, pages 1–4. IEEE.

Cardoso, G., Samsonov, S., Thin, A., Moulines, E., and Olsson, J. (2022c). Br-snis: Bias reduced self-normalized importance sampling. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A., editors, *Advances in Neural Information Processing Systems*, volume 35, pages 716–729. Curran Associates, Inc.

Cardoso, G. V., Bedin, L., Duchateau, J., Dubois, R., and Moulines, E. (2023c). Bayesian ecg reconstruction using denoising diffusion generative models. *arXiv preprint arXiv:2401.05388*.

Chen, J., Lian, D., Jin, B., Huang, X., Zheng, K., and Chen, E. (2022). Fast variational autoencoder with inverted multi-index for collaborative filtering. In *Proceedings of the ACM Web Conference 2022*, pages 1944–1954.

Cheney, M. and Borden, B. (2009). *Fundamentals of radar imaging*. SIAM.

Chiang, H.-T., Hsieh, Y.-Y., Fu, S.-W., Hung, K.-H., Tsao, Y., and Chien, S.-Y. (2019). Noise reduction in ecg signals using fully convolutional denoising autoencoders. *IEEE Access*, 7:60806–60813.

Chopin, N. and Papaspiliopoulos, O. (2020). *An Introduction to Sequential Monte Carlo*, volume 4. Springer.

Chopin, N. and Singh, S. S. (2015). On particle Gibbs sampling. *Bernoulli*, 21(3):1855–1883.

Chung, H., Kim, J., Mccann, M. T., Klasky, M. L., and Ye, J. C. (2023). Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations*.

Ciosek, K., Fortuin, V., Tomioka, R., Hofmann, K., and Turner, R. E. (2020). Conservative uncertainty estimation by fitting prior networks. In *International Conference on Learning Representations*.

Dai, C., Heng, J., Jacob, P. E., and Whiteley, N. (2022). An invitation to sequential monte carlo samplers. *Journal of the American Statistical Association*, 117(539):1587–1600.

Dashti, M. and Stuart, A. M. (2017). The bayesian approach to inverse problems. In *Handbook of uncertainty quantification*, pages 311–428. Springer.

Del Moral, P. (2004). *Feynman-Kac Formulae. Genealogical and Interacting Particle Systems with Applications*. Springer.

Del Moral, P. (2013). *Mean Field Simulation for Monte Carlo Integration*. CRC Press.

Del Moral, P., Doucet, A., and Singh, S. S. (2010). A backward interpretation of Feynman–Kac formulae. *ESAIM: Mathematical Modelling and Numerical Analysis*, 44:947–975.

Del Moral, P. and Guionnet, A. (2001). On the stability of interacting processes with applications to filtering and genetic algorithms. *Annales de l'Institut Henri Poincaré*, 37:155–194.

Del Moral, P. and Jasra, A. (2018). A sharp first order analysis of Feynman–Kac particle models, part II: Particle Gibbs samplers. *Stoch. Proc. Appl.*, 128(1):354–371.

Del Moral, P., Kohn, R., and Patras, F. (2016). On particle Gibbs samplers. *Ann. Inst. H. Poincaré Probab. Statist.*, 52(4):1687–1733.

Dhariwal, P. and Nichol, A. (2021). Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. (2017). Density estimation using real NVP. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Douc, R., Cappé, O., and Moulines, E. (2005). Comparison of resampling schemes for particle filtering. In *4th International Symposium on Image and Signal Processing and Analysis (ISPA)*, Zagreb, Croatia. arXiv: cs.CE/0507025.

Douc, R., Garivier, A., Moulines, E., and Olsson, J. (2011). Sequential Monte Carlo smoothing for general state space hidden Markov models. *Ann. Appl. Probab.*, 21(6):1201–2145.

Douc, R. and Moulines, E. (2008). Limit theorems for weighted samples with applications to sequential Monte Carlo methods. *Ann. Statist.*, 36(5):2344–2376.

Douc, R., Moulines, E., Priouret, P., and Soulier, P. (2018). *Markov chains*. Springer Series in Operations Research and Financial Engineering. Springer, Cham.

Douc, R., Moulines, E., and Stoffer, D. (2014). *Nonlinear time series: Theory, methods and applications with R examples*. CRC press.

Doucet, A., De Freitas, N., Gordon, N. J., et al. (2001). *Sequential Monte Carlo methods in practice*, volume 1. Springer.

Durmus, A., Majewski, S., and Miasojedow, B. (2019). Analysis of langevin monte carlo via convex optimization. *Journal of Machine Learning Research*, 20(73):1–46.

Durmus, A. and Moulines, E. (2017). Nonasymptotic convergence analysis for the unadjusted Langevin algorithm. *The Annals of Applied Probability*, 27(3):1551–1587.

Elbakri, I. A. and Fessler, J. A. (2002). Statistical image reconstruction for polyenergetic x-ray computed tomography. *IEEE transactions on medical imaging*, 21(2):89–99.

Elvira, V. and Martino, L. (2021). Advances in importance sampling. *arXiv preprint arXiv:2102.05407*.

Fergus, R., Singh, B., Hertzmann, A., Roweis, S. T., and Freeman, W. T. (2006). Removing camera shake from a single photograph. In *Acm Siggraph 2006 Papers*, pages 787–794.

Figueiredo, M. A., Bioucas-Dias, J. M., and Nowak, R. D. (2007). Majorization–minimization algorithms for wavelet-based image restoration. *IEEE Transactions on Image processing*, 16(12):2980–2991.

Fort, G., Moulines, E., and Priouret, P. (2011). Convergence of adaptive and interacting markov chain monte carlo algorithms. *The Annals of Statistics*, 39(6).

Fridericia, L. (1921). Die systolendauer im elektrokardiogramm bei normalen menschen und bei herzkranken. *Acta Medica Scandinavica*, 54(1):17–50.

Genevay, A., Cuturi, M., Peyré, G., and Bach, F. (2016). Stochastic optimization for large-scale optimal transport. *Advances in neural information processing systems*, 29.

Gloaguen, P., Le Corff, S., and Olsson, J. (2022). A pseudo-marginal sequential Monte Carlo online smoothing algorithm. *Bernoulli*, 28(4):2606–2633.

Glynn, P. W. and Rhee, C.-H. (2014). Exact estimation for markov chain equilibrium expectations. *Journal of Applied Probability*, 51(A):377–389.

Godsill, S. J., Doucet, A., and West, M. (2004). Monte Carlo smoothing for non-linear time series. *J. Am. Statist. Assoc.*, 50:438–449.

Goldberger, A. L., Amaral, L. A. N., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C.-K., and Stanley, H. E. (2000 (June 13)). PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220. Circulation Electronic Pages: http://circ.ahajournals.org/content/101/23/e215.full PMID:1085218; doi: 10.1161/01.CIR.101.23.e215.

González, R. C., Woods, R. E., and Masters, B. R. (2009). Digital image processing, third edition. *Journal of Biomedical Optics*, 14:029901.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 2672–2680, Cambridge, MA, USA. MIT Press.

Gordon, N., Salmond, D., and Smith, A. F. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proc. F, Radar Signal Process.*, 140:107–113.

Grenioux, L., Oliviero Durmus, A., Moulines, E., and Gabrié, M. (2023). On sampling with approximate transport maps. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J., editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 11698–11733. PMLR.

Gui, J., Sun, Z., Wen, Y., Tao, D., and Ye, J. (2021). A review on generative adversarial networks: Algorithms, theory, and applications. *IEEE transactions on knowledge and data engineering*.

Haïssaguerre, M., Hocini, M., Cheniti, G., Duchateau, J., Sacher, F., Puyo, S., Cochet, H., Takigawa, M., Denis, A., Martin, R., et al. (2018). Localized structural alterations underlying a subset of unexplained sudden cardiac death. *Circulation: Arrhythmia and Electrophysiology*, 11(7):e006120.

Hesterberg, T. (1995). Weighted average importance sampling and defensive mixture distributions. *Technometrics*, 37(2):185–194.

Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851.

Hoffman, M. and Gelman, A. (2011). The no-u-turn sampler: Adaptively setting path lengths in hamiltonian monte carlo. *Journal of Machine Learning Research*, 15.

Hoffman, M. D., Gelman, A., et al. (2014). The no-U-turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *J. Mach. Learn. Res.*, 15(1):1593–1623.

Hsu, D., Kontorovich, A., Levin, D. A., Peres, Y., Szepesvári, C., and Wolfer, G. (2019). Mixing time estimation in reversible markov chains from a single sample path. *The Annals of Applied Probability*, 29(4):2439–2480.

Huggins, J. H. and Roy, D. M. (2019). Sequential Monte Carlo as approximate sampling: bounds, adaptive resampling via ∞-ESS, and an application to particle Gibbs. *Bernoulli*, 25(1):584 – 622.

Hull, J. and White, A. (1987). The pricing of options on assets with stochastic volatilities. *J. Finance*, 42:281–300.

Hyvärinen, A. (2005). Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(24):695–709.

Ibanez, B., James, S., Agewall, S., Antunes, M. J., Bucciarelli-Ducci, C., Bueno, H., Caforio, A. L. P., Crea, F., Goudevenos, J. A., Halvorsen, S., Hindricks, G., Kastrati, A., Lenzen, M. J., Prescott, E., Roffi, M., Valgimigli, M., Varenhorst, C., Vranckx, P., Widimský, P., and Group, E. S. D. (2017). 2017 ESC Guidelines for the management of acute myocardial infarction in patients presenting with ST-segment elevation: The Task Force for the management of acute myocardial infarction in patients presenting with ST-segment elevation of the European Society of Cardiology (ESC). *European Heart Journal*, 39(2):119–177.

Idier, J. (2013). *Bayesian approach to inverse problems*. John Wiley & Sons.

Ivanov, O., Figurnov, M., and Vetrov, D. (2018). Variational autoencoder with arbitrary conditioning. *arXiv preprint arXiv:1806.02382*.

Jabbar, A., Li, X., and Omar, B. (2021). A survey on generative adversarial networks: Variants, applications, and training. *ACM Computing Surveys (CSUR)*, 54(8):1–49.

Jacob, P. E., Lindsten, F., and Schön, T. B. (2020a). Smoothing with couplings of conditional particle filters. *J. Am. Statist. Assoc.*, 115(530):721–729.

Jacob, P. E., O'Leary, J., and Atchadé, Y. F. (2020b). Unbiased markov chain monte carlo methods with couplings. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 82(3):543–600.

Jameson, J. L., Fauci, A. S., Kasper, D. L., Hauser, S. L., Longo, D. L., and Loscalzo, J. (2018). McGraw-Hill Education, New York, NY.

Jarner, H., larsen, T. S., Krogh, A., Saxild, H. H., Brunak, S., and Knudsen, S. (2001). Sigma A recognition sites in the Bacilius subtilis genome. *Microbiology*, 147:2417–2424.

Kahn, H. and Marshall, A. W. (1953). Methods of reducing sample size in monte carlo computations. *Journal of the Operations Research Society of America*, 1(5):263–278.

Kaipio, J. P., Kolehmainen, V., Somersalo, E., and Vauhkonen, M. (2000). Statistical inversion and monte carlo sampling methods in electrical impedance tomography. *Inverse problems*, 16(5):1487.

Kaltenbach, S., Perdikaris, P., and Koutsourelakis, P.-S. (2023). Semi-supervised invertible neural operators for bayesian inverse problems. *Computational Mechanics*, pages 1–20.

Kang, J. and Wen, H. (2022). A Study on Several Critical Problems on Arrhythmia Detection using Varying-Dimensional Electrocardiography. *Physiological Measurement*, 43(6):064007.

Karbalayghareh, A., Qian, X., and Dougherty, E. R. (2018). Optimal bayesian transfer learning. *IEEE Transactions on Signal Processing*, 66(14):3724–3739.

Karimi, B., Miasojedow, B., Moulines, E., and Wai, H.-T. (2019). Non-asymptotic analysis of biased stochastic approximation scheme. In Beygelzimer, A. and Hsu, D., editors, *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pages 1944–1974. PMLR.

Karras, T., Aittala, M., Aila, T., and Laine, S. (2022). Elucidating the design space of diffusion-based generative models. In *Proc. NeurIPS*.

Kawar, B., Elad, M., Ermon, S., and Song, J. (2022). Denoising diffusion restoration models.

Kawar, B., Vaksman, G., and Elad, M. (2021). Snips: Solving noisy inverse problems stochastically. *Advances in Neural Information Processing Systems*, 34:21757–21769.

Kingma, D. P. and Ba, J. (2015a). Adam: A method for stochastic optimization. In *ICLR 2015*.

Kingma, D. P. and Ba, J. (2015b). Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Kingma, D. P. and Welling, M. (2014). Stochastic gradient vb and the variational auto-encoder. In *Second International Conference on Learning Representations, ICLR*, volume 19, page 121.

Kingma, D. P., Welling, M., et al. (2019). An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392.

Kobyzev, I., Prince, S., and Brubaker, M. (2020). Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):3964–3979.

Koskela, J., Jenkins, P. A., Johansen, A. M., and Spanò, D. (2018). Asymptotic genealogies of interacting particle systems with an application to sequential monte carlo. *arXiv: Statistics Theory*.

Kroese, D. P. and Rubinstein, R. Y. (2012). Monte carlo methods. *Wiley Interdisciplinary Reviews: Computational Statistics*, 4(1):48–58.

Kuzborskij, I., Vernade, C., Gyorgy, A., and Szepesvári, C. (2021). Confident off-policy evaluation and selection through self-normalized importance weighting. In *International Conference on Artificial Intelligence and Statistics*, pages 640–648. PMLR.

Lamberti, R., Petetin, Y., Septier, F., and Desbouvries, F. (2018). A double proposal normalized importance sampling estimator. In *2018 IEEE Statistical Signal Processing Workshop (SSP)*, pages 238–242. IEEE.

Lawson, J., Tucker, G., Dai, B., and Ranganath, R. (2019). Energy-inspired models: Learning with sampler-induced distributions. *Advances in Neural Information Processing Systems*, 32.

Lee, A. (2011). *On auxiliary variables and many-core architectures in computational statistics*. PhD thesis, University of Oxford.

Lee, A., Singh, S. S., and Vihola, M. (2020). Coupled conditional backward sampling particle filter. *Ann. Statist.*, 48(5):3066–3089.

Lee, A., Yau, C., Giles, M. B., Doucet, A., and Holmes, C. C. (2010). On the utility of graphics cards to perform massively parallel simulation of advanced Monte Carlo methods. *Journal of computational and graphical statistics*, 19(4):769–789.

Lelièvre, T., Robin, G., Sekkat, I., Stoltz, G., and Cardoso, G. V. (2023). Generative methods for sampling transition paths in molecular dynamics. *ESAIM: Proceedings and Surveys*, 73:238–256.

Li, H., Ditzler, G., Roveda, J., and Li, A. (2023). Descod-ecg: Deep score-based diffusion model for ecg baseline wander and noise removal. *IEEE Journal of Biomedical and Health Informatics*, pages 1–11.

Lindholm, A. and Lindsten, F. (2018). Learning dynamical systems with particle stochastic approximation em.

Lindsten, F., Douc, R., and Moulines, E. (2015). Uniform ergodicity of the particle gibbs sampler. *Scandinavian Journal of Statistics*, 42(3):775–797.

Lindsten, F., Jordan, M. I., and Schön, T. B. (2014a). Particle Gibbs with ancestor sampling. *J. Mach. Learn. Res.*, 15(1):2145–2184.

Lindsten, F., Jordan, M. I., and Schön, T. B. (2014b). Particle gibbs with ancestor sampling.

Liu, J., Wong, W., and Kong, A. (1994). Covariance structure of the Gibbs sampler with applications to the comparisons of estimators and augmentation schemes. *Biometrika*, 81(1):27–40.

Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., and Van Gool, L. (2022). Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11461–11471.

Macfarlane, P. W., Van Oosterom, A., Pahlm, O., Kligfield, P., Janse, M., and Camm, J. (2010). *Comprehensive electrocardiology*. Springer Science & Business Media.

Maddouri, O., Qian, X., Alexander, F. J., Dougherty, E. R., and Yoon, B.-J. (2022). Robust importance sampling for error estimation in the context of optimal bayesian transfer learning. *Patterns*, page 100428.

Malik, M., Hnatkova, K., Kowalski, D., Keirns, J. J., and van Gelderen, E. M. (2013). Qt/rr curvatures in healthy subjects: sex differences and covariates. *American Journal of Physiology-Heart and Circulatory Physiology*, 305(12):H1798–H1806.

Marnissi, Y., Zheng, Y., Chouzenoux, E., and Pesquet, J.-C. (2017). A variational bayesian approach for image restoration—application to image deblurring with poisson–gaussian noise. *IEEE Transactions on Computational Imaging*, 3(4):722–737.

Metelli, A. M., Papini, M., Faccio, F., and Restelli, M. (2018). Policy optimization via importance sampling. *Advances in Neural Information Processing Systems*, 31.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equations of state calculations by fast computing machines. *J. Chem. Phys.*, 21:1087–1092.

Michelot, T., Langrock, R., and Patterson, T. A. (2016). movehmm: an r package for the statistical modelling of animal movement data using hidden markov models. *Methods in Ecology and Evolution*, 7.

Middleton, L., Deligiannidis, G., Doucet, A., and Jacob, P. E. (2019). Unbiased smoothing using particle independent metropolis-hastings. In Chaudhuri, K. and Sugiyama, M., editors, *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 2378–2387. PMLR.

Naesseth, C., Lindsten, F., and Blei, D. (2020). Markovian score climbing: Variational inference with kl (p|| q). *Advances in Neural Information Processing Systems*, 33:15499–15510.

Naesseth, C. A., Lindsten, F., Schön, T. B., et al. (2019). Elements of sequential monte carlo. *Foundations and Trends® in Machine Learning*, 12(3):307–392.

Neal, R. M. et al. (2011). Mcmc using hamiltonian dynamics. *Handbook of markov chain monte carlo*, 2(11):2.

Niknejad, M., Bioucas-Dias, J., and Figueiredo, M. A. (2019). External patch-based image restoration using importance sampling. *IEEE Transactions on Image Processing*, 28(9):4460–4470.

Olsson, J. and Westerborn, J. (2017). Efficient particle-based online smoothing in general hidden Markov models: The PaRIS algorithm. *Bernoulli*, 23(3):1951–1996.

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. (2021). Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64.

Paulin, D. (2015). Concentration inequalities for markov chains by marton couplings and spectral methods. *Electronic Journal of Probability*, 20:1–32.

Peng, J., Liu, D., Xu, S., and Li, H. (2021). Generating diverse structure for image inpainting with hierarchical vq-vae. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10775–10784.

Pitt, M. K. and Shephard, N. (1999). Filtering via simulation: Auxiliary particle filters. *J. Am. Statist. Assoc.*, 94(446):590–599.

Poyiadjis, G., Doucet, A., and Singh, S. S. (2005). Particle methods for optimal filter derivative: application to parameter estimation. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, pages v/925–v/928.

Poyiadjis, G., Doucet, A., and Singh, S. S. (2011). Particle approximations of the score and observed information matrix in state space models with application to parameter estimation. *Biometrika*, 98(1):65–80.

Reed, G. W., Rossi, J. E., and Cannon, C. P. (2017). Acute myocardial infarction. *The Lancet*, 389(10065):197–210.

Reyna, M. A., Sadr, N., Alday, E. A. P., Gu, A., Shah, A. J., Robichaux, C., Rad, A. B., Elola, A., Seyedi, S., Ansari, S., Ghanbari, H., Li, Q., Sharma, A., and Clifford, G. D. (2021). Will two do? varying dimensions in electrocardiography: The physionet/computing in cardiology challenge 2021. In *2021 Computing in Cardiology (CinC)*, volume 48, pages 1–4.

Reyna, M. A., Sadr, N., Alday, E. A. P., Gu, A. P., Shah, A. J., Robichaux, C., Rad, A. B., Andoni, Elola, Seyedi, S., Ansari, S., Ghanbari, H., Qiao, Li, Sharma, A., and Clifford, G. D. (2022). Issues in the automated classification of multilead ecgs using heterogeneous labels and populations. *Physiological Measurement*, 43.

Ribeiro, A. H., Ribeiro, M. H., Paixão, G. M., Oliveira, D. M., Gomes, P. R., Canazart, J. A., Ferreira, M. P., Andersson, C. R., Macfarlane, P. W., Meira Jr, W., et al. (2020). Automatic diagnosis of the 12-lead ecg using a deep neural network. *Nature communications*, 11(1):1760.

Robert, C. P. and Casella, G. (2004). *Monte Carlo Statistical Methods*. Springer, New York, 2nd edition.

Roberts, G. O. and Tweedie, R. L. (1996). Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli*, 2(4):341–363.

Robertson, A. W., Kirshner, S., and Smyth, P. (2004). Downscaling of daily rainfall occurrence over northeast brazil using a hidden markov model. *Journal of Climate*, 17:4407–4424.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer.

Rubin, D. B. (1987a). The calculation of posterior distributions by data augmentation: Comment: A noniterative sampling/importance resampling alternative to the data augmentation algorithm for creating a few imputations when fractions of missing information are modest: The sir algorithm. *Journal of the American Statistical Association*, 82(398):543–546.

Rubin, D. B. (1987b). Comment: A noniterative Sampling/Importance Resampling alternative to the data augmentation algorithm for creating a few imputations when fractions of missing information are modest: The SIR algorithm. *Journal of the American Statistical Association*, 82(398):542–543.

Sagie, A., Larson, M. G., Goldberg, R. J., Bengtson, J. R., and Levy, D. (1992). An improved method for adjusting the qt interval for heart rate (the framingham heart study). *The American journal of cardiology*, 70(7):797–801.

Sahlström, T. and Tarvainen, T. (2023). Utilizing variational autoencoders in the bayesian inverse problem of photoacoustic tomography. *SIAM Journal on Imaging Sciences*, 16(1):89–110.

Salakhutdinov, R. and Murray, I. (2008). On the quantitative analysis of deep belief networks. In *Proceedings of the 25th international conference on Machine learning*, pages 872–879.

Salama, G. and Bett, G. C. (2014). Sex differences in the mechanisms underlying long qt syndrome. *American Journal of Physiology-Heart and Circulatory Physiology*, 307(5):H640–H648.

Särkkä, S. (2013). *Bayesian Filtering and Smoothing*. Cambridge University Press.

Schwedes, T. and Calderhead, B. (2021). Rao-blackwellised parallel mcmc. In Banerjee, A. and Fukumizu, K., editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 3448–3456. PMLR.

Shan, L., Li, Y., Jiang, H., Zhou, P., Niu, J., Liu, R., Wei, Y., Peng, J., Yu, H., Sha, X., and Chang, S. (2022). Abnormal ecg detection based on an adversarial autoencoder. *Frontiers in Physiology*, 13.

Shihab, H. A., Gough, J., Cooper, D. N., Stenson, P. D., Barker, G., Edwards, K. J., Day, I. N. M., and Gaunt, T. R. (2012). Predicting the functional, molecular, and phenotypic consequences of amino acid substitutions using hidden markov models. *Human Mutation*, 34:57 – 65.

Shin, H. and Choi, M. (2023). Physics-informed variational inference for uncertainty quantification of stochastic differential equations. *Journal of Computational Physics*, page 112183.

Singh, P. and Pradhan, G. (2020). A new ecg denoising framework using generative adversarial network. *IEEE/ACM transactions on computational biology and bioinformatics*, 18(2):759–764.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR.

Song, J., Meng, C., and Ermon, S. (2021a). Denoising diffusion implicit models. In *International Conference on Learning Representations*.

Song, Y., Durkan, C., Murray, I., and Ermon, S. (2021b). Maximum likelihood training of score-based diffusion models. *Advances in Neural Information Processing Systems*, 34:1415–1428.

Song, Y. and Ermon, S. (2019). Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32.

Song, Y., Shen, L., Xing, L., and Ermon, S. (2022). Solving inverse problems in medical imaging with score-based generative models. In *International Conference on Learning Representations*.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2021c). Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*.

Stuart, A. M. (2010). Inverse problems: a bayesian perspective. *Acta numerica*, 19:451–559.

Su, J., Xu, B., and Yin, H. (2022). A survey of deep learning approaches to image restoration. *Neurocomputing*, 487:46–65.

Swaminathan, A. and Joachims, T. (2015). The self-normalized estimator for counterfactual learning. *advances in neural information processing systems*, 28.

Tadić, V. B. and Doucet, A. (2017). Asymptotic bias of stochastic gradient search. *The Annals of Applied Probability*, 27(6):3255 – 3304.

Thin, A., Janati El Idrissi, Y., Le Corff, S., Ollion, C., Moulines, E., Doucet, A., Durmus, A., and Robert, C. X. (2021). Neo: Non equilibrium sampling on the orbits of a deterministic transform. *Advances in Neural Information Processing Systems*, 34:17060–17071.

Tjelmeland, H. (2004a). Using all Metropolis–Hastings proposals to estimate mean values. Technical report.

Tjelmeland, H. (2004b). Using all metropolis-hastings proposals to estimate mean values.

Trippe, B. L., Yim, J., Tischer, D., Baker, D., Broderick, T., Barzilay, R., and Jaakkola, T. S. (2023). Diffusion probabilistic modeling of protein backbones in 3d for the motif-scaffolding problem. In *The Eleventh International Conference on Learning Representations*.

Ulyanov, D., Vedaldi, A., and Lempitsky, V. (2018). Deep image prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

van Handel, R. (2008). Uniform time average consistency of monte carlo particle filters. *Stochastic Processes and their Applications*, 119:3835–3861.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Vershynin, R. (2018). *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press.

Vincent, P. (2011). A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674.

Vlaardingerbroek, M. T. and Boer, J. A. (2013). *Magnetic resonance imaging: theory and practice*. Springer Science & Business Media.

Wainwright, M. J. (2019). *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press.

Wan, Z., Zhang, J., Chen, D., and Liao, J. (2021). High-fidelity pluralistic image completion with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4692–4701.

Wei, X., van Gorp, H., Gonzalez-Carabarin, L., Freedman, D., Eldar, Y. C., and van Sloun, R. J. (2022). Deep unfolding with normalizing flow priors for inverse problems. *IEEE Transactions on Signal Processing*, 70:2962–2971.

Wen, H. and Kang, J. (2021). Hybrid Arrhythmia Detection on Varying-Dimensional Electrocardiography: Combining Deep Neural Networks and Clinical Rules. In *2021 Computing in Cardiology (CinC)*. IEEE.

Whiteley, N. (2010). Discussion on particle Markov chain Monte Carlo methods. *J. Roy. Statist. Soc. B*, 72(3):306–307.

Whiteley, N. (2013). Stability properties of some particle filters. *The Annals of Applied Probability*, pages 2500–2537.

Wu, L., Trippe, B. L., Naesseth, C. A., Blei, D. M., and Cunningham, J. P. (2023). Practical and asymptotically exact conditional sampling in diffusion models.

Xiang, H., Zou, Q., Nawaz, M. A., Huang, X., Zhang, F., and Yu, H. (2023). Deep learning for image inpainting: A survey. *Pattern Recognition*, 134:109046.

Yeh, R. A., Lim, T. Y., Chen, C., Schwing, A. G., Hasegawa-Johnson, M., and Do, M. N. (2018). Image restoration with deep generative models. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6772–6776. IEEE.

Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., and Huang, T. S. (2018). Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5505–5514.

Zeng, Y., Fu, J., Chao, H., and Guo, B. (2022). Aggregated contextual transformations for high-resolution image inpainting. *IEEE Transactions on Visualization and Computer Graphics*.

Zhang, G., Ji, J., Zhang, Y., Yu, M., Jaakkola, T., and Chang, S. (2023). Towards coherent image inpainting using denoising diffusion implicit models. *arXiv preprint arXiv:2304.03322*.

Zhao, Y., Nassar, J., Jordan, I., Bugallo, M., and Park, I. M. (2021). Streaming variational monte carlo.

Zheng, C., Cham, T.-J., and Cai, J. (2019). Pluralistic image completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1438–1447.

Zhihang, X., Yingzhi, X., and Qifeng, L. (2023). A domain-decomposed vae method for bayesian inverse problems. *arXiv preprint arXiv:2301.05708*.

Chapter 2

BR-SNIS: Bias Reduced Self-Normalized Importance Sampling

2.1 Introduction

Background and previous work: *Importance sampling* Kahn and Marshall (1953); Agapiou et al. (2017) (IS) is a classical Monte Carlo technique for estimating expectations under some given probability distribution (the *target*) on the basis of a sample of draws from a different distribution (the *proposal*). In the modern era of artificial intelligence and statistical machine learning, characterized by large computational resources and Bayesian inference, IS technologies are enjoying a revival; see, *e.g.*, Niknejad et al. (2019); Kuzborskij et al. (2021) and Elvira and Martino (2021) for a recent survey. The method is not only relevant to situations where sampling from the target is intractable; it can also be used to achieve variance reduction Lamberti et al. (2018). When the proposal is dominating the target—in the sense that the support of the latter is contained in the support of the former—unbiased estimation can be achieved by assigning each draw an *importance weight* given by the likelihood ratio between the target and the proposal. In the very common case where the target is known only up to a normalizing constant, consistent estimation can still be achieved by simply normalizing each importance weight by the total weight of the sample; however, since such *self-normalized importance sampling* (SNIS) involves ratios of random variables, the procedure can only be implemented at the cost of bias, which can be significant in some applications.

More precisely, let $(\mathbb{X}, \mathcal{X})$ be some state space and $\pi(\mathrm{d}x) \propto w(x)\lambda(\mathrm{d}x)$ a given target probability distribution, where w and λ are a positive weight function and a proposal probability distribution on $(\mathbb{X}, \mathcal{X})$, respectively, such that the normalizing constant $\lambda(w) = \int w(x)\lambda(\mathrm{d}x)$ (this will be our generic notation for Lebesgue integrals) of π is finite. The SNIS estimator is given by

$$\Pi_M f(X^{1:M}) = \sum_{i=1}^M \omega_M^i f(X^i), \qquad \omega_M^i = w(X^i)/\sum_{\ell=1}^M w(X^\ell) \tag{2.1}$$

where $X^{1:M} = (X^1, \ldots, X^M)$ are independent draws from λ, and can be used to approximate $\pi(f) = \int f(x)\pi(\mathrm{d}x)$ for any test function f such that $\pi(|f|) < \infty$. The estimator (2.1) can be calculated without knowledge of the normalizing constant $\lambda(w)$, which is intractable in general.

The SNIS estimator is known to be biased; provided that $\lambda(w^2) < \infty$, the bias and mean-squared error (MSE) of the SNIS estimator (2.1) over bounded test functions f satisfying $\|f\|_\infty \leq 1$ are given respectively (see (Agapiou et al., 2017, Theorem 2.1)) by

$$|\mathbb{E}[\Pi_M f(X^{1:M})] - \pi(f)| \leq (12/M)\kappa[\pi, \lambda], \quad \mathbb{E}[\{\Pi_M f(X^{1:M}) - \pi(f)\}^2] \leq (4/M)\kappa[\pi, \lambda], \tag{2.2}$$

where $\kappa[\pi, \lambda] = \lambda(w^2)/\lambda^2(w)$. Although IS is primarily intended to approximate integrals in the form

$\pi(f)$, it can also be used to generate unweighted samples being approximately distributed according to π. In this paper, we consider *iterated sampling importance resampling* (i-SIR), proposed in Tjelmeland (2004a); see (Andrieu et al., 2010a; Lee et al., 2010; Lee, 2011; Andrieu et al., 2018). The i-SIR can be seen as an iterative application of the *sampling importance resampling* (SISR) algorithm proposed by Rubin (1987b); the k-th iteration is defined as follows. Given a state $Y_k \in \mathbb{X}$, (i) set $X_{k+1}^1 = Y_k$ and draw $X_{k+1}^{2:N}$ independently from the proposal distribution λ; (ii) compute, for $i \in \{1, \dots, N\}$, the normalized importance weights $\omega_{N,k+1}^i = w(X_{k+1}^i) / \sum_{\ell=1}^N w(X_{k+1}^\ell)$; (iii) select Y_{k+1} from the set $X_{k+1}^{1:N}$ by choosing X_{k+1}^i with probability $\omega_{N,k+1}^i$. In the following, Y_{k+1} and $X_{k+1}^{1:N}$ will be referred to as the *state* and the *candidate pool*, respectively. Following (Tjelmeland, 2004a) (see Section 2.2.1), i-SIR may be viewed (up to an irrelevant permutation of the samples) as a two-stage Gibbs sampler targeting an extended probability distribution $\boldsymbol{\varphi}_N$ on an enlarged state space including the state as well as the candidate pool. As this extended distribution allows π as a marginal with respect to the state, one can expect the marginal distribution of the generated states $(Y_k)_{k\in\mathbb{N}}$, forming themselves a Markov chain, to approach the target π of interest as k tends to infinity.

This paper: In i-SIR, the only function of the candidate pool is to guide the states selected at stage (iii) towards the target. Thus, since all rejected candidates are discarded, the approach results generally in a large waste of computational work. Thus, in the present paper we propose to recycle *all* the generated samples by incorporating all the proposed candidates $X_k^{1:N}$ into the estimator rather than only the selected candidate Y_k. We proceed in three steps. First, we show that under the stationary distribution $\boldsymbol{\varphi}_N$ of the process $(Y_k, X_k^{1:N})_{k\in\mathbb{N}}$ generated by i-SIR, the expectation of $\Pi_N f(X_k^{1:N})$ (given by (2.1)) equals $\pi(f)$ for every valid test function f (see Theorem 3). Second, we establish that since i-SIR is nothing but a systematic-scan Gibbs sampler, the two processes $(X_k^{1:N})_{k\in\mathbb{N}}$ and $(Y_k)_{k\in\mathbb{N}}$ are *interleaving* (see Theorem 6); thus, if $(Y_k)_{k\in\mathbb{N}}$ is uniformly geometrically ergodic, so is $(X_k^{1:N})_{k\in\mathbb{N}}$ with the same mixing rate κ_N. Third, as the main result of the present paper, we establish a novel $\mathcal{O}(\kappa_N^k/N)$ bound on the bias of the estimator $\Pi_N f(X_k^{1:N})$ (see Theorem 4), where the exponentially diminishing factor κ_N^k indicates a drastic bias reduction *vis-à-vis* the standard IS estimator (2.1) based on i.i.d. samples. As a consequence, approximating $\pi(f)$ by the average of $(\Pi_N f(X_k^{1:N}))_{\ell=k_0+1}^k$, where the "burn-in" period k_0 should be chosen proportionally to the mixing time of the process, yields an estimator whose bias can be furnished with a bound which is, roughly, proportional to $\kappa_N^{k_0}$ and inversely proportional to the total number $M = kN$ of samples generated in the algorithm (see Theorem 5). To complete the theoretical analysis of these estimators, we also equip the same with variance bounds. The procedure of recycling, as described above, all the samples generated in the i-SIR and to incorporate, at negligible computational cost, the same into the final estimator, will from now on be referred as BR-SNIS. Finally, we test numerically the proposed estimators and illustrate how a significant bias reduction relatively to the standard i-SIR can be obtained at basically no cost.

To sum up, our contribution is twofold, since we

- propose a new algorithm, BR-SNIS, which makes better use of the available computational resources by recycling the candidate pool generated at each iteration of i-SIR.
- furnish the proposed algorithm with rigorous theoretical results, including novel bias, variance, and high-probability bounds which support our claim that sample recycling may lead to drastic bias reduction without impairing the variance.

2.2 Main results

2.2.1 Statements

The i-SIR algorithm can be interpreted as a systematic-scan two-stage Gibbs sampler, alternately sampling from the full conditions of an extended target $\boldsymbol{\varphi}_N$ on the product space of states and candidate pools. Once the extended target $\boldsymbol{\varphi}_N$ is properly defined, these full conditionals can be retrieved from a dual representation of $\boldsymbol{\varphi}_N$ presented in Theorem 2. In order to define $\boldsymbol{\varphi}_N$, we introduce the Markov kernel (see Section A.1.1 for comments)

$$\boldsymbol{\Lambda}_N(y, \mathrm{d}x^{1:N}) = N^{-1} \sum_{i=1}^N \delta_y(\mathrm{d}x^i) \prod_{j \neq i} \lambda(\mathrm{d}x^j) \tag{2.3}$$

on $\mathbb{X} \times \mathcal{X}^{\otimes N}$, which describes probabilistically the sampling operation (i) in i-SIR. Using the kernel $\boldsymbol{\Lambda}_N$ we may now define properly the extended target $\boldsymbol{\varphi}_N$ as the probability law

$$\boldsymbol{\varphi}_N(\mathrm{d}(y, x^{1:N})) = \pi(\mathrm{d}y)\boldsymbol{\Lambda}_N(y, \mathrm{d}x^{1:N}) = N^{-1} \sum_{i=1}^N \pi(\mathrm{d}y)\delta_y(\mathrm{d}x^i) \prod_{j \neq i} \lambda(\mathrm{d}x^j) \tag{2.4}$$

on $(\mathbb{X}^{N+1}, \mathcal{X}^{\otimes(N+1)})$. Note that since for every $A \in \mathcal{X}$, $\boldsymbol{\varphi}_N(\mathbb{1}_{A \times \mathbb{X}^N}) = \pi(A)$, the target π coincides with the marginal of $\boldsymbol{\varphi}_N$ with respect to the state. Moreover, it is easily seen that $\boldsymbol{\Lambda}_N$ provides the conditional distribution, under $\boldsymbol{\varphi}_N$, of the candidate pool given the state. Defining the kernels

$$\Gamma_N(x^{1:N}, \mathrm{d}y) = N^{-1} \sum_{i=1}^N w(x^i)\delta_{x^i}(\mathrm{d}y), \quad \Pi_N(x^{1:N}, \mathrm{d}y) = \Gamma_N(x^{1:N}, \mathrm{d}y)/\Gamma_N \mathbb{1}_{\mathbb{X}}(x^{1:N}) \tag{2.5}$$

on $\mathbb{X} \times \mathcal{X}^{\otimes N}$, the marginal distribution $\boldsymbol{\pi}_N$ of $\boldsymbol{\varphi}_N$ with respect to $x^{1:N}$ is given by

$$\boldsymbol{\pi}_N(\mathrm{d}x^{1:N}) = \lambda(w)^{-1}\Gamma_N \mathbb{1}_{\mathbb{X}}(x^{1:N}) \prod_{j=1}^N \lambda(\mathrm{d}x^j). \tag{2.6}$$

It is interesting to note that the marginal $\boldsymbol{\pi}_N$ has a probability density function, proportional to $\Gamma_N \mathbb{1}_{\mathbb{X}}(x^{1:N}) = \sum_{i=1}^N w(x^i)/N$, with respect to the product measure $\lambda^{\otimes N}$. Using (2.6), we immediately obtain the following result.

Theorem 2 (duality of extended target)**.** *For every $N \in \mathbb{N}^*$,*

$$\boldsymbol{\varphi}_N(\mathrm{d}(y, x^{1:N})) = \pi(\mathrm{d}y)\boldsymbol{\Lambda}_N(y, \mathrm{d}x^{1:N}) = \boldsymbol{\pi}_N(\mathrm{d}x^{1:N})\Pi_N(x^{1:N}, \mathrm{d}y). \tag{2.7}$$

Note that the second identity of the dual representation (2.7) provides also the conditional distribution, under $\boldsymbol{\varphi}_N$, of the state given the candidates. Consequently, i-SIR is a systematic scan two-stage Gibbs sampler which generates a Markov chain $(X_k, Y_k)_{k \in \mathbb{N}}$ with time-homogeneous Markov kernel

$$\mathbf{P}_N((y_k, x_k^{1:N}), \mathrm{d}(y_{k+1}, x_{k+1}^{1:N})) = \boldsymbol{\Lambda}_N(y_k, \mathrm{d}x_{k+1}^{1:N})\Pi_N(x_{k+1}^{1:N}, \mathrm{d}y_{k+1}) \tag{2.8}$$

on $\mathbb{X}^{N+1} \times \mathcal{X}^{\otimes(N+1)}$. Note that the law $\mathbf{P}_N(y_k, x_k^{1:N}, \cdot)$ does not depend on $x_k^{1:N}$, which means that only the state Y_k needs to be stored from one iteration to the other. Thus, $(Y_k)_{k \in \mathbb{N}}$ is a Markov chain with Markov transition kernel

$$\mathsf{P}_N(y_k, \mathrm{d}y_{k+1}) = \int \boldsymbol{\Lambda}_N(y_k, \mathrm{d}x_{k+1}^{1:N})\Pi_N(x_{k+1}^{1:N}, \mathrm{d}y_{k+1}) = \boldsymbol{\Lambda}_N\Pi_N(y_k, \mathrm{d}y_{k+1}) \tag{2.9}$$

(where integration is w.r.t. $x_{k+1}^{1:N}$) on $\mathbb{X} \times \mathcal{X}$. The kernel (2.9) was analyzed in Andrieu et al. (2018). Given some probability distribution $\boldsymbol{\xi}$ on $(\mathbb{X}^{N+1}, \mathcal{X}^{\otimes(N+1)})$, we denote by $\mathbb{P}_{\boldsymbol{\xi}}$ the law of the canonical Markov chain $(X_k, Y_k)_{k \in \mathbb{N}}$ with kernel $\mathbf{P}_N$ and initial distribution $\boldsymbol{\xi}$. Our first results establishes the unbiasedness of the estimator $\Pi_N f(X^{1:N})$ under $\boldsymbol{\varphi}_N$.

Theorem 3. *For every $N \in \mathbb{N}^*$ and π-integrable function f,*

$$\int \Pi_N f(x^{1:N})\boldsymbol{\pi}_N(\mathrm{d}x^{1:N}) = \pi(f).$$

The proof of Theorem 3 is postponed to Section A.1.3. Next, we present theoretical bounds on the discrepancy, in terms of bias, MSE and covariance, between $\Pi_N f(X_k^{1:N})$ and $\pi(f)$, for bounded target functions f, when the i-SIR chain is initialized according to an arbitrary distribution $\boldsymbol{\xi}$. We will work under the following assumption.

A1. *It holds that* $\omega = \|w\|_\infty / \lambda(w) < \infty$.

Under **A** 1, the state and candidate-pool Markov chains $(Y_k)_{k\in\mathbb{N}}$ and $(X^{1:N})_{k\in\mathbb{N}}$ can be shown to be uniformly geometrically ergodic with mixing rate and mixing-time upper bound

$$\kappa_N = (2\omega - 1)/(2\omega + N - 2), \quad \tau_{mix,N} = \lceil -\ln 4 / \ln \kappa_N \rceil, \tag{2.10}$$

respectively; see Theorem 7 below for details. Here the mixing time $\tau_{mix,N}$ grows logarithmically with the sample size N. The exact value of $\tau_{mix,N}$ is likely to be grossly pessimistic, but we conjecture that the logarithmic dependence in the minibatch size holds true. In addition, under **A** 1 we define the constants

$$\begin{aligned} \varsigma^{bias} &= 4(\kappa[\pi,\lambda] + 1 + \omega) \\ \varsigma_i^{mse} &= 4(\kappa[\pi,\lambda]\mathbb{1}_{\{0,1\}}(i) + (1+\omega)^2 \mathbb{1}_{\{1,2\}}(i)), \quad \varsigma_i^{cov} = \varsigma^{bias}(\varsigma_i^{mse})^{1/2}, \quad i \in \{0,1,2\}. \end{aligned} \tag{2.11}$$

With these definitions, the following holds true.

Theorem 4. *Assume A1. Then for every initial distribution* $\boldsymbol{\xi}$ *on* $(\mathbb{X}^{N+1}, \mathcal{X}^{\otimes(N+1)})$*, bounded measurable function* f *on* $(\mathbb{X}, \mathcal{X})$ *such that* $\|f\|_\infty \leq 1$*,* $N \geq 2$*, and* $(k, \ell) \in (\mathbb{N}^*)^2$*,*

(i) $\left|\mathbb{E}_{\boldsymbol{\xi}}[\Pi_N f(X_k^{1:N})] - \pi(f)\right| \leq \varsigma^{bias}(N-1)^{-1}\kappa_N^{k-1}$,
(ii) $\mathbb{E}_{\boldsymbol{\xi}}[\{\Pi_N f(X_k^{1:N}) - \pi(f)\}^2] \leq \sum_{i=0}^2 \varsigma_i^{mse}(N-1)^{-1-i/2}$,
(iii) $\left|\mathbb{E}_{\boldsymbol{\xi}}[\{\Pi_N f(X_k^{1:N}) - \pi(f)\}\{\Pi_N f(X_{k+\ell}^{1:N}) - \pi(f)\}]\right| \leq \kappa_N^{\ell-1} \sum_{i=0}^2 \varsigma_i^{cov}(N-1)^{-(3-i/2)/2}$,

where constants are given in (2.10) *and* (2.11)*.*

It is worth noting that the bias decreases inversely with the number of candidates and exponentially with the number of iterations (the mixing time of the chain also depends on N). The MSE is also inversely proportional to the number of candidates N. In the light of the previous results, it is natural to consider an estimator formed by an average across the IS estimators $(\Pi_N f(X_k^{1:N}))_{k\in\mathbb{N}}$ associated with the candidate pools generated at the different i-SIR iterations. To mitigate the bias, we remove a "burn-in" period whose length k_0 should be chosen proportional to the mixing time $\tau_{mix,N}$ of the Markov chain $(Y_k)_{k\in\mathbb{N}}$ (which turns out to coincide with that or the chain $(X_k^{1:N})_{k\in\mathbb{N}}$; see Section 2.2.2). This yields the estimator

$$\Pi_{(k_0,k),N}(f) = (k-k_0)^{-1} \sum_{\ell=k_0+1}^k \Pi_N f(X_\ell^{1:N}) \tag{2.12}$$

of $\pi(f)$. The total number of samples (generated by the proposal λ) underlying this estimator is $M = (N-1)k$. Importantly, all the importance weights included in the estimators are obtained as a by-product of the i-SIR schedule; thus, it is, for a given budget of simulations (*i.e.*, under the constraint that $(k-k_0)N$ is constant), possible to compute $\Pi_{(k_0,k),N}(f)$ for different values of k_0, k and N with a negligible computational cost. We denote by $\upsilon = (k-k_0)/k$ the ratio of the number of candidate pools used in the estimator to the total number of sampled such pools. Note that this type of estimator was already suggested by Tjelmeland (2004b) and also appears in Schwedes and Calderhead (2021).

Our final main result provides bounds on the bias and the MSE of the estimator (2.12) as well as a high-probability bound for the same. Define $\zeta^{bias} = 4\tau_{mix,N}\varsigma^{bias}/3$, $\zeta_i^{mse} = \varsigma_{(i+1)\wedge 2}^{mse}\mathbb{1}_{\{0,2\}}(i) + (8/3)\tau_{mix,N}\varsigma_i^{cov}$, $i \in \{0,1,2\}$, $\zeta^{mse} = \zeta_0^{mse} + \zeta_1^{mse}(N-1)^{-1/4} + \zeta_2^{mse}(N-1)^{-1}$, and $\mathsf{MSE}_M^{is} = (4/M)\kappa[\pi,\lambda]$, see (2.2).

Theorem 5. *Assume A1. Then the following holds true for every initial distribution* $\boldsymbol{\xi}$ *on* $(\mathbb{X}^{N+1}, \mathcal{X}^{\otimes(N+1)})$*, bounded measurable function* f *on* $(\mathbb{X}, \mathcal{X})$ *such that* $\|f\|_\infty \leq 1$*, and* $N \geq 2$*.*

(i) $\left|\mathbb{E}_{\boldsymbol{\xi}}[\Pi_{(k_0,k),N}(f)] - \pi(f)\right| \leq \zeta^{bias}(\upsilon M)^{-1}4^{-k_0/\tau_{mix,N}}$

(ii) $\mathbb{E}_{\boldsymbol{\xi}}[\{\Pi_{(k_0,k),N}(f) - \pi(f)\}^2] \leq \mathsf{MSE}^{is}_{\upsilon M} + \zeta^{mse}(\upsilon M)^{-1}(N-1)^{-1/2}$

(iii) For every $\delta \in (0,1)$, $|\Pi_{(k_0,k),N}(f) - \pi(f)| \leq \varsigma^{hpd}(\upsilon M)^{-1/2}(\log(4/\delta))^{1/2}$ *with probability at least* $1-\delta$, *where* $\varsigma^{hpd} = 664\omega$.

Bootstrap: As established in Theorem 5, the bias of the BR-SNIS estimator decreases exponentially with the burn-in period k_0, leading to potentially significant bias reduction with respect to SNIS. Still, using a large k_0 is done at a price of increased overall MSE (mainly through the term $\mathsf{MSE}^{is}_{\upsilon M}$ in Theorem 5(ii), which is directly related to k_0 via υ). A natural way to reduce the variance is to use bootstrap. More precisely, we first apply a random permutation to the samples and re-compute BR-SNIS on the basis of the bootstrapped samples. After this, we produce a final estimator by averaging over the bootstrapped BR-SNIS replicates. In most applications, the major computational bottleneck consists of sampling from λ and evaluating w and f at the samples; thus, the additional operations that this bootstrap approach entails are computationally cheap. Therefore, in our experiments, we use bootstrap in combination with the choice $k_0 = k-1$ (in order to minimize the bound in Theorem 5(i)).

2.2.2 Elements of proofs

Ergodic properties of i-SIR: The systematic scan two-stage Gibbs sampler is a well-studied MCMC algorithmic structure, and we summarize its most important properties in Theorem 6 below; see Liu et al. (1994); Andrieu (2016) and (Robert and Casella, 2004, Chapter 9) as well as the references therein. In particular, as shown in Liu et al. (1994), the state and candidate-pool Markov chains $(Y_k)_{k\in\mathbb{N}}$ and $(X_k^{1:N})_{k\in\mathbb{N}}$ satisfy a duality property referred to as *interleaving* (Theorem 6(iii)).

Theorem 6. *Assume that for every* $x \in \mathbb{X}$, $w(x) > 0$, $\lambda(w) < \infty$ *and that there exists a set* $C \in \mathcal{X}$ *such that* $\lambda(C) > 0$ *and* $\sup_{x\in C} w(x)/\lambda(w) < \infty$. *Then,*

(i) the Markov kernel $\mathbf{P}_N$ *is Harris recurrent and ergodic with unique invariant distribution* $\boldsymbol{\varphi}_N$.

(ii) the Markov kernel P_N *is* π*-reversible, Harris recurrent and ergodic.*

(iii) the two Markov chains $(Y_k)_{k\in\mathbb{N}}$ *and* $(X_k^{1:N})_{k\in\mathbb{N}}$ *are conjugate of each other with the interleaving property, i.e., for every initial distribution* $\boldsymbol{\xi}$ *and* $k \in \mathbb{N}$, *under* $\mathbb{P}_{\boldsymbol{\xi}}$,

(a) $X_k^{1:N}$ *and* $X_{k+1}^{1:N}$ *are conditionally independent given* Y_k,

(b) Y_k *and* Y_{k+1} *are conditionally independent given* $X_{k+1}^{1:N}$;

(c) moreover, under $\mathbb{P}_{\boldsymbol{\varphi}_N}$, $(Y_k, X_{k-1}^{1:N})$ *and* $(Y_k, X_k^{1:N})$ *are identically distributed.*

The ergodic behavior of the i-SIR algorithm has been studied in many works; see Lee (2011); Lindsten et al. (2015); Andrieu et al. (2018) in particular. The analysis is particularly simple under the assumption that the importance weight function w is bounded, as imposed by **A** 1. Recall that the *total variation-distance* between two probability measures ξ and ξ' on $(\mathbb{X}, \mathcal{X})$ is given by $d_{TV}(\xi, \xi') = \sup_{g:\mathrm{osc}(g)\leq 1}\{\xi(g) - \xi'(g)\}$, where $\mathrm{osc}(g) = \sup_{(x,x')\in\mathbb{X}^2}|g(x) - g(x')|$ denotes the oscillator norm of a measurable function g. The following result establishes the uniform geometric ergodicity of the state chain $(Y_k)_{k\in\mathbb{N}}$.

Theorem 7. *Assume* **A** *1. Then for every* $N \geq 2$, $y \in \mathbb{X}$ *and* $k \in \mathbb{N}$, $d_{TV}(\mathsf{P}_N^k(y,\cdot),\pi) \leq \kappa_N^k$, *where* κ_N *is given in* (2.10).

The proof is given in Lindsten et al. (2015); Andrieu et al. (2018), but we provide it in Section A.1.5 for completeness. For uniformly ergodic Markov chains, it is often more appropriate to work with the mixing time

$$\min\{k \in \mathbb{N} : \sup_{y\in\mathbb{X}} d_{TV}(\mathsf{P}_N^k(y,\cdot),\pi) \leq 1/4\} \leq \tau_{mix,N} \tag{2.13}$$

(where $\tau_{mix,N}$ is given in (2.10)), *i.e.*, the number of time steps required for the distribution of the chain to be within a certain total variation distance from its stationary distribution Aldous et al. (1997); Hsu

et al. (2019). An interesting consequence of the interleaving property is that if the Markov chain $(Y_k)_{k\in\mathbb{N}}$ is (geometrically) ergodic, then the Markov chain $(X_k^{1:N})_{k\in\mathbb{N}}$ is (geometrically) ergodic as well with the same mixing time; see (Robert and Casella, 2004, Corollary 9.14)).

Bias of the BR-SNIS estimator: As the BR-SNIS estimator $\Pi_N f(X_k^{1:N})$ (where Π_N is defined in (2.5)) is made up by a ratio of the two unnormalized estimators $\Gamma_N f(X_k^{1:N})$ and $\Gamma_N \mathbb{1}_{\mathbb{X}}(X_k^{1:N})$, a key ingredient in the proof of Theorem 4 is to bound the bias and the p^{th} order moments of statistics defined as ratios of sums of random variables that are not necessarily independent. The basic idea is to reduce the study of these relations to the analysis of the moments of the numerator and the denominator of these statistics and to exploit their concentration around the respective (conditional and unconditional) means. The main results that we will use in the rest of the paper are summarized in Section A.2.

Lemma 8. *For every initial distribution $\boldsymbol{\xi}$ on $(\mathbb{X}^{N+1}, \mathcal{X}^{\otimes(N+1)})$, $k \in \mathbb{N}^*$, and bounded measurable function $f : \mathbb{X} \to \mathbb{R}$, it holds that*

(i) for every $y \in \mathbb{X}$, $\boldsymbol{\Lambda}_N \Gamma_N f(y) = (1 - 1/N)\lambda(wf) + (1/N)w(y)f(y)$,
(ii) $\mathbb{E}_{\boldsymbol{\xi}}\left[\Gamma_N f(X_k^{1:N}) \,\middle|\, Y_{k-1}\right] = \boldsymbol{\Lambda}_N \Gamma_N f(Y_{k-1})$, $\mathbb{P}_{\boldsymbol{\xi}}$-a.s.,
(iii) $\mathbb{E}_{\boldsymbol{\xi}}\left[\{\Gamma_N f(X_k^{1:N}) - \boldsymbol{\Lambda}_N \Gamma_N f(Y_{k-1})\}^2 \,\middle|\, Y_{k-1}\right] = (N-1)/N^2 \lambda(\{wf - \lambda(wf)\}^2), \mathbb{P}_{\boldsymbol{\xi}}$-a.s.

We now have all the elements that allow us to determine the first important result of this work, namely the bias and the MSE of the estimator $\Pi_N f(X_k^{1:N})$ of $\pi(f)$.

Proof of Theorem 4. We establish the bias bound in (i) and postpone the proof of the bounds on the MSE and the covariance in (ii) and (iii) to the supplement. Define the measure $\xi(A) = \boldsymbol{\xi}(A \times \mathbb{X})$, $A \in \mathcal{X}$, and the kernel $\mathsf{P}_N = \boldsymbol{\Lambda}_N \Pi_N$ on $\mathbb{X} \times \mathcal{X}$. Consequently, $\mathsf{P}_N f(Y_{k-1}) = \mathbb{E}_{\boldsymbol{\xi}}[\Pi_N f(X_k^{1:N}) \mid Y_{k-1}]$ and $\boldsymbol{\Lambda}_N \Gamma_N f(Y_{k-1}) = \mathbb{E}_{\boldsymbol{\xi}}[\Gamma_N f(X_k^{1:N}) \mid Y_{k-1}]$, $\mathbb{P}_{\boldsymbol{\xi}}$-a.s. Since $(Y_k)_{k\in\mathbb{N}}$ is, under $\mathbb{P}_{\boldsymbol{\xi}}$, a Markov chain with initial distribution ξ and Markov kernel P_N (see (2.9)), it holds that

$$\mathbb{E}_{\boldsymbol{\xi}}[\Pi_N f(X_k^{1:N})] = \mathbb{E}_{\boldsymbol{\xi}}[\mathsf{P}_N f(Y_{k-1})] = \mathbb{E}_{\boldsymbol{\xi}}[\mathbb{E}_{\boldsymbol{\xi}}[\mathsf{P}_N f(Y_{k-1}) \,|\, Y_0]] = \xi \mathsf{P}_N^{k-1} \mathsf{P}_N f.$$

Consequently, the proof is concluded by establishing that for every $k \in \mathbb{N}^*$,

$$\left|\xi \mathsf{P}_N^{k-1} \mathsf{P}_N f - \pi(f)\right| \le \varsigma^{bias} \kappa_N^{k-1} (N-1)^{-1}. \tag{2.14}$$

On the other hand, since by Theorem 3, $\pi(\mathsf{P}_N f) = \pi(f)$, we may use Theorem 7 to obtain the bound

$$|\xi \mathsf{P}_N^{k-1} \mathsf{P}_N f - \pi(f)| = |\xi \mathsf{P}_N^{k-1} \mathsf{P}_N f - \pi(\mathsf{P}_N f)| \le \kappa_N^{k-1} \operatorname{osc}(\mathsf{P}_N f).$$

Finally, we establish (2.14) by bounding $\operatorname{osc}(\mathsf{P}_N f)$. Note that

$$\operatorname{osc}(\mathsf{P}_N f) \le 2 \left\| \mathsf{P}_N f - \boldsymbol{\Lambda}_N \Gamma_N f / (\boldsymbol{\Lambda}_N \Gamma_N \mathbb{1}_{\mathbb{X}}) \right\|_\infty + 2 \left\| \boldsymbol{\Lambda}_N \Gamma_N f / (\boldsymbol{\Lambda}_N \Gamma_N \mathbb{1}_{\mathbb{X}}) - \pi(f) \right\|_\infty, \tag{2.15}$$

where, for every $y \in \mathbb{X}$, using Theorem 38,

$$\begin{aligned} &|\mathsf{P}_N f(y) - \boldsymbol{\Lambda}_N \Gamma_N f(y) / \boldsymbol{\Lambda}_N \Gamma_N \mathbb{1}_{\mathbb{X}}(y)| \\ &\le \frac{1}{2} \{\boldsymbol{\Lambda}_N \Gamma_N \mathbb{1}_{\mathbb{X}}(y)\}^{-2} \{\boldsymbol{\Lambda}_N [\{\Gamma_N f - \boldsymbol{\Lambda}_N \Gamma_N f(y)\}^2](y) + 3 \boldsymbol{\Lambda}_N [\{\Gamma_N \mathbb{1}_{\mathbb{X}} - \boldsymbol{\Lambda}_N \Gamma_N \mathbb{1}_{\mathbb{X}}(y)\}^2](y)\}. \end{aligned} \tag{2.16}$$

Now, since $\boldsymbol{\Lambda}_N \Gamma_N \mathbb{1}_{\mathbb{X}}(y) \ge (1 - 1/N)\lambda(w)$, we get, using Lemma 8,

$$\left\| \mathsf{P}_N f - \frac{\boldsymbol{\Lambda}_N \Gamma_N f}{\boldsymbol{\Lambda}_N \Gamma_N \mathbb{1}_{\mathbb{X}}} \right\|_\infty \le (2(N-1))^{-1} \{\lambda(w)\}^{-2} \{\lambda(\{wf - \lambda(wf)\}^2) + 3\lambda(\{w - \lambda(w)\}^2)\} \tag{2.17}$$

$$\le 2(N-1)^{-1} \lambda(w^2) / (\lambda(w))^2. \tag{2.18}$$

On the other hand, using the elementary inequality $a/b - c/d = a(d-b)/bd + (a-c)/d$, we get, as $\pi(f) = \lambda(wf)/\lambda(w)$,

$$\frac{\mathbf{\Lambda}_N \Gamma_N f(y)}{\mathbf{\Lambda}_N \Gamma_N \mathbb{1}_{\mathbb{X}}(y)} - \pi(f) = (1/N) \frac{\mathbf{\Lambda}_N \Gamma_N f(y)}{\mathbf{\Lambda}_N \Gamma_N \mathbb{1}_{\mathbb{X}}(y)} \{1 - w(y)/\lambda(w)\} + (1/N)\{w(y)f(y) - \lambda(wf)\}/\lambda(w). \quad (2.19)$$

Finally, the bound (2.14) is established by noting that

$$\|\mathbf{\Lambda}_N \Gamma_N f/(\mathbf{\Lambda}_N \Gamma_N \mathbb{1}_{\mathbb{X}}) - \pi(f)\|_\infty \leq 2N^{-1}\{1 + w(y)/\lambda(w)\} \leq 2N^{-1}(1+\omega). \quad (2.20)$$

□

2.2.3 Related works

The first use of the IS method, then as a variance reduction technique, dates back to the '50s; see Hesterberg (1995); Kroese and Rubinstein (2012) and the references therein. Today, the renewed interest in IS parallels the flurry of activity in the probabilistic ML community and its ever-increasing computational demands; thus, it is impossible to fully present the literature. We therefore limit ourselves to describing results that have inspired our work, and refer the readers to the recent reviews Agapiou et al. (2017); Elvira and Martino (2021) for additional references.

There is clearly a plethora of modern ML applications where the standard SNIS estimator may be substantially improved using the BR-SNIS method. To mention just a selection of examples, SNIS plays a key role for a robust off-policy selection strategy BY Kuzborskij et al. (2021) (extending Swaminathan and Joachims (2015); Metelli et al. (2018)), Bayesian problems (see, *e.g.*, (Agapiou et al., 2017, Section 3)), Bayesian transfer learning Karbalayghareh et al. (2018); Maddouri et al. (2022), variational autoencoders Chen et al. (2022), inference of energy-based models Lawson et al. (2019), patch-based image restoration Niknejad et al. (2019) and many more. In stochastic-approximation procedures, where a statistical estimator or algorithm is employed repeatedly to produce mean-field estimates, controlling its bias becomes critical Tadić and Doucet (2017); Karimi et al. (2019). Thus, it is natural to aim at minimizing the bias for a given computational budget, provided that the variance does not explode. For this reason, bias reduction (or unbiasedness) in stochastic simulation has been the subject of extensive research during the last decades; see Glynn and Rhee (2014); Jacob et al. (2020b). The present paper contributes to this line of research.

Despite long-standing interest in SNIS, there are only few theoretical results. For example, (Agapiou et al., 2017, Theorem 2.1) provides bounds on the bias and variance of SNIS, results that we extend to BR-SNIS in Theorem 4. Moreover, (Metelli et al., 2018, Proposition D.3) provides a suboptimal variance bound based on a bound for the second-order moment. This result can be compared to the sophisticated sub-Gaussian concentration bound for BR-SNIS obtained in Theorem 5 (a result that can be obtained for SNIS using the same proof mechanism; see Section A.1.8). Finally, Kuzborskij et al. (2021) obtains a semi-empirical sub-Gaussian concentration inequality using the Efron-Stein estimate of variance and the Harris inequality.

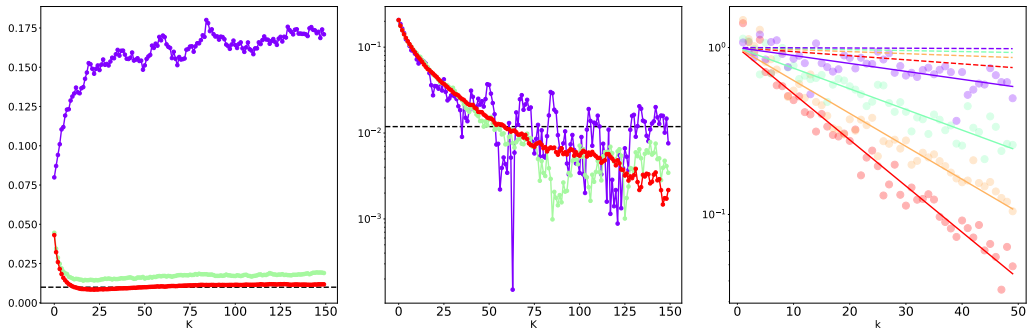
As an MCMC sampling method, the i-SIR algorithm that has been applied successfully in many situations. It was recently used—under the alternative name *conditional importance sampling*—in Naesseth et al. (2020) for *Markovian score climbing*. In the same work, it is mentioned that it is possible to "Rao-Blackwellize" the gradient of the score using the proposed candidates, which is in line with the recycling argument underpinning the estimator suggested by us, but without theoretical justifications. In its most basic form, the i-SIR algorithm appeared in the pioneering work of Tjelmeland (2004a). The same idea played a key role in the development of the *particle Gibbs sampler* Andrieu et al. (2010a, 2018); Naesseth et al. (2019), which extends i-SIR principles to *sequential Monte Carlo methods*. An approach very similar to BR-SNIS can be taken also in this context; however, casting BR-SNIS into the framework of particle Gibbs methods is a non-trivial problem which is the subject of ongoing work.

2.3 Experimental results

In this section we compare numerically the performances of BR-SNIS and SNIS in three different settings: mixture of Gaussians, Bayesian logistic regression and variational autoencoders (VAE). We leave to the supplementary material (Section A.3.1) the detailed numerical verification of the bounds established in Section 2.2.

Mixture of Gaussian distributions: We start with an example where the target distribution π is a mixture of two Gaussian distributions of dimension $d = 7$, as shown in Figure 2.2a. The proposal distribution is a Student distribution with $\nu = 3$ degrees of freedom. The test function is $f = \mathbb{1}_A - \mathbb{1}_B$, where A and B are a d-dimensional rectangle intersecting each of the modes of π (see Section A.3.1 for precise definitions). We verify the positive effect of bootstrap in Figures 2.1a and 2.1b by computing the bias and the MSE over 1000 chains for $N = 129$ for several k. The purple, green, and red curves correspond to a number of bootstrap rounds of $1, 21$, and 201, respectively. We illustrate the decay of the mean Sliced Wasserstein distance (according to Bonneel et al. (2015)) with k for different values of N ($N = 8$ purple, $N = 32$ green, $N = 64$ orange, and $N = 128$ red) in Figure 2.1c. The decay of the Wassertein distance is directly linked to the mixing time of the i-SIR kernel (see (2.10)), and hence allows us to represent the effective mixing time of the chain. Moreover, we represent the theoretical slopes as dashed lines. This illustrates that the effective value of $\tau_{mix,N}$ is smaller than its theoretical bound. The bias and MSE for SNIS with $M = 25600$ are shown in black dashed lines.

We compare the bias (Figure 2.2b) and MSE (Figure 2.2c) of BR-SNIS and SNIS for a fixed budget with a total number of $M = 16384$ samples. We run the experiments 10^6 times; we compute the bias and MSE over batches of 10^4 replications using the true value of $\pi(f)$ computed above (the boxplots in Figure 2.2 are therefore obtained over 100 replications). For the algorithm BR-SNIS, we used $N \in \{129, 513\}$, $k_0 = k_{max} - 1$ and $k_{max} = M/(N-1)$ bootstrap rounds. As can be seen from Figure 2.2b, BR-SNIS



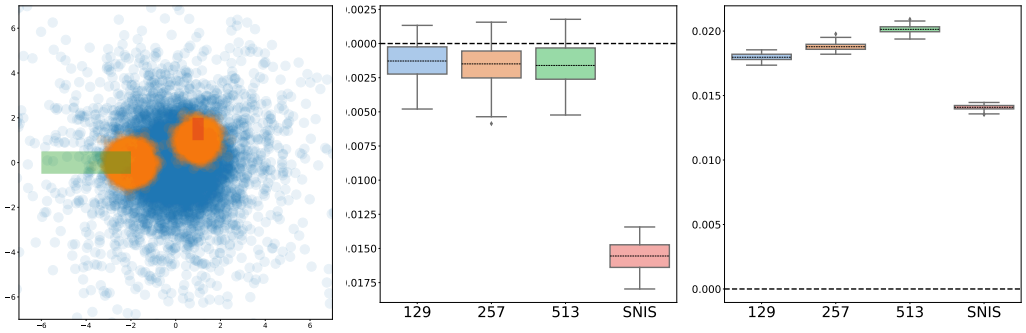
(a) MSE *(b) Bias in log scale* *(c) Sliced Wasserstein*

Figure 2.1

significantly reduces bias (by a factor of almost 10) w.r.t. standard SNIS for both configurations, while MSE increases only slightly (at around 20%), as can be seen in Figure 2.2c. The code used for this experiment is available at [1]. We also show in Section A.3.1 that $k_0 = \lfloor 0.625 k_{max} \rfloor$ can lead to about 3 times less bias w.r.t. standard SNIS while only augmenting the MSE of 10%. We have also compared in BR-SNIS to zero bias estimators based on SNIS such as Middleton et al. (2019), the results are in shown in Section A.3.1.

Bayesian Logistic regression: We consider posterior inference in a Bayesian logistic regression model. Let $\mathcal{D}_{train} = (\mathbf{x}_i, y_i)_{i=1}^T$ be a dataset, where each $\mathbf{x}_i \in \mathbb{R}^d$ is a vector of covariates and $y_i \in \{-1, 1\}$ is a

[1] https://github.com/gabrielvc/br_snis



(a) 2d projection *(b) Bias* *(c) MSE*

*Figure 2.2: Comparison between SNIS and BR-SNIS for the same budget. In each boxplot the dotted line represents the **mean** value of the samples.*

binary response. Let $p(y_i \mid \mathbf{x}_i; \theta) = \{1 + \exp(-y_i\, \mathbf{x}_i^\top \theta)\}^{-1}$ be the probability of the ith observation at $\theta \in \Theta \subseteq \mathbb{R}^d$ and $\pi_0(\mathrm{d}\theta)$ be a prior distribution for θ. The Bayesian posterior is given

$$\pi(\mathrm{d}\theta) = \mathrm{Z}^{-1}\pi_0(\mathrm{d}\theta)\exp(\mathcal{L}_T(\theta)), \quad \mathcal{L}_T(\theta) = \textstyle\sum_{i=1}^T \ln p(y_i \mid \mathbf{x}_i; \theta), \quad \mathrm{Z} = \int \exp(\mathcal{L}_T(\theta))\pi_0(\mathrm{d}\theta).$$

For numerical illustration, we use the heart failure clinical records ($d = 13$, $T = 299$), breast cancer detection ($d = 30$, $T = 569$), and Covertype ($d = 55$, $T = 4 \cdot 10^4$) datasets from the UCI machine learning repository. For Covertype, we use Cover type 1 (Spruce/Fir) and Cover type 2 (Lodgepole Pine) classes to define a binary classification problem. As a prior, we use a Gaussian distribution $\mathrm{N}(0, \tau^{-2}\mathbf{I})$ with $\tau^2 = 5 \cdot 10^{-2}$. The importance distribution λ is Gaussian with mean and diagonal covariance learned by variational inference; see Section A.3.2 for details. The boxplots for bias in Figure 2.3 were constructed in the same way as those in Figure 2.2. We compare two test functions, $f(\theta) = \theta$,

.	CoverType	Breast	Heart
SNIS, M = 32	0.0028 +/- 0.0012	0.00011 +/- 6.04e-5	0.00023 +/- 7.24e-5
BR-SNIS, M= 32	0.0014 +/- 0.0003	7.9e-5 +/- 5.5e-5	0.00012 +/- 6.7e-5
SNIS, M = 512	0.0026 +/- 0.0017	4.3e-5 +/- 3.3e-5	7.8e-5 +/- 6.8e-5
BR-SNIS, M= 512	0.0013 +/- 0.0003	3.5e-5 +/- 2.2e-5	4.9e-5 +/- 5.2e-5

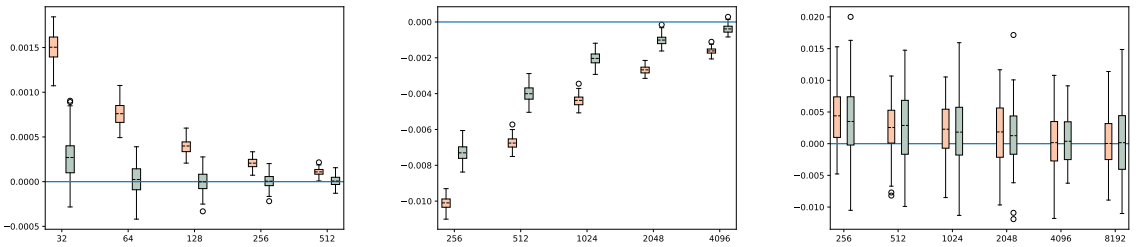
Table 2.1: Comparison of the TV distance between the posteriors (Lower is better).

corresponding to evaluation of the posterior mean, and $f(\theta) = p(y \mid \mathbf{x}, \theta)$, where $(\mathbf{x}, y) \in \mathcal{D}_{test}$. This last function allows us to compute a TV distance for the predictive distribution. Indeed, in a classification context, one can compute the TV distance between any two predictive distributions p and $\hat{p}$ as

$$d_{TV}(\hat{p}, p) = T^{-1} \textstyle\sum_{i=1}^T \frac{1}{2} \sum_{j=0}^1 |\hat{p}(y = j \mid \mathbf{x}_i, \mathcal{D}_{train}) - p(y = j \mid \mathbf{x}_i, \mathcal{D}_{train})|, \tag{2.21}$$

where we compare the predictive distribution $p(y \mid x, \mathcal{D}_{train}) = \int p(y \mid x, \theta)\pi(\theta)\mathrm{d}\theta$ and $\hat{p}$ is the estimation of this quantity, provided in the experiments by SNIS or BR-SNIS. From Figure 2.3 we can see that for each dataset we have a constant decrease in bias, while the variance increases only slightly. We plot the bias in other components of θ and provide further numerical details in Section A.3.2.

Generative Model: We now extend our methodology to the more complex *deep latent generative models* (DLGM). A DLGM defines a family of probability densities $p_\theta(x)$ over an observation space $x \in \mathbb{R}^P$ by introducing a latent variable $z \in \mathbb{R}^d$, defining the joint density function $p_\theta(x, z)$ (with respect



(a) Heart 8 *(b) Breast 11* *(c) Covertype 6*

Figure 2.3: Visualization of the distribution for each datasets. Each boxplot is grouped by budget, the left one represent SNIS and the right represent BR-SNIS.

to Lebesgue measure) and aiming to find a parameter θ maximizing the marginal log-likelihood of the model $p_\theta(x) = \int p_\theta(x, z)\mathrm{d}z$. Under simple technical assumptions, by Fisher's identity,

$$\nabla_\theta \log p_\theta(x) = \int \nabla_\theta \log p_\theta(x, z) p_\theta(z \mid x)\mathrm{d}z, \tag{2.22}$$

In most cases, the conditional density $p_\theta(z \mid x) = p_\theta(x, z)/p_\theta(x)$ is intractable and can only be sampled. The variational autoencoder Kingma and Welling (2014) is based on the introduction of an additional parameter ϕ and a family of variational distributions $q_\phi(z \mid x)$. The joint parameters $\{\theta, \phi\}$ are then inferred by maximizing the *evidence lower bound* (ELBO) defined by

$$\mathcal{L}(\theta, \phi) = \log p_\theta(x) - \mathrm{KL}(q_\phi(\cdot \mid x) \,\|\, p_\theta(\cdot \mid x)) \leq \log p_\theta(x).$$

This basic setup has been further developed and improved in many directions. Here we consider the *importance weighted autoencoder* (IWAE) Burda et al. (2015), which relies on SNIS to design a tighter ELBO on the log-likelihood. The objective of the IWAE is given by

$$\mathcal{L}_M(\theta, \phi) = \int \log \left(M^{-1} \textstyle\sum_{i=1}^M w_{\theta,\phi,x}(z_i) \right) \prod_{\ell=1}^M q_\phi(z_\ell \mid x)\mathrm{d}z_i, \tag{2.23}$$

where $w_{\theta,\phi,x}(z) = p_\theta(x, z)/q_\phi(z \mid x)$ denote the importance weights. However, writing, following (Burda et al., 2015, Eq. (13)),

$$\nabla_\theta \mathcal{L}_M(\theta, \phi) = \int \textstyle\sum_{i=1}^M \omega_{\theta,\phi,x}^{(i)} \nabla_\theta \log w_{\theta,\phi,x}(z_i) \prod_{\ell=1}^N q_\phi(z_\ell \mid x)\mathrm{d}z_\ell,$$

where $\omega_{\theta,\phi,x}^{(i)} = w_{\theta,\phi,x}(z_i)/\sum_{j=1}^M w_{\theta,\phi,x}(z_j)$ are normalized importance weights, yields an expression of the gradient that corresponds exactly to the biased SNIS approximation of (2.22). Thus, the optimization problem will suffer from bias. We hence propose to use BR-SNIS for learning IWAE. The proposed algorithm proceeds in two steps, which are repeated during the optimization (details are given in Section A.3.3)

- First, update the parameter ϕ as in the IWAE algorithm (using the reparameterization trick and following the methodology of Burda et al. (2015)) according to $\phi^{(t+1)} = \phi^{(t)} - \eta \nabla_\phi \mathcal{L}_M(\theta^{(t)}, \phi^{(t)})$.
- Second, update the parameter θ by estimating (2.22) using BR-SNIS for $\pi(z) = p_\theta(x, z)$, $f(z) = \nabla_\theta \log p_\theta(x, z)$ and $\lambda(z) = q_\phi(z \mid x)$.

We refer to this model as BR-IWAE. As an illustration, we train the model using the binarized MNIST dataset Salakhutdinov and Murray (2008), where $x \in \{0, 1\}^{784}$ are binarized digits images in dimension 784. For both for the encoder q_ϕ and the decoder p_θ, we use a convolutional neural network (more details are given in Section A.3.3). For comparison, we estimate the log-likelihood using the VAE, IWAE and BR-IWAE approaches, and the result is reported in Table 2.2. All models are run for 100 epochs, using the Adam optimizer Kingma and Ba (2015a) and a learning rate of 10^{-4}. The complete experimental details are given in Section A.3.3.

Latent dimension (d)	VAE	IWAE	BR-IWAE ($\boldsymbol{k = 8}$)
10	-87.40 ± 0.14	-86.44 ± 0.10	$\mathbf{-86.29 \pm 0.09}$
20	-83.55 ± 0.10	-81.81 ± 0.06	$\mathbf{-81.66 \pm 0.12}$
40	-82.90 ± 0.07	-81.05 ± 0.09	$\mathbf{-81.01 \pm 0.05}$

Table 2.2: Comparison of the mean log likelihood over the MNIST validation set (Higher is better).

2.4 Conclusion

In this paper, we have introduced a novel method, BR-SNIS, which improves over SNIS when it comes to producing close to unbiased estimates of expectations taken w.r.t. to distributions known only up to a normalizing constant, a ubiquitous problem in machine learning and statistics. The high performance of BR-SNIS is supported theoretically by non-asymptotic bias, variance and high-probability bounds. We illustrate our method on various examples, which show the practical advantages of BR-SNIS over SNIS. Finally, BR-SNIS is naturally adapted to other IS based methods, for example Thin et al. (2021), which use a Hamiltonian (gradient-based) transform Neal et al. (2011) as part of the IS proposal. The extension of BR-SNIS to Thin et al. (2021) would produce an Hamiltonian based sampler able to recycle all samples, contrarily to other classical Hamiltonian-based methods Neal et al. (2011); Hoffman et al. (2014). BR-SNIS can also be extended to Particle Markov chain Monte Carlo methods such as Particle Gibbs with Ancestor sampling Lindsten et al. (2014b).

Chapter 3

PPG: Particle-based, Rapid Incremental Smoother Meets Particle Gibbs

3.1 Introduction

Feynman–Kac formulae play a key role in many models used in statistics, physics, and many other fields; see Del Moral (2004); Del Moral (2013); Chopin and Papaspiliopoulos (2020), and the references therein. Let $\{(\mathsf{X}_t, \mathcal{X}_t)\}_{t\in\mathbb{N}}$ be a sequence of measurable spaces and define, for every $t \in \mathbb{N}$, $\mathsf{X}_{0:t} \coloneqq \prod_{m=0}^{t} \mathsf{X}_m$ and $\mathcal{X}_{0:t} \coloneqq \bigotimes_{m=0}^{t} \mathcal{X}_m$. For a sequence $\{M_t\}_{t\in\mathbb{N}}$ of Markov kernels $M_t : \mathsf{X}_t \times \mathcal{X}_{t+1} \to [0,1]$, an initial distribution $\eta_0 \in \mathsf{M}_1(\mathcal{X}_0)$, and a sequence $\{g_t\}_{t\in\mathbb{N}}$ of bounded measurable potential functions $g_t : \mathsf{X}_t \to \mathbb{R}_+$, a sequence $\{\eta_{0:t}\}_{t\in\mathbb{N}}$ of *Feynman–Kac path measures* is defined by

$$\eta_{0:t} : \mathcal{X}_{0:t} \ni A \mapsto \frac{\gamma_{0:t}(A)}{\gamma_{0:t}(\mathsf{X}_{0:t})}, \quad t \in \mathbb{N}, \tag{3.1}$$

where

$$\gamma_{0:t} : \mathcal{X}_{0:t} \ni A \mapsto \int \mathbb{1}_A(x_{0:t})\, \eta_0(\mathrm{d}x_0) \prod_{m=0}^{t-1} Q_m(x_m, \mathrm{d}x_{m+1}), \tag{3.2}$$

with

$$Q_m : \mathsf{X}_m \times \mathcal{X}_{m+1} \ni (x, A) \mapsto g_m(x) M_m(x, A) \tag{3.3}$$

being unnormalized kernels. By convention, $\eta_{0:0} \coloneqq \eta_0$. Note that each $\eta_{0:t}$ is a probability measure, whereas $\gamma_{0:t}$ is not normalized. For every $t \in \mathbb{N}^*$, we also define the marginal distribution $\eta_t : \mathcal{X}_t \ni A \mapsto \eta_{0:t}(\mathsf{X}_{0:t-1} \times A)$. In the context of nonlinear filtering in *general state-space hidden Markov models*(HMMs), $\eta_{0:t}$ and η_t are, the *joint smoothing* and *filter distribution*, respectively, at time t; see Del Moral (2004); Cappé et al. (2005a); Chopin and Papaspiliopoulos (2020).

For most problems of practical interest, the Feynman–Kac path or marginal measures are intractable, and so is any expectation associated with the same. As a result, considerable research has been devoted to developing Monte Carlo, or *particle*, approximations of such measures. A *particle filter* approximates the marginal distribution flow $\{\eta_t\}_{t\in\mathbb{N}}$ by a sequence of occupation measures, associated with a swarm of *particles* $\{\xi_t^i\}_{i=1}^N$, $N \in \mathbb{N}$, where each particle ξ_t^i is a random draw in X_t. Particle filters revolve around two operations: a *selection step*, which duplicates or sorts out particles with large or small importance weights, respectively, and a *mutation step*, which randomly evolves the selected particles in the state space. An alternating and iterative application of selection and mutation results in a swarm of N particles that are both serially and spatially dependent. Feynman–Kac path models can also be interpreted as laws associated with a certain type of Markovian backward dynamics; this interpretation is useful, for

example, for the smoothing problem in nonlinear filtering Douc et al. (2011); Del Moral et al. (2010). Several convergence results have been established for particle filters, as the number N of particles tends to infinity; see for example, Del Moral (2004); Douc and Moulines (2008); Del Moral (2013); Chopin and Papaspiliopoulos (2020). In addition, a number of nonasymptotic results have been obtained for these methods, including bounds on their bias and L_p error, as well as exponential concentration inequalities and propagation of chaos estimates. Extensions to the backward interpretation can also be found in Douc et al. (2011); Del Moral et al. (2010).

In this work, we focus on the problem of recursively computing smoothed expectations

$$\eta_{0:t} h_t = \int h_t(x_{0:t})\, \eta_{0:t}(\mathrm{d}x_{0:t}), \quad t \in \mathbb{N},$$

where we introduce the vector notation $x_{0:t} = (x_0, \ldots, x_t) \in \mathsf{X}_{0:t} \coloneqq \mathsf{X}_0 \times \cdots \times \mathsf{X}_t$ for *additive functionals* h_t of the form

$$h_t(x_{0:t}) \coloneqq \sum_{m=0}^{t-1} \tilde{h}_m(x_{m:m+1}), \quad x_{0:t} \in \mathsf{X}_{0:t}. \tag{3.4}$$

In nonlinear filtering problems, such expectations appear in the context of maximum-likelihood parameter estimation, for instance, when computing the *score function* (the gradient of the log-likelihood function) or the *expectation–maximization* (EM) surrogate; see Cappé (2001); Andrieu and Doucet (2003); Poyiadjis et al. (2005); Cappé (2011); Poyiadjis et al. (2011). In Olsson and Westerborn (2017), the authors propose an efficient *particle-based rapid incremental smoother* (`PARIS`), with linear computational complexity in the number of particles under weak assumptions and limited memory requirements, that samples on-the-fly from the backward dynamics induced by the particle filter. An interesting feature is that it requires two or more backward draws per particle to cope with the degeneracy of the sampled trajectories and remain numerically stable in the long run, with an asymptotic variance that grows only linearly with time.

In this paper, we propose a method to reduce the bias of the `PARIS` estimator of $\eta_{0:t} h_t$. The idea is to mix the `PARIS` with a version of the *particle Gibbs* algorithm with backward sampling Andrieu et al. (2010b); Lindsten et al. (2014a); Chopin and Singh (2015); Del Moral et al. (2016); Del Moral and Jasra (2018) by introducing a conditional `PARIS` algorithm. This leads to the *Parisian particle Gibbs* (`PPG`) *algorithm*, from which we derive an upper bound on the bias that decreases inversely proportionally to the number of particles and exponentially fast with the iteration index (under assumptions guaranteeing that the particle Gibbs sampler is uniformly ergodic).

The remainder of the paper is structured as follows. In 3.2 we discuss the Feynman–Kac model, along with its backward interpretation, and introduce the particle Gibbs sampler. Our presentation is inspired by Del Moral et al. (2016), but differs in that it avoids the use of quotient spaces of Del Moral et al. (2016) and the extension of the distribution to the particle ancestral indices of Andrieu et al. (2010b). In 3.3, we introduce the `PARIS` algorithm and its conditional version, and show how it can be coupled with the particle Gibbs method with backward sampling, yielding the `PPG` algorithm. In 3.4, we present the central result of this study, namely, an upper bound on the bias of the `PPG` estimator as a function of the number of particles and the iteration index of the Gibbs algorithm. In addition, we provide an upper bound on the mean-squared error (MSE). In 3.5, we provide numerical experiment to illustrate our results. In 3.6, we present the most important and original proofs. Finally, the supplementary material contain pseudocode and additional technical proofs, respectively.

Notation. Let $\mathbb{R}_+ \coloneqq [0, \infty)$, $\mathbb{R}_+^* \coloneqq (0, \infty)$, $\mathbb{N} \coloneqq \{0, 1, 2, \ldots\}$, and $\mathbb{N}^* \coloneqq \{1, 2, 3, \ldots\}$ denote the sets of nonnegative and positive real numbers and the same for integers, respectively. We denote by I_N the $N \times N$ identity matrix. For any quantities $\{a_\ell\}_{\ell=m}^t$, we denote vectors as $a_{m:t} \coloneqq (a_m, \ldots, a_t)$, and for any

$(m,t) \in \mathbb{N}^2$ such that $m \le t$, we let $[\![m,t]\!] := \{m, m+1, \dots, t\}$. For a given measurable space $(\mathsf{X}, \mathcal{X})$, where $\mathcal{X}$ is a countably generated σ-field, we denote by $\mathsf{F}(\mathcal{X})$ the set of bounded $\mathcal{X}/\mathcal{B}(\mathbb{R})$-measurable functions on X. For any $h \in \mathsf{F}(\mathcal{X})$, we let $\|h\|_\infty := \sup_{x\in\mathsf{X}} |h(x)|$ and $\operatorname{osc}(h) := \sup_{(x,x')\in\mathsf{X}^2} |h(x) - h(x')|$ denote the supremum and oscillator norms, respectively, of h. Let $\mathsf{M}(\mathcal{X})$ be the set of σ-finite measures on $(\mathsf{X}, \mathcal{X})$, and $\mathsf{M}_1(\mathcal{X}) \subset \mathsf{M}(\mathcal{X})$ be the probability measures.

Let $(\mathsf{Y}, \mathcal{Y})$ be another measurable space. A possibly unnormalized transition kernel K on $\mathsf{X} \times \mathcal{Y}$ induces two integral operators, one acting on measurable functions, and the other on measures; specifically, for $h \in \mathsf{F}(\mathcal{X} \otimes \mathcal{Y})$ and $\nu \in \mathsf{M}_1(\mathcal{X})$, define the measurable function

$$Kh : \mathsf{X} \ni x \mapsto \int h(x,y)\, K(x, \mathrm{d}y)$$

and the measure

$$\nu K : \mathcal{Y} \ni A \mapsto \int K(x,A)\, \nu(\mathrm{d}x),$$

whenever these quantities are well defined. Now, let $(\mathsf{Z}, \mathcal{Z})$ be a third measurable space and L be another possibly unnormalized transition kernel on $\mathsf{Y} \times \mathcal{Z}$; we then define, with K as above, two different products of K and L, namely,

$$KL : \mathsf{X} \times \mathcal{Z} \ni (x,A) \mapsto \int L(y,A)\, K(x, \mathrm{d}y)$$

and

$$K \otimes L : \mathsf{X} \times (\mathcal{Y} \otimes \mathcal{Z}) \ni (x,A) \mapsto \iint \mathbb{1}_A(y,z)\, K(x,\mathrm{d}y)\, L(y,\mathrm{d}z),$$

whenever these are well defined. This also defines the $\otimes$ product of a kernel K on $\mathsf{X} \times \mathcal{Y}$ and a measure ν on $\mathcal{X}$, as well as of a kernel L on $\mathsf{Y} \times \mathcal{X}$ and a measure μ on $\mathcal{Y}$, as the measures

$$\nu \otimes K : \mathcal{X} \otimes \mathcal{Y} \ni A \mapsto \iint \mathbb{1}_A(x,y)\, K(x,\mathrm{d}y)\, \nu(\mathrm{d}x),$$
$$L \otimes \mu : \mathcal{X} \otimes \mathcal{Y} \ni A \mapsto \iint \mathbb{1}_A(x,y)\, L(y,\mathrm{d}x)\, \mu(\mathrm{d}y).$$

3.2 Particle models

In the next sections, we discuss *many-body Feynman–Kac models*, *backward interpretations*, *conditional dual processes*, and the `PARIS` algorithm. Our presentation follows that of Del Moral et al. (2016) closely, but with a different definition of the many-body extensions. We restate (in 10) a duality formula of Del Moral et al. (2016) relating these concepts. This formula provides a foundation for the particle Gibbs sampler described in 3.2.3 and subsequent developments.

3.2.1 Many-body Feynman–Kac models

In the following, we assume that all random variables are defined on a common probability space $(\Omega, \mathcal{F}, \mathbb{P})$. The distribution flow $\{\eta_m\}_{m\in\mathbb{N}}$ is intractable, in general, but can be approximated by using random samples $\boldsymbol{\xi}_m = (\xi_m^1, \dots, \xi_t^N)$, for $m \in \mathbb{N}$, of particles, where $N \in \mathbb{N}^*$ is a fixed Monte Carlo sample size and each particle ξ_m^i is an X_m-valued random variable. Such a particle approximation is based on the recursion $\eta_{m+1} = \Phi_m(\eta_m)$, for $m \in \mathbb{N}$, where Φ_m denotes the mapping

$$\Phi_m : \mathsf{M}_1(\mathcal{X}_m) \ni \eta \mapsto \frac{\eta Q_m}{\eta g_m}, \tag{3.5}$$

taking on values in $\mathsf{M}_1(\mathcal{X}_{m+1})$. In order to describe recursively the evolution of the particle population, let $m \in \mathbb{N}$ and assume that the particles $\boldsymbol{\xi}_m$ form a consistent approximation of η_m, in the sense that

$\mu(\boldsymbol{\xi}_m)h$, where $\mu(\boldsymbol{\xi}_m) \coloneqq N^{-1}\sum_{i=1}^N \delta_{\xi_m^i}$ (with δ_x denoting the Dirac measure located at x) is the occupation measure formed by $\boldsymbol{\xi}_m$, serves as a proxy for $\eta_m h$ for any η_m-integrable test function h. (Under general conditions, $\mu(\boldsymbol{\xi}_m)h$ converges in probability to η_m as $N \to \infty$; see Del Moral (2004); Chopin and Papaspiliopoulos (2020), and the references therein.) Then, in order to generate an updated particle sample approximating η_{m+1}, new particles $\boldsymbol{\xi}_{m+1} = (\xi_{m+1}^1, \dots, \xi_{m+1}^N)$ are drawn conditionally independently given $\boldsymbol{\xi}_m$ according to

$$\xi_{m+1}^i \sim \Phi_m(\mu(\boldsymbol{\xi}_m)) = \sum_{\ell=1}^N \frac{g_m(\xi_m^\ell)}{\sum_{\ell'=1}^N g_m(\xi_m^{\ell'})} M_m(\xi_m^\ell, \cdot), \quad i \in [\![1, N]\!].$$

Because this process of particle updating involves sampling from the mixture distribution $\Phi_m(\mu(\boldsymbol{\xi}_m))$, it can be decomposed into two substeps: *selection* and *mutation*. The selection step randomly chooses the ℓth mixture stratum with probability $g_m(\xi_m^\ell)/\sum_{\ell'=1}^N g_m(\xi_m^{\ell'})$, and the mutation draws a new particle ξ_{m+1}^i from the selected stratum $M_m(\xi_m^\ell, \cdot)$. In Del Moral et al. (2016), the term *many-body Feynman–Kac models* is related to the law of process $\{\boldsymbol{\xi}_m\}_{m\in\mathbb{N}}$. For all $m \in \mathbb{N}$, let $\mathbf{X}_m \coloneqq \mathsf{X}_m^N$ and $\boldsymbol{\mathcal{X}}_m \coloneqq \mathcal{X}_m^{\otimes N}$; then, $\{\boldsymbol{\xi}_m\}_{m\in\mathbb{N}}$ is an inhomogeneous Markov chain on $\{\mathbf{X}_m\}_{m\in\mathbb{N}}$, with transition kernels

$$\boldsymbol{M}_m : \mathbf{X}_m \times \boldsymbol{\mathcal{X}}_{m+1} \ni (\boldsymbol{x}_m, A) \mapsto \Phi_m(\mu(\boldsymbol{x}_m))^{\otimes N}(A)$$

and initial distribution $\boldsymbol{\eta}_0 = \eta_0^{\otimes N}$. Now, denote $\mathbf{X}_{0:t} \coloneqq \prod_{m=0}^t \mathbf{X}_m$ and $\boldsymbol{\mathcal{X}}_{0:t} \coloneqq \bigotimes_{m=0}^t \boldsymbol{\mathcal{X}}_m$. (Here, and in the following, we use a bold symbol to stress that a quantity is related to the many-body process.) The *many-body Feynman–Kac path model* refers to the flows $\{\boldsymbol{\gamma}_m\}_{m\in\mathbb{N}}$ and $\{\boldsymbol{\eta}_m\}_{m\in\mathbb{N}}$ of the unnormalized and normalized probability distributions, respectively, on $\{\boldsymbol{\mathcal{X}}_{0:m}\}_{m\in\mathbb{N}}$ generated by (3.1) and (3.2) for the Markov kernels $\{\boldsymbol{M}_m\}_{m\in\mathbb{N}}$, the initial distribution $\boldsymbol{\eta}_0$, the potential functions

$$\boldsymbol{g}_m : \mathbf{X}_m \ni \boldsymbol{x}_m \mapsto \mu(\boldsymbol{x}_m)g_m = \frac{1}{N}\sum_{i=1}^N g_m(x_m^i), \quad m \in \mathbb{N},$$

and the corresponding unnormalized transition kernels

$$\boldsymbol{Q}_m : \mathbf{X}_m \times \boldsymbol{\mathcal{X}}_{m+1} \ni (\boldsymbol{x}_m, A) \mapsto \boldsymbol{g}_m(\boldsymbol{x}_m)\boldsymbol{M}_m(\boldsymbol{x}_m, A), \quad m \in \mathbb{N}.$$

Finally, note that in the previous construction, the Markov property of the many-body Feynman–Kac model relies on the fact that each potential g_m is a function of a single state x_m only, as is the case in the standard Feynman–Kac model framework Del Moral (2004), and that the evolution of the particles follows the model dynamics given in (3.5) (so-called *bootstrap particle filtering*). In order to extend this to more general models (such as models where the potentials are allowed to depend on two consecutive states Lee et al. (2020) or, even more generally, where no structure at all is assumed for the unnormalized kernels (3.3) Gloaguen et al. (2022)) and particle dynamics (such as the *auxiliary particle filtering* framework introduced in Pitt and Shephard (1999)), we need to form a Markovian many-body process with tractable dynamics by furnishing each particle with an importance weight and an index that records the particle's ancestor in the previous generation. However, to avoid this technicality and to allow for a more clear-cut presentation of the methods and theoretical analysis in the coming sections, we stay within the framework of the standard Feynman–Kac models and bootstrap-type particle filters, even though extensions to more general settings may be possible.

3.2.2 Backward interpretation of Feynman–Kac path flows

Suppose that each kernel Q_t, for $t \in \mathbb{N}$, defined in (3.3), has a transition density q_t with respect to some dominating measure $\lambda_{t+1} \in \mathsf{M}(\mathcal{X}_{t+1})$. Then, for $t \in \mathbb{N}$ and $\eta \in \mathsf{M}_1(\mathcal{X}_t)$, we define the *backward kernel*

$$\overleftarrow{Q}_{t,\eta} : \mathsf{X}_{t+1} \times \mathcal{X}_t \ni (x_{t+1}, A) \mapsto \frac{\int \mathbb{1}_A(x_t) q_t(x_t, x_{t+1})\, \eta(\mathrm{d}x_t)}{\int q_t(x_t', x_{t+1})\, \eta(\mathrm{d}x_t')}. \tag{3.6}$$

Now, for $t \in \mathbb{N}^*$, denoting

$$B_t : \mathsf{X}_t \times \mathcal{X}_{0:t-1} \ni (x_t, A) \mapsto \int \cdots \int \mathbb{1}_A(x_{0:t-1}) \prod_{m=0}^{t-1} \overleftarrow{Q}_{m,\eta_m}(x_{m+1}, \mathrm{d}x_m), \tag{3.7}$$

we may state the following—now classical—*backward decomposition* of the Feynman–Kac path measures, a result that plays a pivotal role in the following.

Proposition 9. *For every $t \in \mathbb{N}^*$, it holds that $\gamma_{0:t} = \gamma_t \otimes B_t$ and $\eta_{0:t} = \eta_t \otimes B_t$.*

Although the decomposition in 9 is well known (see, *e.g.*, Del Moral et al. (2010); Del Moral et al. (2016)), we provide a proof in 3.6.1 for completeness. Using backward decomposition, we can obtain a particle approximation of a given Feynman–Kac path measure $\eta_{0:t}$ by first sampling, in an initial forward pass, particle clouds $\{\boldsymbol{\xi}_m\}_{m=0}^t$ from $\boldsymbol{\eta}_0 \otimes \boldsymbol{M}_0 \otimes \cdots \otimes \boldsymbol{M}_{t-1}$. Then, in a subsequent backward pass, we sample N conditionally independent paths $\{\tilde{\xi}_{0:t}^i\}_{i=1}^N$ from $\mathbb{B}_t(\boldsymbol{\xi}_0, \ldots, \boldsymbol{\xi}_t, \cdot)$, where

$$\mathbb{B}_t : \mathbf{X}_{0:t} \times \mathcal{X}_{0:t} \ni (\boldsymbol{x}_{0:t}, A) \mapsto \int \cdots \int \mathbb{1}_A(x_{0:t}) \left(\prod_{m=0}^{t-1} \overleftarrow{Q}_{m,\mu(\boldsymbol{x}_m)}(x_{m+1}, \mathrm{d}x_m) \right) \mu(\boldsymbol{x}_t)(\mathrm{d}x_t) \tag{3.8}$$

is a Markov kernel describing the time-reversed dynamics induced by the particle approximations generated in the forward pass. (Here, and in the following, we use blackboard notation to denote kernels related to many-body path spaces.) Finally, $\mu(\{\tilde{\xi}_{0:t}^i\}_{i=1}^N)h$ is returned as an estimator of $\eta_{0:t}h$ for any $\eta_{0:t}$-integrable test function h. This algorithm is referred to as the *forward-filtering backward-simulation* (`FFBSi`) *algorithm* in the literature, and was introduced in Godsill et al. (2004); see also Cappé et al. (2007); Douc et al. (2011). More precisely, given the forward particles $\{\boldsymbol{\xi}_m\}_{m=0}^t$, each path $\tilde{\xi}_{0:t}^i$ is generated by first drawing $\tilde{\xi}_t^i$ uniformly from among the particles $\boldsymbol{\xi}_t$ in the previous generation, and then drawing, recursively,

$$\tilde{\xi}_m^i \sim \overleftarrow{Q}_{m,\mu(\boldsymbol{\xi}_m)}(\tilde{\xi}_{m+1}^i, \cdot) = \sum_{j=1}^N \frac{q_m(\xi_m^j, \tilde{\xi}_{m+1}^i)}{\sum_{\ell=1}^N q_m(\xi_m^\ell, \tilde{\xi}_{m+1}^i)} \delta_{\xi_m^j} ; \tag{3.9}$$

that is, given $\tilde{\xi}_{m+1}^i$, $\tilde{\xi}_m^i$ is picked at random from among $\boldsymbol{\xi}_m$ based on weights proportional to $\{q_m(\xi_m^j, \tilde{\xi}_{m+1}^i)\}_{j=1}^N$. Note that in this basic formulation of the `FFBSi` algorithm, each backward-sampling operation (3.9) requires the computation of the normalising constant $\sum_{\ell=1}^N q_m(\xi_m^\ell, \tilde{\xi}_{m+1}^i)$, which implies an overall quadratic complexity of the algorithm. Still, this heavy computational burden can be eased by using an effective accept–reject technique, as discussed in 3.2.4.

3.2.3 Conditional dual processes and particle Gibbs

The *dual process* associated with a given Feynman–Kac model (3.1–3.2) and a given trajectory $\{z_t\}_{t\in\mathbb{N}}$, where $z_t \in \mathsf{X}_t$ for every $t \in \mathbb{N}$, is defined as the canonical Markov chain with kernels

$$\boldsymbol{M}_t\langle z_{t+1}\rangle : \mathbf{X}_t \times \boldsymbol{\mathcal{X}}_{t+1} \ni (\boldsymbol{x}_t, A) \mapsto \frac{1}{N} \sum_{i=0}^{N-1} \left(\Phi_t(\mu(\boldsymbol{x}_t))^{\otimes i} \otimes \delta_{z_{t+1}} \otimes \Phi_t(\mu(\boldsymbol{x}_t))^{\otimes(N-i-1)} \right)(A), \tag{3.10}$$

for $t \in \mathbb{N}$, and initial distribution

$$\boldsymbol{\eta}_0\langle z_0\rangle \coloneqq \frac{1}{N} \sum_{i=0}^{N-1} \left(\eta_0^{\otimes i} \otimes \delta_{z_0} \otimes \eta_0^{\otimes(N-i-1)} \right). \tag{3.11}$$

As is clear from (3.10–3.11), given $\{z_t\}_{t\in\mathbb{N}}$, a realization $\{\boldsymbol{\xi}_t\}_{t\in\mathbb{N}}$ of the dual process is generated as follows. At time zero, the process is initialized by inserting z_0 at a randomly selected position in the vector $\boldsymbol{\xi}_0$, while drawing independently the remaining elements in the same vector from η_0. After this, the process proceeds in a Markovian manner by, given $\boldsymbol{\xi}_t$, inserting z_{t+1} at a randomly selected position in $\boldsymbol{\xi}_{t+1}$, while drawing independently the remaining elements from $\Phi_t(\mu(\boldsymbol{\xi}_t))$.

In order to describe compactly the law of the conditional dual process, we define the Markov kernel

$$\mathbb{C}_t : \mathsf{X}_{0:t} \times \boldsymbol{\mathcal{X}}_{0:t} \ni (z_{0:t}, A) \mapsto \boldsymbol{\eta}_0\langle z_0\rangle \otimes \boldsymbol{M}_0\langle z_1\rangle \otimes \cdots \otimes \boldsymbol{M}_{t-1}\langle z_t\rangle(A).$$

The following result elegantly combines the underlying model (3.1–3.2), the many-body Feynman–Kac model, the backward decomposition, and the conditional dual process.

Theorem 10 (Del Moral et al. (2016))**.** *For all $t \in \mathbb{N}$, it holds that*

$$\mathbb{B}_t \otimes \boldsymbol{\gamma}_{0:t} = \gamma_{0:t} \otimes \mathbb{C}_t. \tag{3.12}$$

In Del Moral et al. (2016), each state $\boldsymbol{\xi}_t$ of the many-body process maps an outcome ω of the sample space Ω onto an *unordered set* of N elements in X_t. However, we have chosen to let each $\boldsymbol{\xi}_t$ take values in the standard *product space* X_t^N, for two reasons. First, the construction of Del Moral et al. (2016) requires sophisticated measure-theoretic arguments to endow such unordered sets with suitable σ-fields and appropriate measures. Second, we see no need to ignore the index order of the particles, as long as the Markovian dynamics (3.10–3.11) of the conditional dual process are symmetrized over the particle cloud. Therefore, in 3.6.2, we include our own proof of duality (3.12) for completeness. Note that the measure (3.12) on $\mathcal{X}_{0:t} \otimes \boldsymbol{\mathcal{X}}_{0:t}$ is unnormalized, but because the kernels $\mathbb{B}_t$ and $\mathbb{C}_t$ are both Markov, normalizing the identity with $\gamma_{0:t}(\mathsf{X}_{0:t}) = \boldsymbol{\gamma}_{0:t}(\boldsymbol{\mathsf{X}}_{0:t})$ immediately yields

$$\mathbb{B}_t \otimes \boldsymbol{\eta}_{0:t} = \eta_{0:t} \otimes \mathbb{C}_t. \tag{3.13}$$

Because the two sides of (3.13) provide the full conditionals, it is natural to take a data-augmentation approach, and sample the target (3.13) using a two-stage deterministic-scan Gibbs sampler Andrieu et al. (2010b); Chopin and Singh (2015). Specifically, assume we generate a state $(\boldsymbol{\xi}_{0:t}[\ell], \zeta_{0:t}[\ell])$ comprising a dual process with an associated path on the basis of $\ell \in \mathbb{N}$ iterations of the sampler. Then, we generate the next state $(\boldsymbol{\xi}_{0:t}[\ell+1], \zeta_{0:t}[\ell+1])$ in a Markovian fashion by first sampling $\boldsymbol{\xi}_{0:t}[\ell+1] \sim \mathbb{C}_t(\zeta_{0:t}[\ell], \cdot)$, and then sampling $\zeta_{0:t}[\ell+1] \sim \mathbb{B}_t(\boldsymbol{\xi}_{0:t}[\ell+1], \cdot)$. After arbitrary initialization (and the discard of possible burn-in), this procedure produces a Markov trajectory $\{(\boldsymbol{\xi}_{0:t}[\ell], \zeta_{0:t}[\ell])\}_{\ell\in\mathbb{N}}$, and under weak additional technical conditions, this Markov chain admits (3.13) as its unique invariant distribution. In such a case, the Markov chain is ergodic (Douc et al., 2018, Chapter 5), and the marginal distribution of the conditioning path $\zeta_{0:t}[\ell]$ converges to the target distribution $\eta_{0:t}$. Therefore, for every $h \in \mathsf{F}(\mathcal{X}_{0:t})$, it holds that $\lim_{L\to\infty} L^{-1} \sum_{\ell=1}^{L} h(\zeta_{0:t}[\ell]) = \eta_{0:t}h$, $\mathbb{P}$-a.s.. This algorithm is given in the discussion in Whiteley (2010) of the original particle Gibbs paper Andrieu et al. (2010b); however, the justification of Whiteley (2010), involving an extension of the law targeted by the particle Gibbs sampler to the ancestral indices of particles, differs somewhat from the one presented here.

3.2.4 The PARIS algorithm

In the following, we assume that we are given a sequence $\{h_t\}_{t\in\mathbb{N}}$ of additive state functionals of type (3.4). Interestingly, as noted in Cappé (2011); Del Moral et al. (2010), the backward decomposition allows, when applied to additive state functionals, a forward recursion for the expectations $\{\eta_{0:t}h_t\}_{t\in\mathbb{N}}$. More specifically, using the forward decomposition $h_{t+1}(x_{0:t+1}) = h_t(x_{0:t}) + \tilde{h}_t(x_t, x_{t+1})$ and the backward kernel B_{t+1} defined in (3.7), we may write, for $x_{t+1} \in \mathsf{X}_{t+1}$,

$$\begin{aligned} B_{t+1}h_{t+1}(x_{t+1}) &= \int \overleftarrow{Q}_{t,\eta_t}(x_{t+1}, \mathrm{d}x_t) \int \Big(h_t(x_{0:t}) + \tilde{h}_t(x_t, x_{t+1})\Big) B_t(x_t, \mathrm{d}x_{0:t-1}) \\ &= \overleftarrow{Q}_{t,\eta_t}(B_t h_t + \tilde{h}_t)(x_{t+1}), \end{aligned} \tag{3.14}$$

which, by 9, implies that

$$\eta_{0:t+1}h_{t+1} = \eta_{t+1}\overleftarrow{Q}_{t,\eta_t}(B_t h_t + \tilde{h}_t). \tag{3.15}$$

The marginal flow $\{\eta_t\}_{t\in\mathbb{N}}$ can be expressed recursively using the mappings $\{\Phi_t\}_{t\in\mathbb{N}}$. Thus, (3.15) provides, in principle, a basis for an online computation of $\{\eta_{0:t}h_t\}_{t\in\mathbb{N}}$. Because the marginals are generally intractable, following Del Moral et al. (2010), we plug particle approximations $\mu(\boldsymbol{\xi}_{t+1})$ and $\overleftarrow{Q}_{t,\mu(\boldsymbol{\xi}_t)}$ (see (3.9)) of η_{t+1} and $\overleftarrow{Q}_{t,\mu(\eta_t)}$, respectively, into the recursion (3.15). More precisely, we proceed recursively, and assume that at time t, we have a sample $\{(\xi_t^i, \beta_t^i)\}_{i=1}^N$ of particles with associated statistics, where each statistic β_t^i serves as an approximation of $B_t h_t(\xi_t^i)$. Then evolving the particle cloud according to $\boldsymbol{\xi}_{t+1} \sim \boldsymbol{M}_t(\boldsymbol{\xi}_t, \cdot)$ and updating the statistics using (3.14), with $\overleftarrow{Q}_{t,\eta_t}$ replaced by $\overleftarrow{Q}_{t,\mu(\boldsymbol{\xi}_t)}$, yields the particle-wise recursion

$$\beta_{t+1}^i = \sum_{\ell=1}^N \frac{q_t(\xi_t^\ell, \xi_{t+1}^i)}{\sum_{\ell'=1}^N q_t(\xi_t^{\ell'}, \xi_{t+1}^i)} \left(\beta_t^\ell + \tilde{h}_t(\xi_t^\ell, \xi_{t+1}^i)\right), \quad i \in [\![1, N]\!], \tag{3.16}$$

and, finally, the estimator

$$\mu(\boldsymbol{\beta}_t)(\mathrm{id}) = \frac{1}{N}\sum_{i=1}^N \beta_t^i \tag{3.17}$$

of $\eta_{0:t}h_t$, where we set $\boldsymbol{\beta}_t := (\beta_t^1, \ldots, \beta_t^N)$, for $i \in [\![1, N]\!]$, and id is the identity mapping. The procedure is initialized by simply letting $\beta_0^i = 0$, for all $i \in [\![1, N]\!]$. Note that (3.17) provides a particle interpretation of the backward decomposition in 9. This algorithm is a special case of the *forward-filtering backward-smoothing* (`FFBSm`) *algorithm* (see Andrieu and Doucet (2003); Godsill et al. (2004); Douc et al. (2011); Särkkä (2013)) for additive functionals satisfying (3.4). It allows for online processing of the sequence $\{\eta_{0:t}h_t\}_{t\in\mathbb{N}}$, but also has the appealing property that only the current particles $\boldsymbol{\xi}_t$ and statistics $\boldsymbol{\beta}_t$ need to be stored in memory. However, because each update (3.16) requires a summation of N terms, the scheme has an overall *quadratic* complexity in the number of particles, leading to a computational bottleneck in applications to complex models that require large particle sample sizes N.

To avoid the computational burden of this forward-only implementation of `FFBSm`, the `PARIS` algorithm Olsson and Westerborn (2017) updates the statistics $\boldsymbol{\beta}_t$ by replacing each sum (3.16) with the Monte Carlo estimate

$$\beta_{t+1}^i = \frac{1}{M}\sum_{j=1}^M \left(\tilde{\beta}_t^{i,j} + \tilde{h}_t(\tilde{\xi}_t^{i,j}, \xi_{t+1}^i)\right), \quad i \in [\![1, N]\!], \tag{3.18}$$

where $\{(\tilde{\xi}_t^{i,j}, \tilde{\beta}_t^{i,j})\}_{j=1}^M$ are drawn randomly from among $\{(\xi_t^i, \beta_t^i)\}_{i=1}^N$ with replacement, by assigning $(\tilde{\xi}_t^{i,j}, \tilde{\beta}_t^{i,j})$ the value of $(\xi_t^\ell, \beta_t^\ell)$ with probability $q_t(\xi_t^\ell, \xi_{t+1}^i)/\sum_{\ell'=1}^N q_t(\xi_t^{\ell'}, \xi_{t+1}^i)$, and the Monte Carlo sample size $M \in \mathbb{N}^*$ is much smaller than N (say, less than five). Formally,

$$\{(\tilde{\xi}_t^{i,j}, \tilde{\beta}_t^{i,j})\}_{j=1}^M \sim \left(\sum_{\ell=1}^N \frac{q_t(\xi_t^\ell, \xi_{t+1}^i)}{\sum_{\ell'=1}^N q_t(\xi_t^{\ell'}, \xi_{t+1}^i)}\delta_{(\xi_t^\ell, \beta_t^\ell)}\right)^{\otimes M}, \quad i \in [\![1, N]\!].$$

The resulting procedure, summarized in 7, allows for online processing with constant memory requirements, because it only needs to store the current particle cloud and the estimated auxiliary statistics at each iteration. Moreover, when the Markov transition densities of the model can be uniformly bounded, that is, there exists, for every $t \in \mathbb{N}$, an upper bound $\bar{\sigma}_t > 0$ such that for all $(x_t, x_{t+1}) \in \mathsf{X}_t \times \mathsf{X}_{t+1}$, $m_t(x_t, x_{t+1}) \leq \bar{\sigma}_t$ (a weak assumption satisfied for most models of interest), then we can generate a sample $(\tilde{\xi}_t^{i,j}, \tilde{\beta}_t^{i,j})$ by drawing, with replacement and until acceptance, candidates $(\tilde{\xi}_t^{i,*}, \tilde{\beta}_t^{i,*})$ from $\{(\xi_t^i, \beta_t^i)\}_{i=1}^N$ based on the normalized particle weights $\{g_t(\xi_t^\ell)/\sum_{\ell'} g_t(\xi_t^{\ell'})\}_{\ell=1}^N$ (obtained as a by-product in the generation of $\boldsymbol{\xi}_{t+1}$), and accepting the same with probability $m_t(\tilde{\xi}_t^{i,*}, \xi_{t+1}^i)/\bar{\sigma}_t$. Because

this sampling procedure bypasses the calculation of the normalizing constant $\sum_{\ell'=1}^{N} q_t(\xi_t^{\ell'}, \xi_{t+1}^i)$ of the targeted categorical distribution, it yields an overall $\mathcal{O}(MN)$ complexity of the algorithm; see Douc et al. (2011) for details.

Increasing M improves the accuracy of the algorithm at the cost of additional computational complexity.

As shown in Olsson and Westerborn (2017), there is a qualitative difference between the cases $M = 1$ and $M \geq 2$, and the latter is required to keep the PARIS numerically stable. More precisely, in the latter case, it can be shown that the PARIS estimator $\mu(\boldsymbol{\beta}_t)$ satisfies, as N tends to infinity while M is held fixed, a central limit theorem (CLT) at the rate $\sqrt{N}$, with an t-normalized asymptotic variance of order $\mathcal{O}(1 - 1/(M-1))$. As is clear from this bound, using a large M only wastes computational work, and setting M to two or three typically works well in practice.

3.3 The PPG sampler

We now introduce the PPG *algorithm*. For all $t \in \mathbb{N}^*$, let $\mathsf{Y}_t := \mathsf{X}_{0:t} \times \mathbb{R}$ and $\mathcal{Y}_t := \mathcal{X}_{0:t} \otimes \mathcal{B}(\mathbb{R})$. Moreover, let $\mathsf{Y}_0 := \mathsf{X}_0 \times \{0\}$ and $\mathcal{Y}_0 := \mathcal{X}_0 \otimes \{\{0\}, \emptyset\}$. An element of Y_t is always denoted by $y_t = (x_{0:t|t}, b_t)$. The PPG sampler includes, as a key ingredient, a *conditional PARIS step*, that recursively updates a set of Y_t-valued random variables $\upsilon_t^i := (\xi_{0:t|t}^i, \beta_t^i)$, for $i \in [\![1, N]\!]$. Let $(\boldsymbol{\upsilon}_t)_{t\in\mathbb{N}}$ denote the corresponding many-body process, with each $\boldsymbol{\upsilon}_t := ((\xi_{0:t|t}^1, \beta_t^1), \ldots, (\xi_{0:t|t}^N, \beta_t^N))$ taking on values in the space $\mathbf{Y}_t := \mathsf{Y}_t^N$, which we furnish with a σ-field $\boldsymbol{\mathcal{Y}}_t := \mathcal{Y}_t^{\otimes N}$. The space $\mathbf{Y}_0$ and the corresponding σ-field $\boldsymbol{\mathcal{Y}}_0$ are defined accordingly. For every $t \in \mathbb{N}$, we write $\boldsymbol{\xi}_{0:t|t} = (\xi_{0:t|t}^1, \ldots, \xi_{0:t|t}^N)$ for the collection of paths in $\boldsymbol{\upsilon}_t$, and $\boldsymbol{\xi}_{t|t} = (\xi_t^1, \ldots, \xi_t^N)$ for the collection of end points of the same.

In the following, we let $t \in \mathbb{N}$ be a fixed time horizon, and describe in detail how the PPG approximates $\eta_{0:t}h_t$ iteratively. In short, at each iteration ℓ, and given an input conditional path $\zeta_{0:t}[\ell]$, the PPG produces a many-body system $\boldsymbol{\upsilon}_t[\ell+1]$ by using a series of conditional PARIS operations. Then, an updated path $\zeta_{0:t}[\ell+1]$, which serves as input at the next iteration, is generated by picking one of the paths $\boldsymbol{\xi}_{0:t|t}[\ell+1]$ in $\boldsymbol{\upsilon}_t[\ell+1]$ at random. At each iteration, the produced statistics $\boldsymbol{\beta}_t$ (in $\boldsymbol{\upsilon}_t$) provide an approximation of $\eta_{0:t}h_t$, according to (3.17).

More precisely, given a path $\zeta_{0:t}[\ell]$, the conditional PARIS operations are executed as follows. In the initial step, $\boldsymbol{\xi}_{0|0}[\ell+1]$ are drawn from $\boldsymbol{\eta}_0\langle\zeta_0[\ell]\rangle$ defined in (3.11), and $\upsilon_0^i[\ell+1] \leftarrow (\xi_{0|0}^i[\ell+1], 0)$, for all $i \in [\![1, N]\!]$; then, recursively, for $m \in [\![0, t]\!]$, assuming access to $\boldsymbol{\upsilon}_m[\ell+1]$, we

(1) generate an updated particle cloud $\boldsymbol{\xi}_{m+1}[\ell+1] \sim \boldsymbol{M}_m\langle\zeta_{m+1}[\ell]\rangle(\boldsymbol{\xi}_{m|m}[\ell+1], \cdot)$,

(2) pick at random, for each $i \in [\![1, N]\!]$, an ancestor path with associated statistics $(\tilde{\xi}_{0:m}^{i,1}[\ell+1], \tilde{\beta}_m^{i,1}[\ell+1])$ from among $\boldsymbol{\upsilon}_m[\ell+1]$ by drawing

$$(\tilde{\xi}_{0:m}^{i,1}[\ell+1], \tilde{\beta}_m^{i,1}[\ell+1]) \sim \sum_{s=1}^{N} \frac{q_m(\xi_{m|m}^s[\ell+1], \xi_{m+1}^i[\ell+1])}{\sum_{s'=1}^{N} q_m(\xi_{m|m}^{s'}[\ell+1], \xi_{m+1}^i[\ell+1])} \delta_{\upsilon_m^s[\ell+1]},$$

(3) pick at random, for each $i \in [\![1, N]\!]$, with replacement, $M-1$ ancestor particles and associated statistics $\{(\tilde{\xi}_m^{i,j}[\ell+1], \tilde{\beta}_m^{i,j}[\ell+1])\}_{j=2}^M$ at random from $\{(\xi_{m|m}^s[\ell+1], \beta_m^s[\ell+1])\}_{s=1}^N$ according to

$$\{(\tilde{\xi}_m^{i,j}[\ell+1], \tilde{\beta}_m^{i,j}[\ell+1])\}_{j=2}^M \sim \left(\sum_{s=1}^{N} \frac{q_m(\xi_{m|m}^s[\ell+1], \xi_{m+1}^i[\ell+1])}{\sum_{s'=1}^{N} q_m(\xi_{m|m}^{s'}[\ell+1], \xi_{m+1}^i[\ell+1])} \delta_{(\xi_{m|m}^s[\ell+1], \beta_m^s[\ell+1])} \right)^{\otimes(M-1)},$$

(4) set, for all $i \in [\![1, N]\!]$, $\xi^i_{0:m+1|m+1}[\ell+1] \leftarrow (\tilde{\xi}^{i,1}_{0:m}[\ell+1], \xi^i_{m+1}[\ell+1])$ and $\upsilon^i_{m+1}[\ell+1] \leftarrow (\xi^i_{0:m+1|m+1}[\ell+1], \beta^i_{m+1}[\ell+1])$, where

$$\beta^i_{m+1}[\ell+1] \leftarrow M^{-1} \sum_{j=1}^{M} \left(\tilde{\beta}^{i,j}_m[\ell+1] + \tilde{h}_m(\tilde{\xi}^{i,j}_m[\ell+1], \xi^i_{m+1}[\ell+1]) \right).$$

This conditional PARIS procedure is summarized in pseudocode in 8 in B.2.

In addition to recursively propagating the statistics $\{\boldsymbol{\beta}_m[\ell+1]\}_{m=0}^t$ to form the final estimator, this scheme also recursively propagates the trajectories $\{\boldsymbol{\xi}_{0:m|m}[\ell+1]\}_{m=0}^t$ used as a pool of candidates for the updated conditional path $\zeta_{0:t}[\ell+1]$. Once we have the set $\boldsymbol{\upsilon}_t[\ell+1]$ of trajectories and associated statistics formed using t recursive conditional PARIS updates, we draw an updated path $\zeta_{0:t}[\ell+1]$ from $\mu(\boldsymbol{\xi}_{0:t|t}[\ell+1])$ (*i.e.*, uniformly among the elements of $\boldsymbol{\xi}_{0:t|t}[\ell+1]$). As a result, the updated conditional path $\zeta_{0:t}[\ell+1]$ and the statistics $\boldsymbol{\beta}_t[\ell+1]$ are statistically intertwined conditionally on the conditional dual particle process underpinning the algorithm. The main reason for this is to avoid computational waste. By letting the updated conditional path $\zeta_{0:t}[\ell+1]$ be formed by reusing the backward samples from those generated to form the statistics $\boldsymbol{\beta}_t[\ell+1]$ included in the estimator, our procedure optimizes available computational resources. The full PPG is summarized in pseudocode in 9 in B.2.

The following Markov kernels play an instrumental role in the following. For a given path $\{z_m\}_{m\in\mathbb{N}}$, the conditional PARIS update in 8 defines an inhomogeneous Markov chain on the spaces $\{(\mathbf{Y}_m, \boldsymbol{\mathcal{Y}}_m)\}_{m\in\mathbb{N}}$ with kernels

$$\mathbf{Y}_m \times \boldsymbol{\mathcal{Y}}_{m+1} \ni (\boldsymbol{y}_m, A) \mapsto \int \boldsymbol{M}_m\langle z_{m+1}\rangle(\boldsymbol{x}_{m|m}, \mathrm{d}\boldsymbol{x}_{m+1})\, \boldsymbol{S}_m(\boldsymbol{y}_m, \boldsymbol{x}_{m+1}, A), \quad m \in \mathbb{N},$$

where

$$\begin{aligned}
&\boldsymbol{S}_m : \mathbf{Y}_m \times \mathbf{X}_{m+1} \times \boldsymbol{\mathcal{Y}}_{m+1} \ni (\boldsymbol{y}_m, \boldsymbol{x}_{m+1}, A) \qquad (3.19)\\
&\mapsto \int \cdots \int \mathbb{1}_A \left(\left\{ \left((\tilde{x}^{i,1}_{0:m}, x^i_{m+1}), \frac{1}{M} \sum_{j=1}^{M} \left(\tilde{b}^{i,j}_m + \tilde{h}_m(\tilde{x}^{i,j}_m, x^i_{m+1}) \right) \right) \right\}_{i=1}^N \right)\\
&\quad \times \prod_{i=1}^N \left(\sum_{\ell=1}^N \frac{q_m(x^\ell_{m|m}, x^i_{m+1})}{\sum_{\ell'=1}^N q_m(x^{\ell'}_{m|m}, x^i_{m+1})} \delta_{y^\ell_m}(\mathrm{d}(\tilde{x}^{i,1}_{0:m}, \tilde{b}^{i,1}_m)) \right.\\
&\quad \left. \times \left(\sum_{\ell=1}^N \frac{q_m(x^\ell_{m|m}, x^i_{m+1})}{\sum_{\ell'=1}^N q_m(x^{\ell'}_{m|m}, x^i_{m+1})} \delta_{(x^\ell_{m|m}, b^\ell_m)} \right)^{\otimes(M-1)} (\mathrm{d}(\tilde{x}^{i,2}_m, \tilde{b}^{i,2}_m, \ldots, \tilde{x}^{i,M}_m, \tilde{b}^{i,M}_m)) \right).
\end{aligned}$$

In addition, we introduce the joint law

$$\begin{aligned}
&\mathbb{S}_t : \mathbf{X}_{0:t} \times \boldsymbol{\mathcal{Y}}_t \ni (\boldsymbol{x}_{0:t}, A)\\
&\qquad \mapsto \int \cdots \int \mathbb{1}_A(\boldsymbol{y}_t)\, \boldsymbol{S}_0(\mathbf{J}\boldsymbol{x}_0, \boldsymbol{x}_1, \mathrm{d}\boldsymbol{y}_1) \prod_{m=1}^{t-1} \boldsymbol{S}_m(\boldsymbol{y}_m, \boldsymbol{x}_{m+1}, \mathrm{d}\boldsymbol{y}_{m+1}), \qquad (3.20)
\end{aligned}$$

where we define $\mathbf{J} \coloneqq \mathrm{I}_N \otimes (0,1)^\intercal$.

The kernel $\mathbb{S}_t$ can be viewed as a *superincumbent sampling kernel* that describes the distribution of the output $\boldsymbol{\upsilon}_t$ generated by a sequence of PARIS iterations when the many-body process $\{\boldsymbol{\xi}_m\}_{m=0}^t$ associated with the underlying particle filter is given. This allows us to describe the PPG alternatively as follows: given $\zeta_{0:t}[\ell]$, draw $\boldsymbol{\xi}_{0:t}[\ell+1] \sim \mathbb{C}_t(\zeta_{0:t}[\ell], \cdot)$; then, draw $\boldsymbol{\upsilon}_t[\ell+1] \sim \mathbb{S}_t(\boldsymbol{\xi}_{0:t}[\ell+1], \cdot)$ and pick a trajectory $\zeta_{0:t}[\ell+1]$ from $\boldsymbol{\xi}_{0:t|t}[\ell+1]$ at random. The following proposition, establishes that the conditional distribution of $\zeta_{0:t}[\ell+1]$ given $\boldsymbol{\xi}_{0:t}[\ell+1]$ coincides, as expected, with the particle-induced backward dynamics $\mathbb{B}_t$.

Proposition 11. *For all $t \in \mathbb{N}^*$, $N \in \mathbb{N}^*$, $\boldsymbol{x}_{0:t} \in \mathbf{X}_{0:t}$, and $h \in \mathsf{F}(\mathcal{X}_{0:t})$,*

$$\int \mathbb{S}_t(\boldsymbol{x}_{0:t}, \mathrm{d}\boldsymbol{y}_t)\, \mu(\boldsymbol{x}_{0:t|t}) h = \mathbb{B}_t h(\boldsymbol{x}_{0:t}).$$

Finally, we define the Markov kernel induced by the PPG, as well as the extended probability distribution targeted by the same. For this purpose, we introduce the extended measurable space $(\mathbf{E}_t, \boldsymbol{\mathcal{E}}_t)$, with

$$\mathbf{E}_t \coloneqq \mathbf{Y}_t \times \mathsf{X}_{0:t}, \quad \boldsymbol{\mathcal{E}}_t \coloneqq \boldsymbol{\mathcal{Y}}_t \otimes \mathcal{X}_{0:t}.$$

The PPG described in 9 defines a Markov chain on $(\mathbf{E}_t, \boldsymbol{\mathcal{E}}_t)$ with the Markov transition kernel

$$\begin{aligned}\mathbb{K}_t : \mathbf{E}_t \times \boldsymbol{\mathcal{E}}_t \ni (\boldsymbol{y}_t, z_{0:t}, A) \\ \mapsto \iiint \mathbb{1}_A(\tilde{\boldsymbol{y}}_t, \tilde{z}_{0:t})\, \mathbb{C}_t(z_{0:t}, \mathrm{d}\tilde{\boldsymbol{x}}_{0:t})\, \mathbb{S}_t(\tilde{\boldsymbol{x}}_{0:t}, \mathrm{d}\tilde{\boldsymbol{y}}_t)\, \mu(\tilde{\boldsymbol{x}}_{0:t|t})(\mathrm{d}\tilde{z}_{0:t}). \quad (3.21)\end{aligned}$$

Note that the values of $\mathbb{K}_t$ defined above do not depend on $\boldsymbol{y}_t$, but only on $(z_{0:t}, A)$. For any given initial distribution $\xi \in \mathsf{M}_1(\mathcal{X}_{0:t})$, let $\mathbb{P}_\xi$ be the distribution of the canonical Markov chain induced by the kernel $\mathbb{K}_t$ and the initial distribution ξ. In the special case where $\xi = \delta_{z_{0:t}}$, for some given path $z_{0:t} \in \mathsf{X}_{0:t}$, we use the short-hand notation $\mathbb{P}_{\delta_{z_{0:t}}} = \mathbb{P}_{z_{0:t}}$. In addition, denote by

$$\begin{aligned}K_t : \mathsf{X}_{0:t} \times \mathcal{X}_{0:t} \ni (z_{0:t}, A) \\ \mapsto \iiint \mathbb{1}_A(\tilde{z}_{0:t})\, \mathbb{C}_t(z_{0:t}, \mathrm{d}\tilde{\boldsymbol{x}}_{0:t})\, \mathbb{S}_t(\tilde{\boldsymbol{x}}_{0:t}, \mathrm{d}\tilde{\boldsymbol{y}}_t)\, \mu(\tilde{\boldsymbol{x}}_{0:t|t})(\mathrm{d}\tilde{z}_{0:t}) \quad (3.22)\end{aligned}$$

the path-marginalized version of $\mathbb{K}_t$. By 11, it holds that $K_t = \mathbb{C}_t \mathbb{B}_t$, which shows that K_t coincides with the Markov transition kernel of the backward-sampling-based particle Gibbs sampler discussed in 3.2.3.

Finally, in order to prepare for the statement of our theoretical results on the PPG, we need to introduce the following Feynman–Kac path model *with a frozen path*. More precisely, for a given path $z_{0:t} \in \mathsf{X}_{0:t}$, define, for every $m \in [\![0, t-1]\!]$, the unnormalized kernel

$$Q_m\langle z_{m+1}\rangle : \mathsf{X}_m \times \mathcal{X}_{m+1} \ni (x_m, A) \mapsto \left(1 - \frac{1}{N}\right) Q_m(x_m, A) + \frac{1}{N} g_m(x_m)\, \delta_{z_{m+1}}(A)$$

and the initial distribution $\eta_0\langle z_0\rangle : \mathcal{X}_0 \ni A \mapsto (1 - 1/N)\eta_0(A) + \delta_{z_0}(A)/N$. Given these quantities, define, for $m \in [\![0, t]\!]$, $\gamma_m\langle z_{0:m}\rangle \coloneqq \eta_0\langle z_0\rangle Q_0\langle z_1\rangle \cdots Q_{m-1}\langle z_m\rangle$, and its normalized counterpart $\eta_m\langle z_{0:m}\rangle \coloneqq \gamma_m\langle z_{0:m}\rangle / \gamma_m\langle z_{0:m}\rangle \mathbb{1}_{\mathsf{X}_{0:m}}$. Finally, we introduce, for $m \in [\![0, t]\!]$, the kernels

$$B_m\langle z_{0:m-1}\rangle : \mathsf{X}_m \times \mathcal{X}_{0:m-1} \ni (x_m, A) \mapsto \int \cdots \int \mathbb{1}_A(x_{0:t-1}) \prod_{m=0}^{t-1} \overleftarrow{Q}_{m, \eta_m\langle z_{0:m}\rangle}(x_{m+1}, \mathrm{d}x_m)$$

and the path model $\eta_{0:m}\langle z_{0:m}\rangle \coloneqq B_m\langle z_{0:m-1}\rangle \otimes \eta_m\langle z_{0:m}\rangle$.

3.4 Main results

3.4.1 Theoretical results

In this section, we establish our main result, namely, the exponentially contracting bias bound stated in 12. This result is proved under the following strong mixing assumptions, which are standard in the literature (see Del Moral (2004); Douc and Moulines (2008); Del Moral (2013); Del Moral et al. (2016)):

A2 (strong mixing). *For every $t \in \mathbb{N}$, there exist $\underline{\tau}_t$, $\bar{\tau}_t$, $\underline{\sigma}_t$, and $\bar{\sigma}_t$ in $\mathbb{R}_+^*$ such that*

(i) $\underline{\tau}_t \leq g_t(x_t) \leq \bar{\tau}_t$ *for every* $x_t \in \mathsf{X}_t$,
(ii) $\underline{\sigma}_t \leq m_t(x_t, x_{t+1}) \leq \bar{\sigma}_t$ *for every* $(x_t, x_{t+1}) \in \mathsf{X}_{t:t+1}$.

Under 2, define, for every $t \in \mathbb{N}$,

$$\rho_t \coloneqq \max_{m \in [\![0,t]\!]} \frac{\bar{\tau}_m \bar{\sigma}_m}{\underline{\tau}_m \underline{\sigma}_m} \tag{3.23}$$

and, for every $t \in \mathbb{N}$ and $N \in \mathbb{N}^*$ such that $N > N_t \coloneqq (1 + 5\rho_t^2 t/2) \vee 2t(1 + 2\rho_t^2)$,

$$\kappa_{N,t} \coloneqq 1 - \frac{1 - (1 + 5t\rho_t^2/2)/N}{1 + 4t(1 + 2\rho_t^2)/N}. \tag{3.24}$$

Note that $\kappa_{N,t} \in (0, 1)$, for all N and t, as above.

Theorem 12. *Assume 2. Then, for every $t \in \mathbb{N}$, there exist c_t^{bias}, c_t^{mse}, and c_t^{cov} in $\mathbb{R}_+^*$ such that for every $M \in \mathbb{N}^*$, $\xi \in \mathsf{M}_1(\mathcal{X}_{0:t})$, $\ell \in \mathbb{N}^*$, $s \in \mathbb{N}^*$, and $N \in \mathbb{N}^*$ such that $N > N_t$,*

$$|\mathbb{E}_\xi \left[\mu(\boldsymbol{\beta}_t[\ell])(\mathrm{id})\right] - \eta_{0:t} h_t| \leq \mathsf{c}_t^{bias} \left(\sum_{m=0}^{t-1} \|\tilde{h}_m\|_\infty \right) N^{-1} \kappa_{N,t}^\ell, \tag{3.25}$$

$$\mathbb{E}_\xi \left[\left(\mu(\boldsymbol{\beta}_t[\ell])(\mathrm{id}) - \eta_{0:t} h_t \right)^2 \right] \leq \mathsf{c}_t^{mse} \left(\sum_{m=0}^{t-1} \|\tilde{h}_m\|_\infty \right)^2 N^{-1}, \tag{3.26}$$

$$\begin{aligned} &|\mathbb{E}_\xi \left[\left(\mu(\boldsymbol{\beta}_t[\ell])(\mathrm{id}) - \eta_{0:t} h_t \right) \left(\mu(\boldsymbol{\beta}_t[\ell + s])(\mathrm{id}) - \eta_{0:t} h_t \right) \right]| \\ &\qquad \leq \mathsf{c}_t^{cov} \left(\sum_{m=0}^{t-1} \|\tilde{h}_m\|_\infty \right)^2 N^{-3/2} \kappa_{N,t}^s. \end{aligned} \tag{3.27}$$

The constants c_t^{bias}, c_t^{mse}, and c_t^{cov} are given explicitly in the proof. Because we focus on the dependence on N and the index ℓ, we make no attempt to optimize the dependence of these constants on t in our proofs; nevertheless, we believe that it is possible to prove, under the stated assumptions, that this dependence is linear. The proof of the bound in 12 is based on four key ingredients. The first is the following unbiasedness property of the PARIS under the many-body Feynman–Kac path model.

Theorem 13. *For every $t \in \mathbb{N}$, $N \in \mathbb{N}^*$, and $\ell \in \mathbb{N}^*$,*

$$\mathbb{E}_{\eta_{0:t}} \left[\mu(\boldsymbol{\beta}_t[\ell])(\mathrm{id})\right] = \int \eta_{0:t} \mathbb{C}_t \mathbb{S}_t(\mathrm{d}\boldsymbol{b}_t)\, \mu(\boldsymbol{b}_t)(\mathrm{id}) = \int \boldsymbol{\eta}_{0:t} \mathbb{S}_t(\mathrm{d}\boldsymbol{b}_t)\, \mu(\boldsymbol{b}_t)(\mathrm{id}) = \eta_{0:t} h_t.$$

The proof of 13 is found in 3.6.3. The second is the uniform geometric ergodicity of the particle Gibbs with backward sampling established in Del Moral and Jasra (2018).

Theorem 14. *Assume 2. Then, for every $t \in \mathbb{N}$, $(\mu, \nu) \in \mathsf{M}_1(\mathcal{X}_{0:t})^2$, $\ell \in \mathbb{N}^*$, and $N \in \mathbb{N}^*$ such that $N > N_t$, $\|\mu K_t^\ell - \nu K_t^\ell\|_{\mathrm{TV}} \leq \kappa_{N,t} t N^\ell$, where $\kappa_{N,t}$ is defined in (3.24).*

As a third ingredient, we require the following uniform exponential concentration inequality of the conditional PARIS with respect to the frozen-path Feynman–Kac model defined in the previous section.

Theorem 15. *For every $t \in \mathbb{N}$, there exist $\mathsf{c}_t > 0$ and $\mathsf{d}_t > 0$ such that for every $M \in \mathbb{N}^*$, $z_{0:t} \in \mathsf{X}_{0:t}$, $N \in \mathbb{N}^*$, and $\varepsilon > 0$,*

$$\int \mathbb{C}_t \mathbb{S}_t(z_{0:t}, \mathrm{d}\boldsymbol{b}_t) \mathbb{1}\left\{ |\mu(\boldsymbol{b}_t)(\mathrm{id}) - \eta_{0:t} \langle z_{0:t} \rangle h_t| \geq \varepsilon \right\} \leq \mathsf{c}_t \exp\left(- \frac{\mathsf{d}_t N \varepsilon^2}{2(\sum_{m=0}^{t-1} \|\tilde{h}_m\|_\infty)^2} \right).$$

The proof of 15 is found in B.3.2, and is based on arguments similar to those used in the proofs of (Olsson and Westerborn, 2017, Theorem 1) and (Douc et al., 2011, Theorem 5) in the framework of the conditional dual process. 15 implies, in turn, the following conditional variance bound.

Proposition 16. *For every $t \in \mathbb{N}$, $M \in \mathbb{N}^*$, $z_{0:t} \in \mathsf{X}_{0:t}$, and $N \in \mathbb{N}^*$,*

$$\int \mathbb{C}_t \mathbb{S}_t(z_{0:t}, \mathrm{d}\boldsymbol{b}_t) \left|\mu(\boldsymbol{b}_t)(\mathrm{id}) - \eta_{0:t}\langle z_{0:t}\rangle h_t\right|^2 \leq \frac{\mathsf{c}_t}{\mathsf{d}_t} \left(\sum_{m=0}^{t-1} \|\tilde{h}_m\|_\infty\right)^2 N^{-1}.$$

Using 16, we deduce, in turn, the following bias bound, the proof is postponed to B.3.4.

Proposition 17. *For every $t \in \mathbb{N}$, there exists $\bar{\mathsf{c}}_t^{bias} > 0$ such that for every $M \in \mathbb{N}^*$, $z_{0:t} \in \mathsf{X}_{0:t}$, and $N \in \mathbb{N}^*$,*

$$\left|\int \mathbb{C}_t \mathbb{S}_t(z_{0:t}, \mathrm{d}\boldsymbol{b}_t)\, \mu(\boldsymbol{b}_t)(\mathrm{id}) - \eta_{0:t}\langle z_{0:t}\rangle h_t\right| \leq \bar{\mathsf{c}}_t^{bias} \left(\sum_{m=0}^{t-1} \|\tilde{h}_m\|_\infty\right) N^{-1}.$$

A fourth and last ingredient in the proof of 12 is the following bound on the discrepancy between the additive expectations under the original and frozen-path Feynman–Kac models. This bound is established using novel results in Gloaguen et al. (2022). More precisely, because for every $m \in \mathbb{N}$, $(x, z) \in \mathsf{X}_m^2$, $N \in \mathbb{N}^*$, and $h \in \mathsf{F}(\mathcal{X}_{m+1})$, using 2,

$$|Q_m\langle z\rangle h(x) - Q_m h(x)| \leq \frac{1}{N}\|g_m\|_\infty \|h\|_\infty \leq \frac{1}{N}\bar{\tau}_m \|h\|_\infty,$$

applying (Gloaguen et al., 2022, Theorem 4.3) yields the following.

Proposition 18. *Assume 2. Then, there exists $\mathsf{c} > 0$ such that for every $t \in \mathbb{N}$, $N \in \mathbb{N}$, and $z_{0:t} \in \mathsf{X}_{0:t}$,*

$$|\eta_{0:t}\langle z_{0:t}\rangle h_t - \eta_{0:t} h_t| \leq \mathsf{c} N^{-1} \sum_{m=0}^{t-1} \|\tilde{h}_m\|_\infty.$$

In addition, we assume $\sup_{t\in\mathbb{N}} \|\tilde{h}_t\|_\infty < \infty$ yields an $\mathcal{O}(n/N)$ bound in 18.

Finally, by combining these ingredients, we are now ready to present a proof of 12.

Proof of 12. Write, using the tower property,

$$\mathbb{E}_\xi\left[\mu(\boldsymbol{\beta}_t[\ell])(\mathrm{id})\right] = \mathbb{E}_\xi\left[\mathbb{E}_{\zeta_{0:t}[\ell]}\left[\mu(\boldsymbol{\beta}_t[0])(\mathrm{id})\right]\right] = \int \xi K_t^\ell \mathbb{C}_t \mathbb{S}_t(\mathrm{d}\boldsymbol{b}_t)\, \mu(\boldsymbol{b}_t)(\mathrm{id}).$$

Thus, by the unbiasedness property in 13,

$$\begin{aligned}
&\left|\mathbb{E}_\xi\left[\mu(\boldsymbol{\beta}_t[\ell])(\mathrm{id})\right] - \eta_{0:t} h_t\right| \\
&\qquad = \left|\int \xi K_t^\ell \mathbb{C}_t \mathbb{S}_t(\mathrm{d}\boldsymbol{b}_t)\, \mu(\boldsymbol{b}_t)(\mathrm{id}) - \int \eta_{0:t} \mathbb{C}_t \mathbb{S}_t(\mathrm{d}\boldsymbol{b}_t)\, \mu(\boldsymbol{b}_t)(\mathrm{id})\right| \\
&\qquad \leq \left\|\xi K_t^\ell - \eta_{0:t}\right\|_{\mathrm{TV}} \operatorname{osc}\left(\int \mathbb{C}_t \mathbb{S}_t(\cdot, \mathrm{d}\boldsymbol{b}_t)\, \mu(\boldsymbol{b}_t)(\mathrm{id})\right),
\end{aligned}$$

where, by 14, $\|\xi K_t^\ell - \eta_{0:t}\|_{\mathrm{TV}} \leq \kappa_{N,t}^\ell$. Moreover, to derive an upper bound on the oscillation, we consider the decomposition

$$\begin{aligned}
&\operatorname{osc}\left(\int \mathbb{C}_t \mathbb{S}_t(\cdot, \mathrm{d}\boldsymbol{b}_t)\, \mu(\boldsymbol{b}_t)(\mathrm{id})\right) \\
&\qquad \leq 2\left(\left\|\int \mathbb{C}_t \mathbb{S}_t(\cdot, \mathrm{d}\boldsymbol{b}_t)\, \mu(\boldsymbol{b}_t)(\mathrm{id}) - \eta_{0:t}\langle\cdot\rangle h_t\right\|_\infty + \left\|\eta_{0:t}\langle\cdot\rangle h_t - \eta_{0:t} h_t\right\|_\infty\right),
\end{aligned}$$

where the two terms on the right-hand side can be bounded using 18 and 17, respectively. This completes the proof of (3.25). We now consider the proof of (3.26). Writing

$$\mathbb{E}_\xi\left[(\mu(\boldsymbol{\beta}_t[\ell])(\mathrm{id}) - \eta_{0:t}h_t)^2\right] = \int \xi K_t^\ell(\mathrm{d}z_{0:t})\,\mathbb{C}_t\mathbb{S}_t(z_{0:t}, \mathrm{d}\boldsymbol{b}_t)\,(\mu(\boldsymbol{b}_t)(\mathrm{id}) - \eta_{0:t}h_t)^2\,,$$

we establish (3.26) using 16 and 18. Finally, WE consider (3.27). Using the Markov property, we obtain

$$\begin{aligned}\mathbb{E}_\xi\left[(\mu(\boldsymbol{\beta}_t[\ell])(\mathrm{id}) - \eta_{0:t}h_t)\,(\mu(\boldsymbol{\beta}_t[\ell+s])(\mathrm{id}) \;- \eta_{0:t}h_t)\right]& \\ = \mathbb{E}_\xi\left[(\mu(\boldsymbol{\beta}_t[\ell])(\mathrm{id}) - \eta_{0:t}h_t)\left(\mathbb{E}_{\zeta_{0:t}[\ell]}[\mu(\boldsymbol{\beta}_t[s])(\mathrm{id})] \;- \eta_{0:t}h_t\right)\right]&,\end{aligned}$$

from which we may deduce (3.27) using (3.25) and (3.26). □

3.4.2 The roll-out PPG estimator

In light of the previous results, it is natural to consider an estimator formed by an average across successive conditional PPG estimators $\{\mu(\boldsymbol{\beta}_t[\ell])\}_{\ell\in\mathbb{N}}$. To mitigate the bias, we remove a "burn-in" period, with length k_0 chosen proportionally to the mixing time of the particle Gibbs chain $\{\zeta_{0:t}[\ell]\}_{\ell\in\mathbb{N}^*}$. This yields the estimator

$$\Pi_{(k_0,k),N}(h_t) = (k-k_0)^{-1}\sum_{\ell=k_0+1}^{k}\mu(\boldsymbol{\beta}_t[\ell])(\mathrm{id}). \tag{3.28}$$

The total number of particles underlying this estimator is $C = (N-1)k$. We denote by $\upsilon = (k-k_0)/k$ the ratio of the number of particles used in the estimator to the total number of sampled particles.

As a final main result, we provide bounds on the bias and the MSE of the estimator (3.28). The proof is postponed to B.3.5.

Theorem 19. *Assume 2. Then, for every $t \in \mathbb{N}$, $M \in \mathbb{N}^*$, $\xi \in \mathsf{M}_1(\mathcal{X}_{0:t})$, $\ell \in \mathbb{N}^*$, $s \in \mathbb{N}^*$, and $N \in \mathbb{N}^*$ such that $N > N_t$,*

$$\left|\mathbb{E}_\xi[\Pi_{(k_0,k),N}(h_t)] - \eta_{0:t}h_t\right| \le \mathsf{c}_t^{bias}\left(\sum_{m=0}^{t-1}\|\tilde{h}_m\|_\infty\right)\frac{\kappa_{N,t}^{k_0}}{N(k-k_0)(1-\kappa_{N,t})}, \tag{3.29}$$

$$\begin{aligned}&\mathbb{E}_\xi\left[\left(\Pi_{(k_0,k),N}(h_t) - \eta_{0:t}h_t\right)^2\right] \\ &\qquad\le \left(\sum_{m=0}^{t-1}\|\tilde{h}_m\|_\infty\right)^2\frac{\mathsf{c}_t^{mse} + 2\mathsf{c}_t^{cov}N^{-1/2}(1-\kappa_{N,t})^{-1}}{N(k-k_0)}\end{aligned} \tag{3.30}$$

Setting the burn-in k_0 in the roll-out estimator is nontrivial. However, because the estimator converges for *any* choice of k_0, including the trivial choice $k_0 = 1$, we can view this algorithmic parameter as an opportunity for the user to optimize the implementation of the algorithm. For given (N,k), the choice of k_0 involves a classical trade-off between bias and variance; indeed, for fixed (N,k), the bias upper bound (3.29) decreases with k_0 proportionally to $\kappa_{N,t}^{k_0}/(k-k_0)$ whereas the MSE upper bound (3.30) increases with k_0 proportionally to $1/(k-k_0)$. These bounds suggest that we should take $k_0 = \lceil k(1-\ell^{-1})\rceil$ if we are willing to bound the MSE increase of the roll-out estimator by a factor ℓ with respect to the PARIS. However, the bias reduction is not easily quantified, because it depends mainly on the mixing rate $\kappa_{N,t}$ of the PPG chain, and we only have access to upper bounds on this rate that are, in general, too conservative.

3.5 Numerical results

In this section, we evaluate numerically the proposed PPG sampler in the context of general state-space HMMs. Given measurable spaces $(\mathsf{X}, \mathcal{X})$ and $(\mathsf{Z}, \mathcal{Z})$, an HMM is a bivariate (possibly inhomogeneous) Markov chain $\{(X_m, Z_m)\}_{m\in\mathbb{N}}$ taking values in the product space $(\mathsf{X}\times\mathsf{Z}, \mathcal{X}\otimes\mathcal{Z})$. In such a model, the process $\{X_t\}_{t\in\mathbb{N}}$, referred to as the *state sequence*, is assumed to be itself a (possibly inhomogeneous) Markov chain, specified by some initial distribution χ and some sequence $\{M_t\}_{t\in\mathbb{N}}$ of Markov kernels. The state sequence is latent and only partially observed through the *observation process* $\{Z_m\}_{m\in\mathbb{N}}$. Conditionally on the state sequence, the observations are assumed to be independent; furthermore, the conditional marginal distribution of each Z_m is assumed to depend only on the corresponding state X_m and to have a density $g_m(X_m, \cdot)$ with respect to some dominating measure. HMMs are used in numerous scientific and engineering disciplines; see Andrieu and Doucet (2002); Cappé et al. (2005a); Chopin and Papaspiliopoulos (2020). Inference in HMMs typically involves computing conditional distributions of unobserved states, given observations. Of particular interest are the sequence of *filter distributions*, where the filter at time $m \in \mathbb{N}$, denoted as η_m, is defined as the conditional distribution of X_m, given $Z_{0:m} \coloneqq (Z_0, \dots, Z_m)$, and the *joint-smoothing distributions*, where the joint-smoothing distribution at time m, denoted as $\eta_{0:m}$, is defined as the joint conditional distribution of the states $X_{0:m} = (X_0, \dots, X_m)$, given the observations $Z_{0:m}$. Consequently, η_m is the marginal of $\eta_{0:m}$ with respect to the last state X_m. Given a sequence $\{z_m\}_{m\in\mathbb{N}}$ of fixed observations, $\{\eta_{0:m}\}_{m\in\mathbb{N}}$ forms a Feynman–Kac model (see 3.1), with Markov kernels $\{M_m\}_{m\in\mathbb{N}}$ and potential functions $g_m \coloneqq g(\cdot, z_m)$, for $m \in \mathbb{N}$, on X.

We now evaluate the proposed algorithm numerically for two HMMs: (i) a linear Gaussian state-space model (for which the filter and the joint-smoothing distribution flows are available in a closed form), and (ii) the stochastic volatility model proposed in Hull and White (1987). The PPG algorithm used in this section is given in 9 (in B.2).

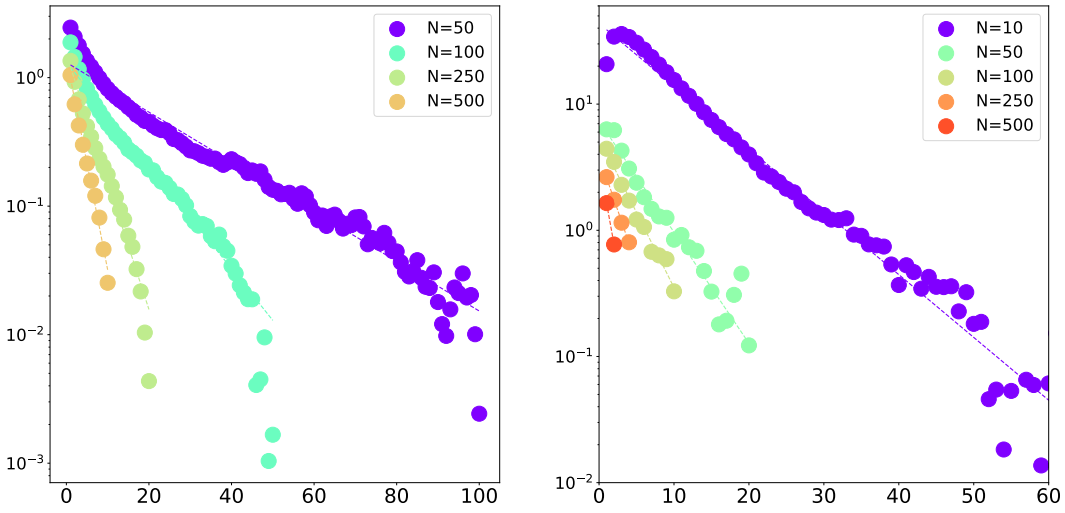
Linear Gaussian state-space model (LGSSM). We first consider an LGSSM

$$X_{m+1} = AX_m + Q\epsilon_{m+1}, \quad Z_m = BX_m + R\zeta_m, \quad m \in \mathbb{N}, \tag{3.31}$$

where $\{\epsilon_m\}_{m\in\mathbb{N}^*}$ and $\{\zeta_m\}_{m\in\mathbb{N}}$ are sequences of independent standard normally distributed random variables. The matrices A, Q, B, and R are assumed to be known 5×5 matrices (see section B.1.1 for the precise values). In this framework, we aim to compute the expectation of the *one-lag state covariance* $h_t(x_{0:t}) \coloneqq \sum_{m=0}^{t-1} x_m x_{m+1}^\intercal$ under the joint-smoothing distribution $\eta_{0:t}$ for observations generated by simulation under the given parameters with $t = 10^3$. In the LGSSM case, the *disturbance smoother* (see (Cappé et al., 2005a, Algorithm 5.2.15)) provides the exact values of $\eta_{0:t}h_t$, which allows us to assess numerically the bias of the PARIS and PPG estimators.

In this setting, we calculate the bias for batch sizes $N \in \{10, 25, 50, 100, 500\}$ and an increasing number k of iterations by averaging the PPG estimator over 10^4 independent runs. 3.1a shows the bias of the PPG estimates of the first diagonal entry of the one-lag covariance. For each batch size N, we estimate and display the regression function $k \mapsto \mathrm{e}^{ak+b}$ to illustrate the exponential decrease of the PPG bias, which is consistent with 12.

3.2a displays, for a given budget $C = 5\times 10^3$, the bias of the estimates of $\eta_{0:t}h_t$ using the PARIS and the PPG for different batch sizes N and different numbers $k = C/N$ of iterations and burn-in periods $k_0 = \lfloor k/2 \rfloor$. The red line corresponds to zero (no bias), and the empirical means are given by black-dashed lines. An extended comparison comprising different choices of k_0 and different budgets C is provided in B.1. In order to estimate the bias for each algorithmic configuration, we average 10^3 independent replications of the corresponding estimator. Moreover, to assess the precision of the resulting bias estimator, we repeat this procedure 10^2 times, and present the bias estimates in a box plot.



(a) LGSMM *(b) StoVol*

Figure 3.1: Output of the PPG roll-out estimator for the LGSSM (left panel) and the StoVol model (right panel). The curves describe the evolution of the bias with increasing k for different batch sizes N.

This enables us to form an idea of whether the PPG provides a statistically significant improvement in terms of bias. In this example, whatever the choice of the batch size is, the PPG bias is significantly reduced compared with the bias of the PARIS estimator. We further observe that a larger k leads to smaller bias.

Stochastic volatility (StoVol). As a second example, consider the stochastic volatility model

$$X_{m+1} = \phi X_m + \sigma_\epsilon \epsilon_{m+1}, \quad Z_m = \beta \exp(X_m/2)\zeta_m, \quad m \in \mathbb{N}, \tag{3.32}$$

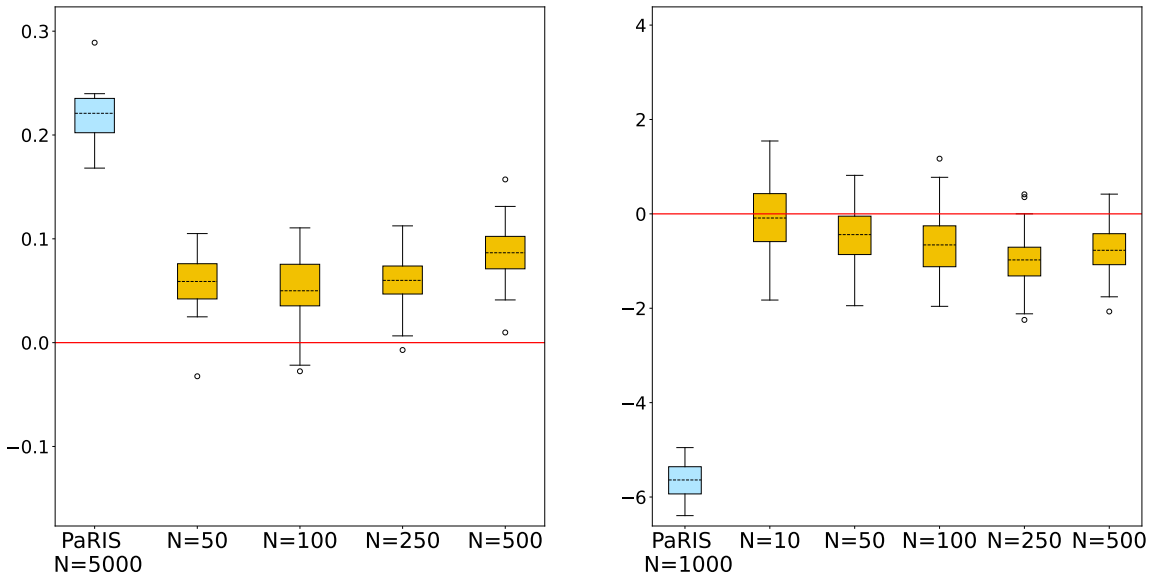
where $\{\epsilon_m\}_{m\in\mathbb{N}^*}$ and $\{\zeta_m\}_{m\in\mathbb{N}}$ are as in the previous example, and the model parameters ϕ, β, and σ_ϵ are set to 0.975, 0.63, and 0.16, respectively. The reference value is calculated by running the PARIS with 5×10^4 particles. In this setting, we repeated the experiments of the previous example for the same additive functional and number $t = 10^3$ of observations, produced by simulation under the parameters above. The computational budget was set to $C = 10^3$. As in the LGSSM example, the bias decay with respect to the iteration index k is displayed in 3.1b, and the comparison with the PARIS is shown in 3.2b. The comments from the previous example apply to this StoVol model context as well. More in-depth numerical assessments of the proposed PPG estimator are found in B.1.2. In particular, in B.1.2.1, we compare our estimator with the Rhee–Glynn-type estimator with ancestor sampling proposed by Jacob et al. (2020a), showing that the variance of the latter is significantly larger than that of the PPG for a given computational effort.

3.6 Proofs

3.6.1 Proof of 9

Using the identity

$$\eta_0 Q_0 \cdots Q_{t-1} \mathbb{1}_{\mathsf{X}_t} = \prod_{m=0}^{t-1} \eta_m Q_m \mathbb{1}_{\mathsf{X}_{m+1}}$$



(a) LGSSM *(b) StoVol*

Figure 3.2: `PARIS` and `PPG` bias dispersions for the LGSSM and StoVol model as a function of the mini-batch size N for fixed computational budgets $C = Nk$ of 5×10^3 (LGSSM) and 10^3 (StoVol model) and with $k_0 = \lfloor 2^{-1}k \rfloor$ burn-in steps.

and that each kernel Q_m has a transition density, write, for $h \in \mathsf{F}(\mathcal{X}_{0:t})$,

$$\begin{aligned}
\eta_{0:t}h &= \int \cdots \int h(x_{0:t})\, \eta_0(\mathrm{d}x_0) \prod_{m=0}^{t-1} \left(\frac{\eta_m[q_m(\cdot, x_{m+1})]\, \lambda_{m+1}(\mathrm{d}x_{m+1})}{\eta_m Q_m \mathbb{1}_{\mathsf{X}_{m+1}}} \right) \left(\frac{q_m(x_m, x_{m+1})}{\eta_m[q_m(\cdot, x_{m+1})]} \right) \\
&= \int \cdots \int h(x_{0:t})\, \eta_t(\mathrm{d}x_t) \prod_{m=0}^{t-1} \frac{\eta_m(\mathrm{d}x_m)\, q_m(x_m, x_{m+1})}{\eta_m[q_m(\cdot, x_{m+1})]} \\
&= \left(\overleftarrow{Q}_{0,\eta_0} \otimes \cdots \otimes \overleftarrow{Q}_{t-1,\eta_{t-1}} \otimes \eta_t \right) h,
\end{aligned} \tag{3.33}$$

which establishes the proof.

3.6.2 Proof of 10

Lemma 20. *For all $t \in \mathbb{N}$, $\boldsymbol{x}_t \in \mathbf{X}_t$, and $h \in \mathsf{F}(\boldsymbol{\mathcal{X}}_{t+1} \otimes \mathcal{X}_{t+1})$,*

$$\begin{aligned}
&\iint h(\boldsymbol{x}_{t+1}, z_{t+1})\, \boldsymbol{Q}_t(\boldsymbol{x}_t, \mathrm{d}\boldsymbol{x}_{t+1})\, \mu(\boldsymbol{x}_{t+1})(\mathrm{d}z_{t+1}) \\
&\qquad = \iint h(\boldsymbol{x}_{t+1}, z_{t+1})\, \mu(\boldsymbol{x}_t) Q_t(\mathrm{d}z_{t+1})\, \boldsymbol{M}_t\langle z_{t+1}\rangle(\boldsymbol{x}_t, \mathrm{d}\boldsymbol{x}_{t+1}).
\end{aligned} \tag{3.34}$$

In addition, for all $h \in \mathsf{F}(\boldsymbol{\mathcal{X}}_0 \otimes \mathcal{X}_0)$,

$$\iint h(\boldsymbol{x}_0, z_0)\, \boldsymbol{\eta}_0(\mathrm{d}\boldsymbol{x}_0)\, \mu(\boldsymbol{x}_0)(\mathrm{d}z_0) = \iint h(\boldsymbol{x}_0, z_0)\, \boldsymbol{\eta}_0\langle z_0\rangle(\mathrm{d}\boldsymbol{x}_0)\, \eta_0(\mathrm{d}z_0). \tag{3.35}$$

Proof. Because $\mu(\boldsymbol{x}_t)\, Q_t(\mathrm{d}z_{t+1}) = \boldsymbol{g}_t(\boldsymbol{x}_t)\, \Phi_t(\mu(\boldsymbol{x}_t))(\mathrm{d}z_{t+1})$, we may rewrite the right-hand side of

(3.34) as

$$\begin{aligned}
&\iint h(\boldsymbol{x}_{t+1}, z_{t+1})\, \mu(\boldsymbol{x}_t) Q_t(\mathrm{d}z_{t+1})\, \boldsymbol{M}_t\langle z_{t+1}\rangle(\boldsymbol{x}_t, \mathrm{d}\boldsymbol{x}_{t+1}) \\
&\qquad = \boldsymbol{g}_t(\boldsymbol{x}_t) \frac{1}{N} \sum_{i=0}^{N-1} \iint h(\boldsymbol{x}_{t+1}, z_{t+1})\, \Phi_t(\mu(\boldsymbol{x}_t))(\mathrm{d}z_{t+1}) \\
&\qquad\qquad \times \left(\Phi_t(\mu(\boldsymbol{x}_t))^{\otimes i} \otimes \delta_{z_{t+1}} \otimes \Phi_t(\mu(\boldsymbol{x}_t))^{\otimes(N-i-1)}\right)(\mathrm{d}\boldsymbol{x}_{t+1}) \\
&\qquad = \boldsymbol{g}_t(\boldsymbol{x}_t) \frac{1}{N} \sum_{i=1}^{N} \int \cdots \int h((x_{t+1}^1, \ldots, x_{t+1}^{i-1}, z_{t+1}, x_{t+1}^{i+1}, \ldots, x_{t+1}^N), z_{t+1}) \\
&\qquad\qquad \times \Phi_t(\mu(\boldsymbol{x}_t))(\mathrm{d}z_{t+1}) \prod_{\ell \neq i} \Phi_t(\mu(\boldsymbol{x}_t))(\mathrm{d}x_{t+1}^{\ell}) \\
&\qquad = \boldsymbol{g}_t(\boldsymbol{x}_t) \frac{1}{N} \sum_{i=1}^{N} \int h(\boldsymbol{x}_{t+1}, x_{t+1}^i)\, \boldsymbol{M}_t(\boldsymbol{x}_t, \mathrm{d}\boldsymbol{x}_{t+1}).
\end{aligned}$$

On the other hand, note that the left-hand side of (3.34) can be expressed as

$$\begin{aligned}
&\iint h(\boldsymbol{x}_{t+1}, z_{t+1})\, \boldsymbol{Q}_t(\boldsymbol{x}_t, \mathrm{d}\boldsymbol{x}_{t+1})\, \mu(\boldsymbol{x}_{t+1})(\mathrm{d}z_{t+1}) \\
&\qquad\qquad = \boldsymbol{g}_t(\boldsymbol{x}_t) \frac{1}{N} \sum_{i=1}^{N} \int h(\boldsymbol{x}_{t+1}, x_{t+1}^i)\, \boldsymbol{M}_t(\boldsymbol{x}_t, \mathrm{d}\boldsymbol{x}_{t+1}), \quad (3.36)
\end{aligned}$$

which establishes the identity. The identity (3.35) is established along similar lines. □

We establish 10 by induction. Thus, assume that the claim holds for t, and show that for all $h \in \mathsf{F}(\boldsymbol{\mathcal{X}}_{0:t+1} \otimes \mathcal{X}_{0:t+1})$,

$$\begin{aligned}
&\iint h(\boldsymbol{x}_{0:t+1}, z_{0:t+1})\, \boldsymbol{\gamma}_{0:t+1}(\mathrm{d}\boldsymbol{x}_{0:t+1})\, \mathbb{B}_{t+1}(\boldsymbol{x}_{0:t+1}, \mathrm{d}z_{0:t+1}) \\
&\qquad\qquad = \iint h(\boldsymbol{x}_{0:t+1}, z_{0:t+1})\, \gamma_{0:t+1}(\mathrm{d}z_{0:t+1})\, \mathbb{C}_{t+1}(z_{0:t+1}, \mathrm{d}\boldsymbol{x}_{0:t+1}). \quad (3.37)
\end{aligned}$$

To prove this, we process, using definition (3.8), the left-hand side of (3.37) according to

$$\begin{aligned}
&\iint h(\boldsymbol{x}_{0:t+1}, z_{0:t+1})\, \boldsymbol{\gamma}_{0:t+1}(\mathrm{d}\boldsymbol{x}_{0:t+1})\, \mathbb{B}_{t+1}(\boldsymbol{x}_{0:t+1}, \mathrm{d}z_{0:t+1}) \\
&= \iint \boldsymbol{\gamma}_{0:t}(\mathrm{d}\boldsymbol{x}_{0:t})\, \mathbb{B}_t(\boldsymbol{x}_{0:t}, \mathrm{d}z_{0:t}) \quad (3.38) \\
&\qquad\qquad \times \iint \bar{h}(\boldsymbol{x}_{0:t+1}, z_{0:t+1})\, \boldsymbol{Q}_t(\boldsymbol{x}_t, \mathrm{d}\boldsymbol{x}_{t+1})\, \mu(\boldsymbol{x}_{t+1})(\mathrm{d}z_{t+1}),
\end{aligned}$$

where we define the function

$$\bar{h}(\boldsymbol{x}_{0:t+1}, z_{0:t+1}) := \frac{q_t(z_t, z_{t+1}) h(\boldsymbol{x}_{0:t+1}, z_{0:t+1})}{\mu(\boldsymbol{x}_t)[q_t(\cdot, z_{t+1})]}.$$

Now, applying 20 to the inner integral and using

$$\mu(\boldsymbol{x}_t) Q_t(\mathrm{d}z_{t+1}) = \mu(\boldsymbol{x}_t)[q_t(\cdot, z_{t+1})]\, \lambda_{t+1}(\mathrm{d}z_{t+1})$$

yields, for every $\boldsymbol{x}_{0:t}$ and $z_{0:t}$,

$$\begin{aligned}
&\iint \bar{h}(\boldsymbol{x}_{0:t+1}, z_{0:t+1})\, \boldsymbol{Q}_t(\boldsymbol{x}_t, \mathrm{d}\boldsymbol{x}_{t+1})\, \mu(\boldsymbol{x}_{t+1})(\mathrm{d}z_{t+1}) \\
&= \iint \bar{h}(\boldsymbol{x}_{0:t+1}, z_{0:t+1})\, \mu(\boldsymbol{x}_t) Q_t(\mathrm{d}z_{t+1})\, \boldsymbol{M}_t\langle z_{t+1}\rangle(\boldsymbol{x}_t, \mathrm{d}\boldsymbol{x}_{t+1}) \\
&= \iint h(\boldsymbol{x}_{0:t+1}, z_{0:t+1})\, Q_t(z_t, \mathrm{d}z_{t+1})\, \boldsymbol{M}_t\langle z_{t+1}\rangle(\boldsymbol{x}_t, \mathrm{d}\boldsymbol{x}_{t+1}).
\end{aligned}$$

Inserting the previous identity into (3.38) and using the induction hypothesis yields

$$\begin{aligned}
&\iint h(\boldsymbol{x}_{0:t+1}, z_{0:t+1})\, \boldsymbol{\gamma}_{0:t+1}(\mathrm{d}\boldsymbol{x}_{0:t+1})\, \mathbb{B}_{t+1}(\boldsymbol{x}_{0:t+1}, \mathrm{d}z_{0:t+1}) \\
&= \iint \gamma_{0:t}(\mathrm{d}z_{0:t})\, \mathbb{C}_t(z_{0:t}, \mathrm{d}\boldsymbol{x}_{0:t}) \\
&\qquad\qquad \times \iint h(\boldsymbol{x}_{0:t+1}, z_{0:t+1})\, Q_t(z_t, \mathrm{d}z_{t+1})\, \boldsymbol{M}_t\langle z_{t+1}\rangle(\boldsymbol{x}_t, \mathrm{d}\boldsymbol{x}_{t+1}) \\
&= \iint h(\boldsymbol{x}_{0:t+1}, z_{0:t+1})\, \gamma_{0:t+1}(\mathrm{d}z_{0:t+1})\, \mathbb{C}_{t+1}(z_{0:t+1}, \mathrm{d}\boldsymbol{x}_{0:t+1}),
\end{aligned}$$

which establishes (3.37).

3.6.3 Proof of 13

First, define, for $m \in \mathbb{N}$,

$$\boldsymbol{P}_m : \mathbf{Y}_m \times \boldsymbol{\mathcal{Y}}_{m+1} \ni (\boldsymbol{y}_m, A) \mapsto \int \boldsymbol{M}_m(\boldsymbol{x}_{m|m}, \mathrm{d}\boldsymbol{x}_{m+1})\, \boldsymbol{S}_m(\boldsymbol{y}_m, \boldsymbol{x}_{m+1}, A). \tag{3.39}$$

For any given initial distribution $\boldsymbol{\psi}_0 \in \mathsf{M}_1(\boldsymbol{\mathcal{Y}}_0)$, let $\mathbb{P}^{\boldsymbol{P}}_{\boldsymbol{\psi}_0}$ be the distribution of the canonical Markov chain induced by the Markov kernels $\{\boldsymbol{P}_m\}_{m\in\mathbb{N}}$ and the initial distribution $\boldsymbol{\psi}_0$. With a slight abuse of notation we write, for $\boldsymbol{\eta}_0 \in \mathsf{M}_1(\boldsymbol{\mathcal{X}}_0)$, $\mathbb{P}^{\boldsymbol{P}}_{\boldsymbol{\eta}_0}$ instead of $\mathbb{P}^{\boldsymbol{P}}_{\boldsymbol{\psi}_0[\boldsymbol{\eta}_0]}$, where we define the extension $\boldsymbol{\psi}_0[\boldsymbol{\eta}_0](A) = \int \mathbb{1}_A(\mathbf{J}\boldsymbol{x}_0)\, \boldsymbol{\eta}_0(\mathrm{d}\boldsymbol{x}_0)$, for $A \in \boldsymbol{\mathcal{Y}}_0$. We preface the proof of 13 with some technical lemmas and a proposition.

Lemma 21. *For all $t \in \mathbb{N}$ and $(f_{t+1}, \tilde{f}_{t+1}) \in \mathsf{F}(\mathcal{X}_{t+1})^2$,*

$$\gamma_{t+1}(f_{t+1} B_{t+1} h_{t+1} + \tilde{f}_{t+1}) = \gamma_t\{Q_t f_{t+1} B_t h_t + Q_t(\tilde{h}_t f_{t+1} + \tilde{f}_{t+1})\}.$$

Proof. Pick arbitrary $\varphi \in \mathsf{F}(\mathcal{X}_{t:t+1})$ and, from definition (3.7) and that Q_t has a transition density, write

$$\begin{aligned}
&\iint \varphi(x_{t:t+1})\, \gamma_t(\mathrm{d}x_t)\, Q_t(x_t, \mathrm{d}x_{t+1}) \\
&\qquad = \iint \varphi(x_{t:t+1}) \gamma_t[q_t(\cdot, x_{t+1})]\, \lambda_{t+1}(\mathrm{d}x_{t+1})\, \frac{\gamma_t(\mathrm{d}x_t) q_t(x_t, x_{t+1})}{\gamma_t[q_t(\cdot, x_{t+1})]} \\
&\qquad = \iint \varphi(x_{t:t+1})\, \gamma_{t+1}(\mathrm{d}x_{t+1})\, \overleftarrow{Q}_{t,\eta_t}(x_{t+1}, \mathrm{d}x_t).
\end{aligned} \tag{3.40}$$

Now, by (3.14), it holds that

$$B_{t+1} h_{t+1}(x_{t+1}) = \int \overleftarrow{Q}_{t,\eta_t}(x_{t+1}, \mathrm{d}x_t) \left(\tilde{h}_t(x_{t:t+1}) + \int h_t(x_{0:t})\, B_t(x_t, \mathrm{d}x_{0:t-1})\right);$$

therefore, by applying (3.40) with

$$\varphi(x_{t:t+1}) \coloneqq f_{t+1}(x_{t+1}) \left(\tilde{h}_t(x_{t:t+1}) + \int h_t(x_{0:t})\, B_t(x_t, \mathrm{d}x_{0:t-1})\right),$$

we obtain that

$$\begin{aligned}\gamma_{t+1}(f_{t+1}B_{t+1}h_{t+1}) &= \iint \varphi(x_{t:t+1})\,\gamma_{t+1}(\mathrm{d}x_{t+1})\,\overleftarrow{Q}_{t,\eta_t}(x_{t+1},\mathrm{d}x_t)\\ &= \iint \varphi(x_{t:t+1})\,\gamma_t(\mathrm{d}x_t)\,Q_t(x_t,\mathrm{d}x_{t+1})\\ &= \gamma_t(Q_t f_{t+1}B_t h_t + Q_t\tilde{h}_t f_{t+1}).\end{aligned}$$

Now, the proof is concluded by noting that because $\gamma_{t+1} = \gamma_t Q_t$, $\gamma_{t+1}\tilde{f}_{t+1} = \gamma_t Q_t \tilde{f}_{t+1}$. □

Lemma 22. *For every $t \in \mathbb{N}^*$, $h_t \in \mathsf{F}(\mathcal{Y}_t)$, and $\boldsymbol{\eta}_0 \in \mathsf{M}_1(\boldsymbol{\mathcal{X}}_0)$, it holds that*

$$\mathbb{E}^{\boldsymbol{P}}_{\boldsymbol{\eta}_0}[h_t(\boldsymbol{v}_t) \mid \boldsymbol{\xi}_{0|0},\dots,\boldsymbol{\xi}_{t|t}] = \mathbb{S}_t h_t(\boldsymbol{\xi}_{0|0},\dots,\boldsymbol{\xi}_{t|t}), \quad \mathbb{P}^{\boldsymbol{P}}_{\boldsymbol{\eta}_0}\text{-a.s.}$$

Proof. Pick arbitrary $v_t \in \mathsf{F}(\mathcal{X}_{0:t})$. We show that

$$\mathbb{E}^{\boldsymbol{P}}_{\boldsymbol{\eta}_0}[v_t(\boldsymbol{\xi}_{0|0},\dots,\boldsymbol{\xi}_{t|t})h_t(\boldsymbol{v}_t)] = \mathbb{E}^{\boldsymbol{P}}_{\boldsymbol{\eta}_0}[v_t(\boldsymbol{\xi}_{0|0},\dots,\boldsymbol{\xi}_{t|t})\mathbb{S}_t h_t(\boldsymbol{\xi}_{0|0},\dots,\boldsymbol{\xi}_{t|t})], \tag{3.41}$$

from which the claim follows. Using definition (3.39), the left-hand side of the previous identity may be rewritten as

$$\begin{aligned}&\int\cdots\int \boldsymbol{\psi}_0[\boldsymbol{\eta}_0](\mathrm{d}\boldsymbol{y}_0)\prod_{m=0}^{t-1}\boldsymbol{P}_m(\boldsymbol{y}_m,\mathrm{d}\boldsymbol{y}_{m+1})\,h_t(\boldsymbol{y}_t)v_t(\boldsymbol{x}_{0|0},\dots,\boldsymbol{x}_{t|t})\\ &\quad= \int\cdots\int \boldsymbol{\eta}_0(\mathrm{d}\boldsymbol{x}_{0|0})\prod_{m=0}^{t-1}\boldsymbol{M}_m(\boldsymbol{x}_{m|m},\mathrm{d}\boldsymbol{x}_{m+1})\,\boldsymbol{S}_0(\mathbf{J}\boldsymbol{x}_{0|0},\boldsymbol{x}_1,\mathrm{d}\boldsymbol{y}_1)\\ &\qquad\times\prod_{m=0}^{t-1}\boldsymbol{S}_m(\boldsymbol{y}_m,\boldsymbol{x}_{m+1},\mathrm{d}\boldsymbol{y}_{m+1})\,h_t(\boldsymbol{y}_t)v_t(\boldsymbol{x}_{0|0},\dots,\boldsymbol{x}_{t|t})\\ &\quad= \int\cdots\int \boldsymbol{\eta}_0(\mathrm{d}\boldsymbol{x}_0)\prod_{m=0}^{t-1}\boldsymbol{M}_m(\boldsymbol{x}_m,\mathrm{d}\boldsymbol{x}_{m+1})\,\boldsymbol{S}_0(\mathbf{J}\boldsymbol{x}_0,\boldsymbol{x}_1,\mathrm{d}\boldsymbol{y}_1)\\ &\qquad\times\prod_{m=0}^{t-1}\boldsymbol{S}_m(\boldsymbol{y}_m,\boldsymbol{x}_{m+1},\mathrm{d}\boldsymbol{y}_{m+1})\,h_t(\boldsymbol{y}_t)v_t(\boldsymbol{x}_0,\dots,\boldsymbol{x}_t).\end{aligned}$$

Thus, we conclude the proof by using the definition (3.20) of $\mathbb{S}_t$, together with Fubini's theorem. □

Lemma 23. *For every $t \in \mathbb{N}^*$ and $h_t \in \mathsf{F}(\mathcal{Y}_t)$, it holds that*

$$\mathbb{E}_{\boldsymbol{\eta}_0}\left[\left(\prod_{m=0}^{t-1}\boldsymbol{g}_m(\boldsymbol{\xi}_{m|m})\right)h_t(\boldsymbol{v}_t)\right] = \int \boldsymbol{\gamma}_{0:t}\mathbb{S}_t(\mathrm{d}\boldsymbol{y}_t)\,h_t(\boldsymbol{y}_t).$$

Proof. The claim of the lemma is a direct implication of 22; indeed, by applying the tower property and the latter, we obtain

$$\begin{aligned}&\mathbb{E}^{\boldsymbol{P}}_{\boldsymbol{\eta}_0}\left[\left(\prod_{m=0}^{t-1}\boldsymbol{g}_m(\boldsymbol{\xi}_{m|m})\right)h_t(\boldsymbol{v}_t)\right]\\ &\quad= \mathbb{E}^{\boldsymbol{P}}_{\boldsymbol{\eta}_0}\left[\left(\prod_{m=0}^{t-1}\boldsymbol{g}_m(\boldsymbol{\xi}_{m|m})\right)\mathbb{S}_t h_t(\boldsymbol{\xi}_{0|0},\dots,\boldsymbol{\xi}_{t|t})\right]\\ &\quad= \int\cdots\int \boldsymbol{\eta}_0(\mathrm{d}\boldsymbol{x}_0)\prod_{m=0}^{t-1}\boldsymbol{g}_m(\boldsymbol{x}_m)\,\boldsymbol{M}_m(\boldsymbol{x}_m,\mathrm{d}\boldsymbol{x}_{m+1})\,\mathbb{S}_t h_t(\boldsymbol{x}_{0:t})\\ &\quad= \int \boldsymbol{\gamma}_{0:t}\mathbb{S}_t(\mathrm{d}\boldsymbol{y}_t)\,h_t(\boldsymbol{y}_t).\end{aligned}$$

□

Proposition 24. *For all $t \in \mathbb{N}^*$, $(N, M) \in (\mathbb{N}^*)^2$, and $(f_t, \tilde{f}_t) \in \mathsf{F}(\mathcal{X}_t)^2$,*

$$\int \boldsymbol{\gamma}_{0:t} \mathbb{S}_t(\mathrm{d}\boldsymbol{y}_t) \left(\frac{1}{N} \sum_{i=1}^{N} \{ b_t^i f_t(x_{t|t}^i) + \tilde{f}_t(x_{t|t}^i) \} \right) = \gamma_t(f_t B_t h_t + \tilde{f}_t).$$

Proof. Applying 23 yields

$$\int \boldsymbol{\gamma}_{0:t} \mathbb{S}_t(\mathrm{d}\boldsymbol{y}_t) \left(\frac{1}{N} \sum_{i=1}^{N} \{ b_t^i f_t(x_{t|t}^i) + \tilde{f}_t(x_{t|t}^i) \} \right) = \mathbb{E}_{\boldsymbol{\eta}_0}^{\boldsymbol{P}} \left[\left(\prod_{m=0}^{t-1} \boldsymbol{g}_m(\boldsymbol{\xi}_{m|m}) \right) \frac{1}{N} \sum_{i=1}^{N} \{ \beta_t^i f_t(\xi_{t|t}^i) + \tilde{f}_t(\xi_{t|t}^i) \} \right]. \quad (3.42)$$

In the following, we repeatedly use the following filtrations. Let $\tilde{\mathcal{F}}_t \coloneqq \sigma(\{\boldsymbol{\upsilon}_m\}_{m=0}^t)$ be the σ-field generated by the output of the PARIS (7) during the first t iterations. In addition, let $\mathcal{F}_t \coloneqq \tilde{\mathcal{F}}_{t-1} \vee \sigma(\boldsymbol{\xi}_{t|t})$. We proceed by induction. Thus, assume that the statement of the proposition holds for a given $t \in \mathbb{N}^*$, and consider, for arbitrarily chosen $(f_{t+1}, \tilde{f}_{t+1}) \in \mathsf{F}(\mathcal{X}_{t+1})^2$,

$$\mathbb{E}_{\boldsymbol{\eta}_0}^{\boldsymbol{P}} \left[\left(\prod_{m=0}^{t} \boldsymbol{g}_m(\boldsymbol{\xi}_{m|m}) \right) \frac{1}{N} \sum_{i=1}^{N} \{ \beta_{t+1}^i f_{t+1}(\xi_{t+1|t+1}^i) + \tilde{f}_{t+1}(\xi_{t+1|t+1}^i) \} \mid \tilde{\mathcal{F}}_t \right] = \left(\prod_{m=0}^{t} \boldsymbol{g}_m(\boldsymbol{\xi}_{m|m}) \right) \mathbb{E}_{\boldsymbol{\eta}_0}^{\boldsymbol{P}} [\beta_{t+1}^1 f_{t+1}(\xi_{t+1|t+1}^1) + \tilde{f}_{t+1}(\xi_{t+1|t+1}^1) \mid \tilde{\mathcal{F}}_t],$$

where we use that the variables $\{\beta_{t+1}^i f_{t+1}(\xi_{t+1|t+1}^i) + \tilde{f}_{t+1}(\xi_{t+1|t+1}^i)\}_{i=1}^N$ are conditionally independent and identically distributed (i.i.d.) given $\tilde{\mathcal{F}}_t$. Note that, by symmetry,

$$\begin{aligned} \mathbb{E}_{\boldsymbol{\eta}_0}^{\boldsymbol{P}} \left[\beta_{t+1}^1 \mid \mathcal{F}_{t+1} \right] &= \int \boldsymbol{S}_t(\boldsymbol{\upsilon}_t, \boldsymbol{\xi}_{t+1|t+1}, \mathrm{d}\boldsymbol{y}_{t+1})\, b_{t+1}^1 \\ &= \int \cdots \int \left(\prod_{j=1}^{M} \sum_{\ell=1}^{N} \frac{q_t(\xi_{t|t}^\ell, \xi_{t+1|t+1}^1)}{\sum_{\ell'=1}^{N} q_t(\xi_{t|t}^{\ell'}, \xi_{t+1|t+1}^1)} \delta_{(\xi_{t|t}^\ell, \beta_t^\ell)}(\mathrm{d}\tilde{x}_t^{1,j}, \mathrm{d}\tilde{b}_t^{1,j}) \right) \\ &\qquad \times \frac{1}{M} \sum_{j=1}^{M} \left(\tilde{b}_t^{1,j} + \tilde{h}_t(\tilde{x}_t^{1,j}, \xi_{t+1|t+1}^1) \right) \\ &= \sum_{\ell=1}^{N} \frac{q_t(\xi_{t|t}^\ell, \xi_{t+1|t+1}^1)}{\sum_{\ell'=1}^{N} q_t(\xi_{t|t}^{\ell'}, \xi_{t+1|t+1}^1)} \left(\beta_t^\ell + \tilde{h}_t(\xi_{t|t}^\ell, \xi_{t+1|t+1}^1) \right). \end{aligned} \quad (3.43)$$

Thus, using the tower property,

$$\mathbb{E}_{\boldsymbol{\eta}_0}^{\boldsymbol{P}} \left[\beta_{t+1}^1 f_{t+1}(\xi_{t+1|t+1}^1) \mid \tilde{\mathcal{F}}_t \right] = \int \Phi_t(\mu(\boldsymbol{\xi}_{t|t}))(\mathrm{d}x_{t+1})\, f_{t+1}(x_{t+1}) \sum_{\ell=1}^{N} \frac{q_t(\xi_{t|t}^\ell, x_{t+1})}{\sum_{\ell'=1}^{N} q_t(\xi_{t|t}^{\ell'}, x_{t+1})} \left(\beta_t^\ell + \tilde{h}_t(\xi_{t|t}^\ell, x_{t+1}) \right),$$

and, consequently, using definition (3.5),

$$\begin{aligned}
&\left(\prod_{m=0}^{t} \boldsymbol{g}_m(\boldsymbol{\xi}_{m|m})\right) \mathbb{E}^{\boldsymbol{P}}_{\boldsymbol{\eta}_0}\left[\beta^1_{t+1} f_{t+1}(\xi^1_{t+1|t+1}) \mid \tilde{\mathcal{F}}_t\right] \\
&= \left(\prod_{m=0}^{t-1} \boldsymbol{g}_m(\boldsymbol{\xi}_{m|m})\right) \int \frac{1}{N}\sum_{i=1}^{N} q_t(\xi^i_{t|t}, x_{t+1}) \\
&\quad \times f_{t+1}(x_{t+1}) \sum_{\ell=1}^{N} \frac{q_t(\xi^\ell_{t|t}, x_{t+1})}{\sum_{\ell'=1}^{N} q_t(\xi^{\ell'}_{t|t}, x_{t+1})} \left(\beta^\ell_t + \tilde{h}_t(\xi^\ell_{t|t}, x_{t+1})\right) \lambda_{t+1}(\mathrm{d}x_{t+1}) \\
&= \left(\prod_{m=0}^{t-1} \boldsymbol{g}_m(\boldsymbol{\xi}_{m|m})\right) \frac{1}{N}\sum_{\ell=1}^{N} \left(\beta^\ell_t Q_t f_{t+1}(\xi^\ell_{t|t}) + Q_t(\tilde{h}_t f_{t+1})(\xi^\ell_{t|t})\right).
\end{aligned}$$

Thus, applying the induction hypothesis,

$$\begin{aligned}
&\mathbb{E}^{\boldsymbol{P}}_{\boldsymbol{\eta}_0}\left[\left(\prod_{m=0}^{t} \boldsymbol{g}_m(\boldsymbol{\xi}_{m|m})\right) \frac{1}{N}\sum_{i=1}^{N} \beta^i_{t+1} f_{t+1}(\xi^i_{t+1|t+1})\right] \\
&= \mathbb{E}^{\boldsymbol{P}}_{\boldsymbol{\eta}_0}\left[\left(\prod_{m=0}^{t-1} \boldsymbol{g}_m(\boldsymbol{\xi}_{m|m})\right) \frac{1}{N}\sum_{\ell=1}^{N} \left(\beta^\ell_t Q_t f_{t+1}(\xi^\ell_{t|t}) + Q_t(\tilde{h}_t f_{t+1})(\xi^\ell_{t|t})\right)\right] \\
&= \gamma_t \left(Q_t f_{t+1} B_t h_t + Q_t(\tilde{h}_t f_{t+1})\right).
\end{aligned} \tag{3.44}$$

In the same manner, it can be shown that

$$\mathbb{E}^{\boldsymbol{P}}_{\boldsymbol{\eta}_0}\left[\left(\prod_{m=0}^{t} \boldsymbol{g}_m(\boldsymbol{\xi}_{m|m})\right) \frac{1}{N}\sum_{i=1}^{N} \tilde{f}_{t+1}(\xi^i_{t+1|t+1})\right] = \gamma_t Q_t \tilde{f}_{t+1}. \tag{3.45}$$

Now, by (3.44–3.45) and 21,

$$\begin{aligned}
&\mathbb{E}^{\boldsymbol{P}}_{\boldsymbol{\eta}_0}\left[\left(\prod_{m=0}^{t} \boldsymbol{g}_m(\boldsymbol{\xi}_{m|m})\right) \frac{1}{N}\sum_{i=1}^{N} \{\beta^i_{t+1} f_{t+1}(\xi^i_{t+1|t+1}) + \tilde{f}_{t+1}(\xi^i_{t+1|t+1})\}\right] \\
&= \gamma_t \left(Q_t f_{t+1} B_t h_t + Q_t(\tilde{h}_t f_{t+1} + Q_t \tilde{f}_{t+1})\right) \\
&= \gamma_{t+1}(f_{t+1} B_{t+1} h_{t+1} + \tilde{f}_{t+1}),
\end{aligned}$$

which shows that the claim of the proposition holds at time $t+1$.

It remains to check the base case $t = 0$, which holds trivially, because $\boldsymbol{\beta}_0 = \mathbf{0}$ and $B_0 h_0 = 0$ by convention, and the initial particles $\boldsymbol{\xi}_{0|0}$ are drawn from $\boldsymbol{\eta}_0$. This completes the proof. □

Proof of 13. The identity $\int \boldsymbol{\eta}_{0:t}(\mathrm{d}\boldsymbol{x}_{0:t})\, \mathbb{S}_t(\boldsymbol{x}_{0:t}, \mathrm{d}\boldsymbol{b}_t)\, \mu(\boldsymbol{b}_t)(\mathrm{id}) = \eta_{0:t} h_t$ follows immediately by letting $f_t \equiv 1$ and $\tilde{f}_t \equiv 0$ in 24, and using that $\boldsymbol{\gamma}_{0:t}(\mathbf{X}_{0:t}) = \gamma_{0:t}(\mathsf{X}_{0:t})$. Moreover, applying 10 yields

$$\begin{aligned}
\int \eta_{0:t} \mathbb{C}_t \mathbb{S}_t(\mathrm{d}\boldsymbol{b}_t)\, \mu(\boldsymbol{b}_t)(\mathrm{id}) &= \iint \eta_{0:t}(\mathrm{d}z_{0:t})\, \mathbb{C}_t(z_{0:t}, \mathrm{d}\boldsymbol{x}_{0:t}) \int \mathbb{S}_t(\boldsymbol{x}_{0:t}, \mathrm{d}\boldsymbol{b}_t)\, \mu(\boldsymbol{b}_t)(\mathrm{id}) \\
&= \iint \boldsymbol{\eta}_{0:t}(\mathrm{d}\boldsymbol{x}_{0:t})\, \mathbb{B}_t(\boldsymbol{x}_{0:t}, \mathrm{d}z_{0:t}) \int \mathbb{S}_t(\boldsymbol{x}_{0:t}, \mathrm{d}\boldsymbol{b}_t)\, \mu(\boldsymbol{b}_t)(\mathrm{id}) \\
&= \int \boldsymbol{\eta}_{0:t} \mathbb{S}_t(\mathrm{d}\boldsymbol{b}_t)\, \mu(\boldsymbol{b}_t)(\mathrm{id}).
\end{aligned}$$

Finally, the first identity holds because K_t leaves $\eta_{0:t}$ invariant. □

Chapter 4

Parameter learning with PPG

4.1 Parameter learning with PPG

We now turn to parameter learning using PPG and gradient-based methods. We set the focus on learning the parameter θ of a function $V(\theta)$ whose gradient is the smoothed expectation of an additive functional $s_{0:t,\theta}$ in the form (3.4). Note that θ can include parameters of $\{M_n\}_{n\in\mathbb{N}}$ and $\{g_n\}_{n\in\mathbb{N}}$, thus we add a θ subscript to all the quantities related to the associated *Feynman-Kac path measures* defined in chapter 3. Algorithm 3 defines a stochastic approximation (SA) scheme where the noise forms a parameter dependent Markov chain with associated invariant measure π_θ. We follow the approach of Karimi et al. (2019) to establish a non-asymptotic bound over the mean field $h(\theta) := \pi_\theta s_{0:t,\theta}$. Such a setting encompasses for instance the following estimation procedures.

(1) *Score ascent.* In the case of fully dominated HMMs, we are often interested in optimizing the log-likelihood of the observations given by $V(\theta) = \log \int \gamma_{0:t,\theta}(\mathrm{d}x_{0:t})$. By applying *Fisher's identity*, we may express its gradient as a smoothed expectation of an additive functional according to

$$\begin{aligned}\nabla_\theta V(\theta) &= \int \nabla_\theta \log \gamma_{0:t}(x_{0:t})\, \eta_{0:t,\theta}(\mathrm{d}x_{0:t}),\\ &= \int \sum_{\ell=0}^{t-1} s_{\ell,\theta}(x_\ell, x_{\ell+1})\, \eta_{0:t,\theta}(\mathrm{d}x_{0:t}),\end{aligned}$$

where $s_{\ell,\theta} : \mathsf{X}_{\ell:\ell+1} \ni (x, x') \mapsto \nabla_\theta \log\{g_{\ell,\theta}(x) m_{\ell,\theta}(x, x')\}$ and $s_{0:t,\theta} := \sum_{\ell=0}^{t-1} s_{\ell,\theta}$.

(2) *Backward KL surrogates.* Inspired by Naesseth et al. (2020), we may consider the problem of learning a surrogate model for $\eta_{0:t,\theta}$ in the form $q_\phi(x_{0:t}) = q_\phi(x_0) \prod_{\ell=0}^{t-1} q_\phi(x_{\ell+1}, x_\ell)$ by minimizing $V(\phi) = \mathrm{KL}(\eta_{0:t,\theta}, q_\phi)$.

Algorithm 2 Gradient estimation with roll-out PPG ($\widehat{\mathsf{Gd}}$)

Input: $\theta, \zeta_{0:t}[0]$, $s_{0:t,\theta}$, number k of PPG iterations, burn-in k_0.
Result: $\beta_t^{1:N}[k_0 : k], \zeta_{0:t}[k]$

```
for ℓ ← 0 to k − 1 do
    (β̃_t^{1:N}[ℓ + 1], ζ_{0:t}[ℓ + 1]) ← PPG(θ; ζ_{0:t}[ℓ], s_{0:t,θ})
    if ℓ ≥ k_0 − 1 then
        set β_t^{1:N}[ℓ + 1] = β̃_t^{1:N}[ℓ + 1]
```

Note that Algorithm 2 is simply algorithm 9 wih $s_{0:t,\theta}$ as the additive functional. For convenience, we recall the definition of the PPG kernel introduced in chapter 3. For $(k_0, k) \in (\mathbb{N}^*)^2$ such that $k_0 < k$, we

Algorithm 3 Score ascent with PPG.

Input: θ_0, $\zeta_{0:t}[0]$, number k of PPG iterations, burn-in k_0, number of SA iterations n, learning-rate sequence $\{\gamma_\ell\}_{\ell\in\mathbb{N}}$.

Result: θ_n

for $i \leftarrow 0$ **to** $n-1$ **do**

  $\beta_t^{1:N}[k_0:k], \zeta_{0:t}[i+1] \leftarrow \widehat{\mathsf{Gd}}(\theta_i, \zeta_{0:t}[i], s_{0:t,\theta_i}, k, k_0)$

  set $\Pi_{(k_0,k),N}(s_{0:t,\theta_i}) = \frac{1}{N(k-k_0)} \sum_{\ell=k_0}^{k-1} \sum_{j=1}^{N} \beta_t^j[\ell]$

  set $\theta_{i+1} \leftarrow \theta_i + \gamma_{i+1}\Pi_{(k_0,k),N}(s_{0:t,\theta_i})$

define

$$\mathbb{P}_{\theta,t} : \mathbf{E}_t^{k-k_0} \times \boldsymbol{\mathcal{E}}_t^{\otimes(k-k_0)} \ni (\boldsymbol{y}_t[k_0:k], z_{0:t}[k_0:k], A) \mapsto \mathbb{K}_{\theta,t}^{k_0} \otimes \mathbb{K}_{\theta,t}^{\otimes(k-k_0)}(z_{0:t}[k], A), \tag{4.1}$$

where $\mathbb{K}_{\theta,t}$ is the PPG kernel defined in (3.21). We write $\mathbb{P}_{\theta,t}$ instead of $\mathbb{P}_\theta$ to explicit the dependence of the kernel on the *fixed* number of observations t. Note that $\mathbb{P}_{\theta,t}$ depends only on the last frozen path, namely $z_{0:t}[k]$. Note also that, since $\mathbb{K}_{\theta,t}$ depends only on the paths, there is no dependence between $\boldsymbol{y}_{t,\ell}[k_0:k]$ and $\boldsymbol{y}_{t,\ell+1}[k_0:k]$. Evaluating the function

$$b_t^{1:N}[k_0:k] \mapsto [N(k-k_0)]^{-1} \sum_{\ell=k_0+1}^{k} \sum_{j=1}^{N} b_t^j[\ell]$$

at a realisation of this kernel gives the roll-out estimator whose properties are analysed in Theorem 19.

The following assumptions, are vital when analysing the convergence of Algorithm 3.

A3. *(i) The function $\theta \mapsto V(\theta)$ is L^V-smooth.*

(ii) The function $\theta \mapsto \eta_{0:t,\theta}$ is L^η-Lipschitz in total variation distance.

(iii) For each path $\zeta_{0:t} \in \mathsf{X}_{0:t}$, the function $\theta \mapsto K_{\theta,t}(\zeta_{0:t}, \mathrm{d}\tilde{\zeta}_{0:t})$ is L_1^P-Lipschitz in total variation distance, where $K_{\theta,t}$ is the path-marginalized Markov transition kernel associated with the PPG algorithm when the model is parameterized by θ, see (3.22).

(iv) For each path $\zeta_{0:t} \in \mathsf{X}_{0:t}$, the function

$$\theta \mapsto \mathbb{P}_{\theta,t}\Pi_{k_0-1,k,N}(s_{0:t,\theta})(\zeta_{0:t}) \tag{4.2}$$

is L_2^P-Lipschitz in total variation distance.

In the case of score ascent we check, in Section C.1, that these assumptions hold if the strong mixing assumption **A** 2 is satisfied uniformly in θ, and with additional assumptions on the model. We are now ready to state a bound on the mean field $h(\theta)$ for Algorithm 3.

Theorem 25. *Assume **A** 2 uniformly in θ and **A**3 and suppose that the stepsizes $\{\gamma_{\ell+1}\}_{\ell\in[\![0,n]\!]}$ satisfy $\gamma_{\ell+1} \le \gamma_\ell$, $\gamma_\ell < a\gamma_{\ell+1}$, $\gamma_\ell - \gamma_{\ell+1} < a'\gamma_\ell^2$ and $\gamma_1 \le 0.5(L^V + C_h)$ for some $a > 0$, $a' > 0$ and all $n \in \mathbb{N}$. Then,*

$$\mathbb{E}\left[\|h(\theta_\varpi)\|^2\right] \le 2\frac{V_{0,n} + C_{0,n} + C_{0,\gamma}\sum_{k=0}^{n}\gamma_{k+1}^2}{\sum_{k=0}^{n}\gamma_{k+1}}, \tag{4.3}$$

where $V_{0,n} = \mathbb{E}\left[V(\theta) - V(\theta_n)\right]$ *and*

$$C_{0,n} \coloneqq \gamma_1 h(\theta_0) C_0 + \sigma_{bias}(\gamma_1 - \gamma_{n+1} + 1)\delta_{k,N,t}^{-1} , \tag{4.4}$$

$$C_{0,\gamma} \coloneqq \sigma_{mse}^2 L^V + \sigma_{mse} C_1 + \sigma_{bias} L^V \delta_{k,N,t}^{-1} + \sigma_{mse}\sigma_{bias}\left(L^V + \frac{C_2}{1-\kappa_{N,t}}\right)\delta_{k,N,t}^{-1} , \tag{4.5}$$

$$C_h \coloneqq (L^V + a' + 1)\sigma_{bias}\delta_{k,N,t}^{-1} + \left(C_1 + \frac{\sigma_{bias} C_2}{(1-\kappa_{N,t})\delta_{k,N,t}}\right)\left[\frac{a+1}{2} + a\sigma_{mse}\right] , \tag{4.6}$$

$$C_1 = L_2^P\left[1 + \kappa_{N,t}^k \delta_{k,N,t}^{-1}\right] + L^V \tag{4.7}$$

$$C_2 = L_1^P \delta_{k,N,t}^{-1} + L^\eta \kappa_{N,t}^k . \tag{4.8}$$

where C_0 *is independent of* $\sigma_{bias}, \sigma_{mse}, N$ *and where* $\delta_{k,N,t} = 1 - \kappa_{N,t}^k$.

Theorem 25 establishes not only the convergence of Algorithm 3, but also illustrates the impact of the bias and the variance of the PPG on the convergence rate.

Remark 26. *Under additional assumptions on the model (cf Section C.1), if we consider* $\gamma_1 \leq 0.5(L^V + C_h)$, $\gamma_\ell = \gamma_1 \ell^{-1/2}$ *for all* $\ell \in [\![1, n]\!]$, *then* $\sum_{k=0}^n \gamma_{k+1}^2 / \sum_{k=0}^n \gamma_{k+1} \sim \log n/\sqrt{n}$, *showing that* $\mathbb{E}\left[\|h(\theta_\varpi)\|^2\right]$ *is* $\mathcal{O}(\log n/\sqrt{n})$, *where the leading constant depends on* σ_{bias} *and* σ_{mse}.

Remark 26 establishes the rate of convergence of Algorithm 3. In principle we could try to optimize the parameters k, k_0 and N of the algorithm using these bounds, but one of the main challenges with this approach is the determination of the mixing rate, which is crudely upper bounded by $\kappa_{N,t}$. Still, our bound provides interesting information of the role of both bias and MSE.

We now proceed to present the proof of Theorem 25. Section 4.1.1 establishes, following closely Karimi et al. (2019), a non-asymptotic bound for stochastic approximation schemes under general assumptions. Section 4.1.2 shows how assumptions **A** 3 and **A** 2 imply the assumptions provided in Section 4.1.1 and therefore allow to establish Theorem 25. Finally, in the appendix, Section C.1 provides sufficient assumptions on the model ensuring that **A**3 holds.

4.1.1 Non-asymptotic bound

We follow closely Karimi et al. (2019). Consider the recursion

$$\theta_{n+1} = \theta_n - \gamma_{n+1} H_{\theta_n}(X_{n+1}), \quad n \in \mathbb{N},$$

where $\theta_n \in \Theta \subset \mathbb{R}^d$ for some $d \in \mathbb{N}^*$ and $\{X_n\}_{n\in\mathbb{N}}$ is a *state-dependent* Markov chain on some measurable space $(\mathsf{X}, \mathcal{X})$ in the sense that $X_{n+1} \sim \mathbb{P}_{\theta_n}(X_n, \cdot)$ with $\mathbb{P}_\theta$ being some Markov kernel on $(\mathsf{X}, \mathcal{X})$. Let $h(\theta) = \int H_\theta(x)\, \pi_\theta(\mathrm{d}x)$, where π_θ is the invariant measure of $\mathbb{P}_\theta$ and $e_{n+1} \coloneqq H_{\theta_n}(X_{n+1}) - h(\theta_n)$. As all norms are equivalent in finite dimensional vector spaces, we use $\|\cdot\|$ to denote a generic norm. We denote by $\{\mathcal{F}_n\}_{n\in\mathbb{N}}$ the natural filtration of the Markov chain $\{X_n\}_{n\in\mathbb{N}}$.

A4. *There exists a Borel measurable function* $V : \Theta \to \mathbb{R}$ *such that for every* $\theta \in \Theta$, $\nabla V(\theta) = h(\theta)$.

A5. *There exists* $L^V \in \mathbb{R}_{\geq 0}$ *such that for every* $(\theta, \theta') \in \Theta^2$,

$$\|\nabla V(\theta) - \nabla V(\theta')\| \leq L^V \|\theta - \theta'\|.$$

A6. *There exists a Borel measurable function* $\widehat{H} : \Theta \times \mathsf{X} \to \Theta$ *such that for every* $\theta \in \Theta$ *and* $x \in \mathsf{X}$,

$$\widehat{H}_\theta(x) - \mathbb{P}_\theta \widehat{H}_\theta(x) = H_\theta(x) - h(\theta) .$$

A7. *There exists* $L^{\mathbb{P}\widehat{H}} \in \mathbb{R}_{\geq 0}$ *such that for every* $(\theta_0, \theta_1) \in \Theta^2$,

$$\sup_{x \in \mathsf{X}} \|\mathbb{P}_{\theta_0} \widehat{H}_{\theta_0}(x) - \mathbb{P}_{\theta_0} \widehat{H}_{\theta_1}(x)\| \leq L^{\mathbb{P}\widehat{H}} \|\theta_0 - \theta_1\| \, .$$

A8. *There exists* $L_0^{\mathbb{P}\widehat{H}} \in \mathbb{R}_{\geq 0}$ *such that*

$$\sup_{\theta \in \Theta} \|\mathbb{P}_\theta \widehat{H}_\theta\| \leq L_0^{\mathbb{P}\widehat{H}} \, .$$

A9. *There exists* $\sigma_{mse} \in \mathbb{R}_{\geq 0}$ *such that for every* $x \in \mathsf{X}$ *and* $\theta \in \Theta$,

$$\int \|H_\theta(x') - h(\theta)\|^2 \, \mathbb{P}_\theta(x, \mathrm{d}x') \leq \sigma_{mse}^2 \, .$$

A10. *There exists* $L^{\widehat{H}} \in \mathbb{R}_{\geq 0}$ *such that for every* $x \in \mathsf{X}$,

$$\sup_{\theta \in \Theta} \int \|\widehat{H}_\theta\| \, \mathbb{P}_\theta(x, \mathrm{d}x') \leq L^{\widehat{H}} \, .$$

Theorem 27. *Assume that* ***A*** *4–****A*** *10 hold. In addition, assume that there exist* $a > 0$ *and* $a' > 0$ *such that for all* $n \in \mathbb{N}$,

$$\gamma_{n+1} \leq \gamma_n \leq a\gamma_{n+1} \, , \quad \gamma_n - \gamma_{n+1} \leq a'\gamma_n^2 \, , \quad \gamma_1 \leq (L^V + C_h)^{-1}/2 \, .$$

Moreover, for any $n \in \mathbb{N}^*$, *let* ϖ *be a* $[\![0, n]\!]$*-valued random variable, independent of* $\{\mathcal{F}_\ell\}_{\ell \geq 0}$ *and such that* $\mathbb{P}(\varpi = k) = \gamma_{k+1} / \sum_{\ell=0}^n \gamma_{\ell+1}$ *for* $k \in [\![0, n]\!]$. *Then,*

$$\mathbb{E}\left[\|h(\theta_\varpi)\|^2\right] \leq 2 \frac{V_{0,n} + C_{0,n} + (\sigma_{mse}^2 L^V + C_\gamma) \sum_{k=0}^n \gamma_{k+1}^2}{\sum_{k=0}^n \gamma_{k+1}} \, ,$$

where $V_{0,n} \coloneqq \mathbb{E}\left[V(\theta) - V(\theta_n)\right]$ *and*

$$C_{0,n} \coloneqq \gamma_1 h(\theta_0) L^{\widehat{H}} + L_0^{\mathbb{P}\widehat{H}} (\gamma_1 - \gamma_{n+1} + 1) \, , \tag{4.9}$$

$$C_\gamma \coloneqq \sigma_{mse} L^{\mathbb{P}\widehat{H}} + (1 + \sigma_{mse}) L^V L_0^{\mathbb{P}\widehat{H}} \, , \tag{4.10}$$

$$C_h \coloneqq L^{\mathbb{P}\widehat{H}} \left((a+1)/2 + a\sigma_{mse}\right) + (L^V + a' + 1) L_0^{\mathbb{P}\widehat{H}} \, . \tag{4.11}$$

Proof. We follow closely the proof of (Karimi et al., 2019, Theorem 2) and adapt it to our setting. First, note that by **A** 4, assumptions **A1** and **A2** of (Karimi et al., 2019, Theorem 2) hold with $c_0 = d_0 = 0$ and $c_1 = d_1 = 1$. In addition, the claim in (Karimi et al., 2019, Lemma 1) holds true since by **A**5, **A3** holds. Moreover, (Karimi et al., 2019, Equation 17) can also be established under **A** 9, as we may rewrite it as

$$\sum_{\ell=0}^n \gamma_{\ell+1}^2 \mathbb{E}\left[\|e_{\ell+1}\|^2\right] = \sum_{\ell=0}^n \gamma_{\ell+1}^2 \mathbb{E}\left[\mathbb{E}\left[\|e_{\ell+1}\|^2 \mid \mathcal{F}_\ell\right]\right] \leq \sigma_{mse}^2 \sum_{\ell=0}^n \gamma_{\ell+1}^2 \, .$$

Following the proof of (Karimi et al., 2019, Lemma 2), consider the decomposition

$$\mathbb{E}\left[-\sum_{\ell=0}^n \gamma_{\ell+1} \left\langle \nabla V(\theta_\ell), e_{\ell+1} \right\rangle\right] = \mathbb{E}\left[A_1 + A_2 + A_3 + A_4 + A_5\right] ,$$

where

$$A_1 \coloneqq -\sum_{\ell=1}^{n} \gamma_{\ell+1} \left\langle \nabla V(\theta_\ell), \widehat{H}_{\theta_\ell}(X_{\ell+1}) - \mathbb{P}_{\theta_\ell} \widehat{H}_{\theta_\ell}(X_\ell) \right\rangle,$$

$$A_2 \coloneqq -\sum_{\ell=1}^{n} \gamma_{\ell+1} \left\langle \nabla V(\theta_\ell), \mathbb{P}_{\theta_\ell} \widehat{H}_{\theta_\ell}(X_\ell) - \mathbb{P}_{\theta_{\ell-1}} \widehat{H}_{\theta_{\ell-1}}(X_\ell) \right\rangle,$$

$$A_3 \coloneqq -\sum_{\ell=1}^{n} \gamma_{\ell+1} \left\langle \nabla V(\theta_\ell) - \nabla V(\theta_{\ell-1}), \mathbb{P}_{\theta_{\ell-1}} \widehat{H}_{\theta_{\ell-1}}(X_\ell) \right\rangle,$$

$$A_4 \coloneqq -\sum_{\ell=1}^{n} (\gamma_{\ell+1} - \gamma_\ell) \left\langle \nabla V(\theta_{\ell-1}), \mathbb{P}_{\theta_{\ell-1}} \widehat{H}_{\theta_{\ell-1}}(X_\ell) \right\rangle,$$

$$A_5 \coloneqq -\gamma_1 \left\langle \nabla V(\theta_0), \widehat{H}_{\theta_0}(X_1) \right\rangle + \gamma_{n+1} \left\langle \nabla V(\theta_n), \mathbb{P}_{\theta_n} \widehat{H}_{\theta_n}(X_{n+1}) \right\rangle.$$

As $\widehat{H}_{\theta_\ell}(X_{\ell+1}) - \mathbb{P}_{\theta_\ell} \widehat{H}_{\theta_\ell}(X_\ell)$ is a martingale difference, it holds that $\mathbb{E}[A_1] = 0$. The upper bounds on the expectations of A_2, A_3 and A_4 are obtained similarly as in Karimi et al. (2019). Using **A** 7,

$$A_2 \leq L^{\mathbb{P}\widehat{H}} \left(\sigma_{mse} \sum_{k=1}^{n} \gamma_k^2 + \frac{1}{2} (1 + 2a\sigma_{mse} + a) \sum_{k=0}^{n} \gamma_{k+1}^2 \|h(\theta_k)\|^2 \right).$$

By **A** 5 and 8,

$$A_3 \leq L^V L_0^{\mathbb{P}\widehat{H}} \left((1 + \sigma_{mse}) \sum_{k=1}^{n} \gamma_k^2 + \sum_{k=1}^{n} \gamma_k^2 \|h(\theta_k)\|^2 \right).$$

On the other hand,

$$A_4 \leq L_0^{\mathbb{P}\widehat{H}} \left(\gamma_1 - \gamma_{n+1} + a' \sum_{k=1}^{n} \gamma_k^2 \|h(\theta_{k-1})\|^2 \right).$$

We now focus on A_5. As in the proof of (Karimi et al., 2019, Lemma 2), the expectation of the first term can be straightforwardly bounded by $\gamma_1 \|h(\theta_0)\| L^{\widehat{H}}$ using the Cauchy–Schwarz inequality and **A** 10. The second term can, using **A** 8 and $\gamma_{n+1}\|h(\theta_n)\| \leq 1 + \gamma_{n+1}^2 \|h(\theta_n)\|^2$, be bounded in the same way according to

$$\begin{aligned} \gamma_{n+1} \left\langle \nabla V(\theta_n), \mathbb{P}_{\theta_n} \widehat{H}_{\theta_n}(X_{n+1}) \right\rangle \leq L_0^{\mathbb{P}\widehat{H}} \gamma_{n+1} \|h(\theta_n)\| &\leq L_0^{\mathbb{P}\widehat{H}} \left(1 + \gamma_{n+1}^2 \|h(\theta_n)\|^2\right) \\ &\leq L_0^{\mathbb{P}\widehat{H}} \left(1 + \sum_{\ell=0}^{n} \gamma_{\ell+1}^2 \|h(\theta_\ell)\|^2\right). \end{aligned}$$

The rest of the proof follows that of (Karimi et al., 2019, Theorem 2). □

4.1.2 Application to Theorem 25

The goal of this section is to establish that the assumptions of Theorem 25 ensure all the assumptions in section 4.1.1, which in turn allows Theorem 27 to be applied.

4.1.2.1 Verification of the assumptions of Theorem 27

The score ascent algorithm (Algorithm 3) can be formulated as follows.

1. Sample $(z_{0:t,\ell}[k_0 : k], \boldsymbol{y}_{t,\ell}[k_0 : k]) \sim \mathbb{P}_{\theta_\ell,t}((z_{0:t,\ell-1}[k_0 : k], \boldsymbol{y}_{t,\ell-1}[k_0 : k]), \cdot)$.

2. Update the parameter according to $\eta_{\ell+1} = \eta_\ell + \gamma_{\ell+1} H(z_{0:t,\ell}[k_0:k], \boldsymbol{y}_{t,\ell}[k_0:k])$, where

$$H(z_{0:t,\ell}[k_0:k], \boldsymbol{y}_{t,\ell}[k_0:k]) = \frac{1}{k-k_0+1} \sum_{i=k_0}^{k} \mu(\boldsymbol{\beta}_{t,\ell}[i])(\mathrm{id}) = \Pi_{(k_0-1,k),N}(h_t),$$

where $\Pi_{(k_0-1,k),N}(h_t)$ is defined in (3.28). We denote by $\pi_{\theta,t}$ the invariant distribution of $\mathbb{P}_{\theta,t}$, which, by Theorem 13, is given by $\pi_{\theta,t} = (\eta_{0:t} \otimes \mathbb{C}_t \mathbb{S}_t)^{\otimes(k-k_0)}$.

We also require the strong mixing assumption to hold uniformly in θ.

A11 (Strong mixing uniformly in θ)**.** *For every $s \in \mathbb{N}$ there exist $\underline{\tau}_s$, $\bar{\tau}_s$, $\underline{\sigma}_s$, and $\bar{\sigma}_s$ in $\mathbb{R}_+^*$ such that for all $\theta \in \Theta$,*

(i) $\underline{\tau}_s \le g_{s,\theta}(x_s) \le \bar{\tau}_s$ *for every* $x_s \in \mathsf{X}_s$,
(ii) $\underline{\sigma}_s \le m_{s,\theta}(x_s, x_{s+1}) \le \bar{\sigma}_s$ *for every* $(x_s, x_{s+1}) \in \mathsf{X}_{s:s+1}$.

Note that the assumption above implies that $\kappa_{N,t}$ is also uniform in θ.

Proof that A 4 holds.

Proposition 28. *For all $\theta \in \Theta$, $h(\theta) = \nabla V(\theta)$, where $V(\theta) = \log \gamma_{0:t,\theta}(\mathsf{X}_{0:t})$ is the log-likelihood function.*

Proof. By Theorem 13,

$$\begin{aligned} h(\theta) &= \int H(\tilde{\boldsymbol{y}}_t[k_0:k], \tilde{x}_{0:t}[k_0:k])\, \pi_{\theta,t}(\mathrm{d}(\tilde{\boldsymbol{y}}_t[k_0:k], \tilde{x}_{0:t}[k_0:k])) \\ &= \frac{1}{k-k_0+1} \sum_{i=k_0}^{k} \int [\eta_{0:t,\theta} \otimes \mathbb{C}_{t,\theta} \mathbb{S}_{t,\theta}]\, (\mathrm{d}(\tilde{\boldsymbol{y}}_t[i], \tilde{x}_{0:t}[i])) \mu(\tilde{\boldsymbol{\beta}}_{t,\ell}[i])(\mathrm{id}) \\ &= \eta_{0:t,\theta}\, (s_{0:t,\theta}) = \nabla V(\theta). \end{aligned}$$

□

Proof that A 5 holds. **A** 5 is trivially implied by **A** 3(i).

Proof that A 6 and 8 hold. Let $\widehat{H}_\theta$ be given by

$$\widehat{H}_\theta : \mathsf{E}_t^{k-k_0} \ni (\boldsymbol{y}_t[k_0:k], z_{0:t}[k_0:k]) \mapsto \sum_{r=0}^{\infty} \{\mathbb{P}_{\theta,t}^r H(\boldsymbol{y}_t[k_0:k], z_{0:t}[k_0:k]) - h(\theta)\}. \tag{4.12}$$

Then the following holds true.

Lemma 29. *Assume A 11. Then for all $\theta \in \Theta$ and $t \in \mathbb{N}^*$,*

$$\|\mathbb{P}_{\theta,t} \widehat{H}_\theta\|_\infty \le \sigma_{bias} (1 - \kappa_{N,t}^k)^{-1}\ .$$

Proof. By Theorem 19, we have for any $r > 0$

$$\left|\mathbb{P}_{\theta,t}^r H(\boldsymbol{y}_t[k_0:k], z_{0:t}[k_0:k]) - h(\theta)\right| \le \sigma_{bias} \kappa_{N,t}^{(r-1)k}$$

and thus

$$\|\mathbb{P}_{\theta,t} \widehat{H}_\theta\|_\infty \le \sum_{r=1}^{\infty} \left\|\mathbb{P}_{\theta,t}^r H - h(\theta)\right\|_\infty \le \sigma_{bias} \sum_{r=0}^{\infty} \kappa_{N,t}^{rk} \le \sigma_{bias} (1 - \kappa_{N,t}^k)^{-1}\ ,$$

where $\kappa_{N,t} \in (0,1)$. □

Lemma 29 proves **A** 6 and 8 with $L_0^{\mathbb{P}\widehat{H}} := \sigma_{bias}(1 - \kappa_{N,t}^k)^{-1}$.

Proof that A 7 holds.

Theorem 30. *Assume A 11 and A 3. Then for every $t \in \mathbb{N}$, $\theta \in \Theta$ and $N \in \mathbb{N}^*$ such that $N > 1+5\rho_t^2 t/2$,*

$$\left\| \mathbb{P}_{\theta_1,t} \widehat{H}_{\theta_1} - \mathbb{P}_{\theta_2,t} \widehat{H}_{\theta_2} \right\|_\infty \leq L^{\mathbb{P}\widehat{H}} \|\theta_1 - \theta_2\| \ ,$$

where

$$L^{\mathbb{P}\widehat{H}} := \|L_2^P\|_\infty \left[1 + \kappa_{N,t}^k (1 - \kappa_{N,t}^k)\right] + L^V + \sigma_{bias} (1-\kappa_{N,t})^{-1} (1-\kappa_{N,t}^k)^{-1} \left[\|L_1^P\|_\infty (1-\kappa_{N,t}^k)^{-1} + L^\eta \kappa_{N,t}^k\right] \ . \quad (4.13)$$

Proof. We establish the claim by adapting the proof of (Karimi et al., 2019, Lemma 7). First, recall that the kernel $K_{\theta,t}$ defined in (3.22) is the path marginalized version of $\mathbb{K}_{\theta,t}$ given in (3.21). Note that for every $x \in \mathsf{E}_t^{k-k_0}$,

$$\mathbb{P}_{\theta_1,t} \widehat{H}_{\theta_1}(x) = \sum_{n=0}^\infty \delta_x \mathbb{P}_{\theta_1,t} \left\{ \mathbb{P}_{\theta_1,t}^n H - h(\theta_1) \right\} = \sum_{n=0}^\infty \delta_x K_{\theta_1,t}^{kn} \left\{ \mathbb{P}_{\theta_1,t} H - \eta_{0:t,\theta_1} \mathbb{P}_{\theta_1,t} H \right\} \ ,$$

where we have used (i) the fact that the backward statistics output by $\mathbb{P}_{\theta,t}$ are independent of the input backward statistics and (ii) the penultimate line in the computation of $h(\theta)$ above. We follow the proof of (Fort et al., 2011, Lemma 4.2) and consider the following decomposition: for $n \in \mathbb{N}^*$,

$$\begin{aligned} &\delta_x K_{\theta_1,t}^{kn} \left(\mathbb{P}_{\theta_1,t} H - \eta_{0:t,\theta_1} \mathbb{P}_{\theta_1,t} H \right) - \delta_x K_{\theta_2,t}^{kn} \left(\mathbb{P}_{\theta_2,t} H - \eta_{0:t,\theta_2} \mathbb{P}_{\theta_2,t} H \right) \quad (4.14) \\ &\quad = \sum_{j=0}^{n-1} \left(\delta_x K_{\theta_1,t}^{kj} - \eta_{0:t,\theta_1} \right) \left(K_{\theta_1,t}^{kj} - K_{\theta_2,t}^{kj} \right) \left(K_{\theta_2,t}^{k(n-j-1)} \mathbb{P}_{\theta_1,t} H - \eta_{0:t,\theta_2} \mathbb{P}_{\theta_1,t} H \right) \\ &\quad - \left(\delta_x K_{\theta_2,t}^{kn} \mathbb{P}_{\theta_2,t} H - \eta_{0:t,\theta_2} \mathbb{P}_{\theta_2,t} H \right) + \left(\delta_x K_{\theta_2,t}^{kn} \mathbb{P}_{\theta_1,t} H - \eta_{0:t,\theta_2} \mathbb{P}_{\theta_1,t} H \right) \\ &\quad - \eta_{0:t,\theta_1} \left(K_{\theta_2,t}^{kn} \mathbb{P}_{\theta_1,t} H - \eta_{0:t,\theta_2} \mathbb{P}_{\theta_1,t} H \right) \ . \end{aligned}$$

Applying Theorem 14 with $\mu = \delta_x$ and $\nu = \eta_{0:t,\theta}$ and using the fact that $\eta_{0:t,\theta} K_{\theta,t}^\ell = \eta_{0:t,\theta}$ for all $\ell \in \mathbb{N}$, we obtain that for all $\ell \in \mathbb{N}$ and all $\theta \in \Theta$, $\left\| \delta_x K_{\theta,t}^\ell - \eta_{0:t,\theta} \right\|_{\mathrm{TV}} \leq \kappa_{N,t}^\ell$. Note that by **A** 3(iii), $K_{\theta,t}$ is Lipschitz; therefore, for all $r \in \mathbb{N}^*$, by Lemma 62, $K_{\theta,t}^r$ is Lipschitz with constant $\|L_1^P\|_\infty (1-\kappa_{N,t})^{-1}$. Combining all this together, we obtain

$$\begin{aligned} &\left| \left(\delta_x K_{\theta_1,t}^{kj} - \eta_{0:t,\theta_1} \right) \left(K_{\theta_1,t}^{kj} - K_{\theta_2,t}^{kj} \right) \left(K_{\theta_2,t}^{k(n-j-1)} \mathbb{P}_{\theta_1,t} H - \eta_{0:t,\theta_2} \mathbb{P}_{\theta_1,t} H \right) \right| \\ &\quad = \left| \left(\delta_x K_{\theta_1,t}^{kj} - \eta_{0:t,\theta_1} \right) \left(K_{\theta_1,t}^{kj} - K_{\theta_2,t}^{kj} \right) \left\{ K_{\theta_2,t}^{k(n-j-1)} \left[\mathbb{P}_{\theta_1,t} H - h(\theta_1) \right] - \eta_{0:t,\theta_2} \left[\mathbb{P}_{\theta_1,t} H - h(\theta_1) \right] \right\} \right| \\ &\quad \leq \|L_1^P\|_\infty (1-\kappa_{N,t})^{-1} \kappa_{N,t}^{kj} \kappa_{N,t}^{k(n-j-1)} \|\mathbb{P}_{\theta_1,t} H - h(\theta_1)\|_\infty \|\theta_1 - \theta_2\| \\ &\quad \leq \sigma_{bias} \|L_1^P\|_\infty (1-\kappa_{N,t})^{-1} \kappa_{N,t}^{k(n-1)} \|\theta_1 - \theta_2\| \ , \end{aligned}$$

where the last inequality is due to Theorem 19. Therefore, the first term of the right side of (4.14) is upper bounded by $\sigma_{bias} \|L_1^P\|_\infty (1-\kappa_{N,t})^{-1} n \kappa_{N,t}^{k(n-1)} \|\theta_1 - \theta_2\|$. The second term of (4.14) can be written

$$\begin{aligned} &- \left(\delta_x K_{\theta_2,t}^{kn} \mathbb{P}_{\theta_2,t} H - \eta_{0:t,\theta_2} \mathbb{P}_{\theta_2,t} H \right) + \left(\delta_x K_{\theta_2,t}^{kn} \mathbb{P}_{\theta_1,t} H - \eta_{0:t,\theta_2} \mathbb{P}_{\theta_1,t} H \right) \\ &\qquad = \left(\delta_x K_{\theta_2,t}^{kn} - \eta_{0:t,\theta_2} \right) \left(\mathbb{P}_{\theta_1,t} H - \mathbb{P}_{\theta_2,t} H \right) \ , \end{aligned}$$

and using again the ergodicity of $K_{\theta,t}$ and the fact that $\theta \mapsto \mathbb{P}_{\theta,t}H$ is uniformly Lipschitz by **A** 3(iv), we may conclude that it is upper bounded by $\|L_2^P\|_\infty \kappa_{N,t}^{kn} \|\theta_1 - \theta_2\|$. Finally, for the last term, using the facts that $K_{\theta,t}^k$ is $\eta_{0:t,\theta}$-invariant and geometrically ergodic and that $\theta \mapsto \eta_{0:t,\theta}$ is Lipschitz by **A** 3(iv) yields

$$\begin{aligned}
&\left|\eta_{0:t,\theta_1}\left(K_{\theta_2,t}^{kn}\mathbb{P}_{\theta_1,t}H - \eta_{0:t,\theta_2}\mathbb{P}_{\theta_1,t}H\right)\right| \\
&\quad = \left|\left(\eta_{0:t,\theta_1} - \eta_{0:t,\theta_2}\right)\left\{K_{\theta_2,t}^{kn}\left[\mathbb{P}_{\theta_1,t}H - h(\theta_1)\right] - \eta_{0:t,\theta_2}\left[\mathbb{P}_{\theta_1,t}H - h(\theta_1)\right]\right\}\right| \\
&\quad \leq L^\eta \kappa_{N,t}^{kn} \|\mathbb{P}_{\theta_1,t}H - h(\theta_1)\|_\infty \|\theta_1 - \theta_2\| \\
&\quad \leq L^\eta \sigma_{bias}(1-\kappa_{N,t})^{-1}\kappa_{N,t}^{kn}\|\theta_1 - \theta_2\| \, .
\end{aligned}$$

Therefore, we have that

$$\begin{aligned}
&\delta_x K_{\theta_1,t}^{kn}\left(\mathbb{P}_{\theta_1,t}H - \eta_{0:t,\theta_1}\mathbb{P}_{\theta_1,t}H\right) - \delta_x K_{\theta_2,t}^{kn}\left(\mathbb{P}_{\theta_2,t}H - \eta_{0:t,\theta_2}\mathbb{P}_{\theta_2,t}H\right) \\
&\quad \leq \left\{\sigma_{bias}\|L_1^P\|_\infty(1-\kappa_{N,t})^{-1} n \kappa_{N,t}^{k(n-1)} + \left[\|L_2^P\|_\infty + L^\eta \sigma_{bias}(1-\kappa_{N,t})^{-1}\right]\kappa_{N,t}^{kn}\right\}\|\theta_1 - \theta_2\| \, .
\end{aligned}$$

Therefore, we obtain

$$\begin{aligned}
&\left|\mathbb{P}_{\theta_1,t}\widehat{H}_{\theta_1}(x) - \mathbb{P}_{\theta_2,t}\widehat{H}_{\theta_2}(x)\right| \\
&\leq \left|\delta_x\mathbb{P}_{\theta_1,t}H - \delta_x\mathbb{P}_{\theta_2,t}H\right| + \left|\eta_{0:t,\theta_1}\mathbb{P}_{\theta_1,t}H - \eta_{0:t,\theta_2}\mathbb{P}_{\theta_2,t}H\right| \\
&\qquad + \left|\sum_{n=1}^{\infty}\delta_x K_{\theta_1,t}^{kn}\left(\mathbb{P}_{\theta_1,t}H - \eta_{0:t,\theta_1}\mathbb{P}_{\theta_1,t}H\right) - \delta_x K_{\theta_2,t}^{kn}\left(\mathbb{P}_{\theta_2,t}H - \eta_{0:t,\theta_2}\mathbb{P}_{\theta_2,t}H\right)\right| \\
&\leq \left|\delta_x\mathbb{P}_{\theta_1,t}H - \delta_x\mathbb{P}_{\theta_2,t}H\right| + \left|\eta_{0:t,\theta_1}\mathbb{P}_{\theta_1,t}H - \eta_{0:t,\theta_2}\mathbb{P}_{\theta_2,t}H\right| \\
&+ \left\{\sigma_{bias}\|L_1^P\|_\infty(1-\kappa_{N,t})^{-1}(1-\kappa_{N,t}^k)^{-2}\right. \\
&+ \left.\left[\|L_2^P\|_\infty + L^\eta\sigma_{bias}(1-\kappa_{N,t})^{-1}\right]\kappa_{N,t}^k(1-\kappa_{N,t}^k)^{-1}\right\}\|\theta_1 - \theta_2\| \, .
\end{aligned}$$

To conclude, note that by **A** 3(iv), $\|\delta_x\mathbb{P}_{\theta_1,t}H - \delta_x\mathbb{P}_{\theta_2,t}H\| \leq \|L_2^P\|_\infty\|\theta_1 - \theta_2\|$. Furthermore, note that by Theorem 13 we obtain that for all $\theta \in \Theta$, $\eta_{0:t,\theta}\mathbb{P}_{\theta,t}H = \eta_{0:t,\theta}s_{0:t,\theta} = \nabla V(\theta)$. Therefore, by **A** 3(i) we obtain that $\|\eta_{0:t,\theta_1}\mathbb{P}_{\theta_1,t}H - \eta_{0:t,\theta_2}\mathbb{P}_{\theta_2,t}H\| \leq L^V\|\theta_1 - \theta_2\|$, concluding the proof. □

Proof that A 9 holds. **A** 9 is simply a bound on the MSE of the roll-out PPG estimator, given by Theorem 19.

Proof that A 10 holds.

Proposition 31. *For all $\theta \in \Theta$ and all $\ell \in [\![1, t-1]\!]$*

$$\mathbb{E}\left[\|\widehat{H}_\theta\| \mid \mathcal{F}_\ell\right] \leq 2\|s_{0:t,\theta}\|_\infty + \sigma_{bias}(1-\kappa_{N,t}^k)^{-1} \, .$$

Proof. Note that for all $x \in \mathsf{E}_t^{k-k_0}$ and all $\theta \in \Theta$,

$$\widehat{H}_\theta(x) = H(x) - h(\theta) + \mathbb{P}_{\theta,t}\widehat{H}_\theta(x) \, . \tag{4.15}$$

Lemma 29 shows that $\|\mathbb{P}_{\theta,t}\widehat{H}_\theta\|_\infty \leq \sigma_{bias}(1-\kappa_{N,t}^k)^{-1}$. Note that $h(\theta) \leq \|s_{0:t,\theta}\|_\infty$ We write

$$\mathbb{E}\left[\|H\| \mid \mathcal{F}_\ell\right] \leq \frac{1}{(k-k_0+1)N}\sum_{i=k_0}^{k}\sum_{j=1}^{N}\mathbb{E}\left[\|\beta_{t,\ell}^j[i]\| \mid \mathcal{F}_\ell\right] \, .$$

By Proposition 61, $\mathbb{E}\left[\|\beta_{t,\ell}^j[i]\| \mid \mathcal{F}_\ell\right] \leq \|s_{0:t,\theta}\|_\infty$, concluding the proof. □

A10 follows directly by Proposition 31 and by considering $\sup_{\theta\in\Theta}\|s_{0:t,\theta}\|_\infty$.

4.1.2.2 Proof of Theorem 25

We have shown in Section 4.1.2.1 that under **A**3 and 11, it is possible to apply Theorem 27. To conclude the proof of Theorem 25 we just have to rearrange the constants. We start by rewriting the constant in Theorem 30

$$L^{\mathbb{P}^{\widehat{H}}} = C_1 + \sigma_{bias}(1-\kappa_{N,t})^{-1}(1-\kappa_{N,t}^k)^{-1}C_2,$$

with

$$\begin{aligned} C_1 &= \left\|L_2^P\right\|_\infty \left[1+\kappa_{N,t}^k(1-\kappa_{N,t}^k)^{-1}\right] + L^V \\ C_2 &= \left\|L_1^P\right\|_\infty (1-\kappa_{N,t}^k)^{-1} + L^\eta \kappa_{N,t}^k \ . \end{aligned}$$

By (4.10) and Lemma 29,

$$\begin{aligned} C_\gamma &= \sigma_{mse} L^{\mathbb{P}^{\widehat{H}}} + (1+\sigma_{mse}) L^V L_0^{\mathbb{P}^{\widehat{H}}} \\ &= \sigma_{mse}\left[C_1 + \sigma_{bias}(1-\kappa_{N,t})^{-1}(1-\kappa_{N,t}^k)^{-1}C_2\right] + (1+\sigma_{mse})L^V\sigma_{bias}(1-\kappa_{N,t}^k)^{-1} \\ &= \sigma_{mse}C_1 + \sigma_{mse}\sigma_{bias}(1-\kappa_{N,t}^k)^{-1}\left[L^V + (1-\kappa_{N,t})^{-1}C_2\right] + \sigma_{bias}L^V(1-\kappa_{N,t}^k)^{-1} \ . \end{aligned}$$

Therefore,

$$\begin{aligned} C_{0,\gamma} &\coloneqq \sigma_{mse}^2 L^V + C_\gamma \\ &= \sigma_{mse}^2 L^V + \sigma_{mse}C_1 + \sigma_{mse}\sigma_{bias}(1-\kappa_{N,t}^k)^{-1}\left[L^V + (1-\kappa_{N,t})^{-1}C_2\right] + \sigma_{bias}L^V(1-\kappa_{N,t}^k)^{-1} \ . \end{aligned}$$

In the same way, we can rewrite (4.11) as

$$\begin{aligned} C_h &= L^{\mathbb{P}^{\widehat{H}}}\left[(a+1)/2 + a\sigma_{mse}\right] + (L^V + a' + 1)L_0^{\mathbb{P}^{\widehat{H}}} \\ &= \left[C_1 + \sigma_{bias}(1-\kappa_{N,t})^{-1}(1-\kappa_{N,t}^k)^{-1}C_2\right]\left[(a+1)/2 + a\sigma_{mse}\right] + (L^V + a' + 1)\sigma_{bias}(1-\kappa_{N,t}^k)^{-1} \ . \end{aligned}$$

The constant C_0 from Theorem 25 is $L^{\widehat{H}} = 2\sup_{\theta\in\Theta}\|s_{0:t,\theta}\|_\infty + \sigma_{bias}(1-\kappa_{N,t}^k)^{-1}$ which completes the proof.

4.2 Numerics

In this section, we focus on the numerical analysis of the efficiency of using PPG for learning in the framework developed in Section 4.1. We will restrict ourselves to the case of parameter learning via score ascent. The code used in this section is available [1]. Throughout this section, we set $M = 2$ for the PPG algorithm. In this setting, the competing method that corresponds most closely to the one presented here consists of using, as presented in Algorithm 4, a standard particle Gibbs sampler Π_θ instead of the PPG. One of the most common such samplers is the *particle Gibbs with ancestor sampling* (PGAS) presented in Lindsten et al. (2014a). In Lindholm and Lindsten (2018), the PGAS is used for parameter learning in HMMs via the Expectation Maximization (EM) algorithm.

LGSSM. We consider the LGSSM with state and observation spaces being $\mathbb{R}^5$. We assume that the parameters R and Q are known and consider the inference of $\theta = (A, B)$ on the basis of a simulated sequence of $n = 999$ observations. In this setting, the M-step of the EM algorithm can be solved exactly with the disturbance smoother (Cappé et al., 2005a, Chapter 11). The parameter obtained by this procedure (denoted θ_{mle}) is the reference value for any likelihood maximization algorithm. Table 4.1

[1] https://anonymous.4open.science/r/ppg/

Algorithm 4 Score ascent with particle Gibbs kernel.

Data: $\zeta_{0:t}[0]$, θ_0, number k of paths per trajectory, burn-in k_0, number n of SA iterations, learning-rate sequence $\{\gamma_\ell\}_{\ell\in\mathbb{N}}$, $\Pi_\theta(\zeta_{0:t}, \mathrm{d}\tilde{\zeta}_{0:t})$ a Markov kernel targeting $\eta_{0:t}$.

Result: θ_n

for $i \leftarrow 0$ **to** $n-1$ **do**
  for $j \leftarrow 0$ **to** $k-1$ **do**
    sample $\tilde{\zeta}_{0:t}[j+1] \sim \Pi_\theta(\tilde{\zeta}_{0:t}[j], \cdot)$
  set $\theta_{i+1} \leftarrow \theta_i + \frac{\gamma_{i+1}}{k-k_0} \sum_{\ell=k_0+1}^{k} s_{0:t,\theta_i}(\tilde{\zeta}_{0:t}[\ell])$
  set $\zeta_{0:t}[i+1] = \tilde{\zeta}_{0:t}[k]$

Algorithm	N	k_0	k	D_{mle}	δt(s)
PGAS	64	24	48	0.72 ± 0.04	5.66
PGAS	128	12	24	0.59 ± 0.04	2.84
PGAS	256	6	12	0.59 ± 0.05	1.42
PPG	64	16	32	0.37 ± 0.03	4.56
PPG	128	8	16	0.36 ± 0.04	2.37
PPG	256	4	8	0.35 ± 0.04	1.57

Table 4.1: Distance to θ_{MLE} (D_{mle}) for each configuration in the LGSSM case. δt(s) represents the average running time for each configuration.

shows the L_2 distance between the singular values of θ_{mle} and those of the parameters obtained by Algorithm 3 and Algorithm 4. The CLT confidence intervals were obtained on the basis of 25 replicates. The configurations of the PPG estimators respect a given particle budget $kN = C = 1024$. For a fair comparison, for each configuration of the PPG estimator, we run an equivalent w.r.t. clock time PGAS estimator. The time needed for one gradient step for each estimator averaged over 100 replicates is reported in Table 4.1. The choice of keeping $k_0 = k/2$ is a heuristic rule to achieve a good bias–variance trade-off, but other combinations of k_0 and k may lead to better performance for different problems. We analyse the impact of the different settings for the LGSMM in Section C.3.All settings are the same for both algorithms and are described in Section C.3. The PPG achieves consistently a smaller distance to θ_{mle}. Figure 4.1 displays, for each estimator and configuration, the evolution of the distance to the MLE estimator as a function of the iteration index.

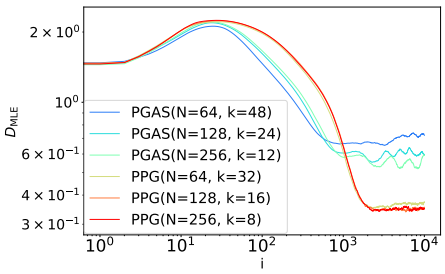


Figure 4.1: Distance to the MLE estimator as a function of the iteration step for the PGAS and PPG configurations from table 4.1. The solid lines and the shaded region represent the mean and CLT confidence intervals obtained with 25 replications.

CRNN. We consider now the problem of inference in a non-linear HMM and in particular the chaotic recurrent neural network introduced by Zhao et al. (2021). We use the same setting as in the original

Algorithm	N	k_0	k	NLL	δt(s)
PGAS	32	32	64	31887 ± 128	3.90
PGAS	64	16	32	31269 ± 254	1.99
PGAS	128	8	16	30994 ± 288	1.16
PPG	32	16	32	22292 ± 48	2.79
PPG	64	8	16	22315 ± 25	1.39
PPG	128	4	8	22353 ± 39	0.92

Table 4.2: Per configuration negative loglikelihood for the CRNN model.

paper. The state and observation equations are

$$X_{m+1} = X_m + \tau^{-1}\Delta\left(-X_m + \gamma W \tanh(X_m)\right) + \epsilon_{m+1},$$
$$Y_m = BX_m + \zeta_m, \quad m \in \mathbb{N},$$

where $\{\epsilon_m\}_{m\in\mathbb{N}^*}$ is a sequence of 20-dimensional independent multivariate Gaussian random variables with zero mean and covariance $0.01\mathbf{I}$ and $\{\zeta_m\}_{m\in\mathbb{N}}$ is a sequence of independent random variables where each component is distributed independently according to a Student's t-distribution with scale 0.1 and 2 degrees of freedom. We consider $\theta = (W, B)$.

In this case, the natural metric used to evaluate the different estimators is the negative log likelihood (NLL). We use the unbiased estimator of the likelihood given by the mean of the log weights produced by a particle filter (Douc et al., 2014, Section 12.1) using $N = 10^4$ particles. Table 4.2 shows the results obtained for 25 different replications for several different configurations of PPG while keeping total budget of particles fixed. As for the LGSSM, for each configuration of the PPG we run the time-equivalent PGAS estimator. Further numerical details and the system configuration used in the experiments are given in Section C.3. We observe that PPG achieves the a considerably lower NLL than PGAS in all configurations.

4.3 Conclusion

We propose a way of using PPG in a learning framework and derive a non-asymptotic bound over the gradient of the updates when doing score ascent with the PPG with explicit dependence on the bias and MSE of the estimator. We provide numerical simulations to support our claims, and we show that our algorithm outperforms the current competitors in the two different examples analysed.

Chapter 5

MCG-DIFF: Monte Carlo guided Diffusion for Bayesian linear inverse problems

5.1 Introduction

This paper is concerned with linear inverse problems $y = \mathrm{A}x + \sigma_y \varepsilon$, where $y \in \mathbb{R}^{\mathsf{d}}_y$ is a vector of indirect observations, $x \in \mathbb{R}^{\mathsf{d}_x}$ is the vector of unknowns, $\mathrm{A} \in \mathbb{R}^{\mathsf{d}_y \times \mathsf{d}_x}$ is the linear forward operator and $\varepsilon \in \mathbb{R}^{\mathsf{d}_y}$ is an unknown noise vector. This general model is used throughout computational imaging, including various tomographic imaging applications such as common types of magnetic resonance imaging Vlaardingerbroek and Boer (2013), X-ray computed tomography Elbakri and Fessler (2002), radar imaging Cheney and Borden (2009), and basic image restoration tasks such as deblurring, super-resolution, and image inpainting González et al. (2009). The classical approach to solving linear inverse problems relies on prior knowledge about x, such as its smoothness, sparseness in a dictionary, or its geometric properties. These approaches attempt to estimate a $\widehat{x}$ by minimizing a regularized inverse problem, $\hat{x} = \mathrm{argmin}_x\{\|y - \mathrm{A}x\|^2 + \mathrm{Reg}(x)\}$, where Reg is a regularization term that balances data fidelity and noise while enabling efficient computations. However, a common difficulty in the regularized inverse problem is the selection of an appropriate regularizer, which has a decisive influence on the quality of the reconstruction.

Whereas regularized inverse problems continue to dominate the field, many alternative statistical formulations have been proposed; see Besag et al. (1991); Idier (2013); Marnissi et al. (2017) and the references therein - see Stuart (2010) for a mathematical perspective. A main advantage of statistical approaches is that they allow for **uncertainty quantification** in the reconstructed solution; see Dashti and Stuart (2017). The **Bayes' formulation** of the regularized inverse problem is based on considering the indirect measurement Y, the state X and the noise ε as random variables, and to specify $p(y|x)$ the *likelihood* (the conditional distribution of Y at X) and the prior $p(x)$ (the distribution of the state). One can use Bayes' theorem to obtain the **posterior distribution** $p(x|y) \propto p(y|x)p(x)$, where "$\propto$" means that the two sides are equal to each other up to a multiplicative constant that does not depend on x. Moreover, the use of an appropriate method for Bayesian inference allows the quantification of the uncertainty in the reconstructed solution x. A variety of priors are available, including but not limited to Laplace Figueiredo et al. (2007), total variation (TV) Kaipio et al. (2000) and mixture-of-Gaussians Fergus et al. (2006). In the last decade, a variety of techniques have been proposed to design and train generative models capable of producing perceptually realistic samples from the original data, even in challenging high-dimensional data such as images or audio Kingma et al. (2019); Kobyzev et al. (2020); Gui et al.

(2021). Denoising diffusion models have been shown to be particularly effective generative models in this context Sohl-Dickstein et al. (2015); Song et al. (2021c,a,b); Benton et al. (2022). These models convert noise into the original data domain through a series of denoising steps. A popular approach is to use a generic diffusion model that has been pre-trained, eliminating the need for re-training and making the process more efficient and versatile Trippe et al. (2023); Zhang et al. (2023). Although this was not the main motivation for developing these models, they can of course be used as prior distributions in Bayesian inverse problems. This simple observation has led to a new, fast-growing line of research on how linear inverse problems can benefit from the flexibility and expressive power of the recently introduced deep generative models; see Arjomand Bigdeli et al. (2017); Wei et al. (2022); Su et al. (2022); Kaltenbach et al. (2023); Shin and Choi (2023); Zhihang et al. (2023); Sahlström and Tarvainen (2023).

Contributions

• We propose `MCGdiff`, a novel algorithm for sampling from the Bayesian posterior of Gaussian linear inverse problems with denoising diffusion model priors. `MCGdiff` specifically exploits the structure of both the linear inverse problem and the denoising diffusion generative model to design an efficient SMC sampler.
• We establish under sensible assumptions that the empirical distribution of the samples produced by `MCGdiff` converges to the target posterior when the number of particles goes to infinity. To the best of our knowledge, `MCGdiff` is the first provably consistent algorithm for conditional sampling from the denoising diffusion posteriors.
• To evaluate the performance of `MCGdiff`, we perform numerical simulations on several examples for which the target posterior distribution is known. Simulation results support our theoretical results, i.e. the empirical distribution of samples from `MCGdiff` converges to the target posterior distribution. This is **not** the case for the competing methods (using the same denoising diffusion generative priors) which are shown, when run with random initialization of the denoising diffusion, to generate a significant number of samples outside the support of the target posterior. We also illustrate samples from `MCGdiff` in imaging inverse problems.

Background and notations. This section provides a concise overview of the diffusion model framework and notations used in this paper. We cover the elements that are important for understanding our approach, and we recommend that readers refer to the original papers for complete details and derivations Sohl-Dickstein et al. (2015); Ho et al. (2020); Song et al. (2021c,a). A denoising diffusion model is a generative model consisting of a forward and a backward process. The forward process involves sampling $X_0 \sim \mathsf{q}_{\mathrm{data}}$ from the data distribution, which is then converted to a sequence $X_{1:n}$ of recursively corrupted versions of X_0. The backward process involves sampling X_n according to an easy-to-sample reference distribution on $\mathbb{R}^{\mathsf{d}_x}$ and generating $X_0 \in \mathbb{R}^{\mathsf{d}_x}$ by a sequence of denoising steps. Following Sohl-Dickstein et al. (2015); Song et al. (2021a), the forward process can be chosen as a Markov chain with joint distribution

$$\mathsf{q}_{0:n}(x_{0:n}) = \mathsf{q}_{\mathrm{data}}(x_0) \prod_{t=1}^{n} q_t(x_t|x_{t-1}), \quad q_t(x_t|x_{t-1}) = \mathcal{N}(x_t; (1-\beta_t)^{1/2} x_{t-1}, \beta_t \, \mathrm{I}_{\mathsf{d}_x}), \tag{5.1}$$

where $\mathrm{I}_{\mathsf{d}_x}$ is the identity matrix of size d_x, $\{\beta_t\}_{t\in\mathbb{N}} \subset (0,1)$ is a non-increasing sequence and $\mathcal{N}(\mathbf{x}; \mu, \Sigma)$ is the p.d.f. of the Gaussian distribution with mean μ and covariance matrix Σ (assumed to be non-singular) evaluated at $\mathbf{x}$. For all $t > 0$, set $\bar{\alpha}_t = \prod_{\ell=1}^{t}(1-\beta_\ell)$ with the convention $\alpha_0 = 1$. We have for all $0 \le s < t \le n$,

$$q_{t|s}(x_t|x_s) := \int \prod_{\ell=s+1}^{t} q_\ell(x_\ell|x_{\ell-1}) \mathrm{d}x_{s+1:t-1} = \mathcal{N}(x_t; (\bar{\alpha}_t/\bar{\alpha}_s)^{1/2} x_s, (1-\bar{\alpha}_t/\bar{\alpha}_s)\, \mathrm{I}_{\mathsf{d}_x}). \tag{5.2}$$

For the standard choices of $\bar{\alpha}_t$, the sequence of distributions $(\mathsf{q}_t)_{t\in\mathbb{N}}$ converges weakly to the standard normal distribution as $t \to \infty$, which we chose as the reference distribution. For the reverse process, Song

et al. (2021a,b) introduce an *inference distribution* $q^{\sigma}_{1:n|0}(x_{1:n}|x_0)$, depending on a sequence $\{\sigma_t\}_{t\in\mathbb{N}}$ of hyperparameters satisfying $\sigma_t^2 \in [0, 1-\bar{\alpha}_{t-1}]$ for all $t \in \mathbb{N}^*$, and defined as

$$q^{\sigma}_{1:n|0}(x_{1:n}|x_0) = q^{\sigma}_{n|0}(x_n|x_0) \prod_{t=n}^{2} q^{\sigma}_{t-1|t,0}(x_{t-1}|x_t, x_0)\,,$$

where

$$q^{\sigma}_{n|0}(x_n|x_0) = \mathcal{N}\left(x_n; \bar{\alpha}_n^{1/2} x_0, (1-\bar{\alpha}_n)\,\mathrm{I}_{\mathsf{d}_x}\right)$$

and

$$q^{\sigma}_{t-1|t,0}(x_{t-1}|x_t, x_0) = \mathcal{N}\left(x_{t-1}; \boldsymbol{\mu}_t(x_0, x_t), \sigma_t^2\,\mathrm{I}_{\mathsf{d}_x}\right)\,,$$

with $\boldsymbol{\mu}_t(x_0, x_t) = \bar{\alpha}_{t-1}^{1/2} x_0 + (1-\bar{\alpha}_{t-1} - \sigma_t^2)^{1/2}(x_t - \bar{\alpha}_t^{1/2} x_0)/(1-\bar{\alpha}_t)^{1/2}$. For $t \in [1:n-1]$, we define by backward induction the sequence $q^{\sigma}_{t|0}(x_t|x_0) = \int q^{\sigma}_{t|t+1,0}(x_t|x_{t+1}, x_0) q^{\sigma}_{t+1|0}(x_{t+1}|x_0)\mathrm{d}x_{t+1}$. It is shown in (Song et al., 2021a, Lemma 1) that for all $t \in [1:n]$, the distributions of the forward and inference process conditioned on the initial state coincide, i.e. that $q^{\sigma}_{t|0}(x_t|x_0) = q_{t|0}(x_t|x_0)$. The backward process is derived from the inference distribution by replacing, for each $t \in [2:n]$, x_0 in the definition $q^{\sigma}_{t-1|t,0}(x_{t-1}|x_t, x_0)$ with a prediction where $\boldsymbol{\chi}^{\theta}_{0|t}(x_t) := \bar{\alpha}_t^{-1/2}\left(x_t - (1-\bar{\alpha}_t)^{1/2}\mathbf{e}^{\theta}(x_t, t)\right)$ where $\mathbf{e}^{\theta}(x, t)$ is typically a neural network parameterized by θ. More formally, the backward distribution is defined as $\mathsf{p}^{\theta}_{0:n}(x_{0:n}) = \mathsf{p}_n(x_n)\prod_{t=0}^{n-1} p^{\theta}_t(x_t|x_{t+1})$, where $\mathsf{p}_n(x_n) = \mathcal{N}(x_n; 0_{\mathsf{d}_x}, \mathrm{I}_{\mathsf{d}_x})$ and for all $t \in [1:n-1]$,

$$p^{\theta}_t(x_t|x_{t+1}) := q^{\sigma}_{t|t+1,0}(x_t|x_{t+1}, \boldsymbol{\chi}^{\theta}_{0|t+1}(x_{t+1})) = \mathcal{N}(x_t, \mathsf{m}^{\theta}_{t+1}(x_{t+1}), \sigma^2_{t+1}\mathrm{I}_{\mathsf{d}_x})\,, \tag{5.3}$$

where $\mathsf{m}_{t+1}(x_{t+1}) := \boldsymbol{\mu}(\boldsymbol{\chi}^{\theta}_{0|t+1}(x_{t+1}), x_{t+1})$ and 0_{d_x} is the null vector of size d_x. At step 0, we set $p_0(x_0|x_1) := \mathcal{N}(x_0; \boldsymbol{\chi}^{\theta}_{0|1}(x_1), \sigma_1^2\mathrm{I}_{\mathsf{d}_x})$. The parameter θ is obtained (Song et al., 2021a, Theorem 1) by solving the following optimization problem:

$$\theta_* \in \mathrm{argmin}_{\theta} \textstyle\sum_{t=1}^{n} (2\mathsf{d}_x\sigma_t^2\alpha_t)^{-1} \int \|\epsilon - \mathbf{e}^{\theta}(\sqrt{\alpha_t}x_0 + \sqrt{1-\alpha_t}\epsilon, t)\|_2^2 \mathcal{N}(\epsilon; 0_{\mathsf{d}_x}, \mathrm{I}_{\mathsf{d}_x})\mathsf{q}_{\mathrm{data}}(\mathrm{d}x_0)\mathrm{d}\epsilon\,. \tag{5.4}$$

Thus, $\mathbf{e}^{\theta_*}(X_t, t)$ might be seen as the predictor of the noise added to X_0 to obtain X_t (in the forward pass) and justifies the "prediction" terminology. The time 0 marginal $\mathsf{p}^{\theta_*}_0(x_0) = \int \mathsf{p}^{\theta_*}_{0:n}(x_{0:n})\mathrm{d}x_{1:n}$ which we will refer to as the *prior* is used as an approximation of $\mathsf{q}_{\mathrm{data}}$ and the time s marginal is $\mathsf{p}^{\theta_*}_s(x_s) = \int \mathsf{p}^{\theta_*}_{0:n}(x_{0:n})\mathrm{d}x_{1:s-1}\mathrm{d}x_{s+1:n}$. In the rest of the paper, we drop the dependence on the parameter θ_*. We define for all $v \in \mathbb{R}^{\ell}, w \in \mathbb{R}^k$, the concatenation operator $v^\frown w = [v^T, w^T]^T \in \mathbb{R}^{\ell+k}$. For $i \in [1:\ell]$, we let $v[i]$ the i-th coordinate of v.

Related works. The subject of Bayesian problems is very vast, and it is impossible to discuss here all the results obtained in this very rich literature. One of such domains is image restoration problems, such as deblurring, denoising inpainting, which are challenging problems in computer vision that involves restoring a partially observed degraded image. Deep learning techniques are widely used for this task Arjomand Bigdeli et al. (2017); Yeh et al. (2018); Xiang et al. (2023); Wei et al. (2022) with many of them relying on auto-encoders, VAEs Ivanov et al. (2018); Peng et al. (2021); Zheng et al. (2019), GANs Yeh et al. (2018); Zeng et al. (2022), or autoregressive transformers Yu et al. (2018); Wan et al. (2021). In what follows, we focus on methods based on denoising diffusion that has recently emerged as a way to produce high-quality realistic samples from the original data distribution on par with the best GANs in terms of image and audio generation, without the intricacies of adversarial training; see Sohl-Dickstein et al. (2015); Song et al. (2021c, 2022). Diffusion-based approaches do not require specific training for degradation types, making them much more versatile and computationally efficient. In Song et al. (2022), noisy linear inverse problems are proposed to be solved by diffusing the degraded observation forward, leading to intermediate observations $\{y_s\}_{s=0}^n$, and then running a modified backward process

that promotes consistency with y_s at each step s. The Denoising-Diffusion-Restoration model (DDRM) Kawar et al. (2022) also modifies the backward process so that the unobserved part of the state follows the backward process while the observed part is obtained as a noisy weighted sum between the noisy observation and the prediction of the state. As observed by Lugmayr et al. (2022), DDRM is very efficient, but the simple blending used occasionally causes inconsistency in the restoration process. DPS Chung et al. (2023) considers a backward process targeting the posterior. DPS approximates the score of the posterior using the Tweedie formula, which incorporates the learned score of the prior. The approximation error is quantified and shown to decrease when the noise level is large, i.e., when the posterior is close to the prior distribution. As shown in Section 5.3 with a very simple example, neither DDRM nor DPS can be used to sample the target posterior and therefore do not solve the Bayesian recovery problem (even if we run DDRM and DPS several time with independent initializations). Indeed, we show that DDRM and DPS produce samples under the "prior" distribution (which is generally captured very well by the denoising diffusion model), but which are not consistent with the observations (many samples land in areas with very low likelihood). In Trippe et al. (2023), the authors introduce SMCdiff, a Sequential Monte Carlo-based denoising diffusion model that aims at solving specifically the *inpainting problem*. SMCdiff produces a particle approximation of the conditional distribution of the non observed part of the state conditionally on a forward-diffused trajectory of the observation. The resulting particle approximation is shown to converge to the true posterior of the SGM under the assumption that the joint laws of the forward and backward processes coincide, which fails to be true in realistic setting. In comparison with SMCdiff, MCGdiff is a versatile approach that solves any Bayesian linear inverse problem while being consistent under mild assumptions. In parallel to our work, Wu et al. (2023) also developed a similar SMC based methodology but with a different proposal kernel.

5.2 The MCGdiff algorithm

In this section, we present our methodology for the inpainting problem (5.5), both with noise and without noise. The more general case is treated in Section 5.2.1. Let $\mathsf{d}_y \in [1 : \mathsf{d}_x - 1]$. In what follows we denote the d_y top coordinates of a vector $x \in \mathbb{R}^{\mathsf{d}_x}$ by $\overline{x}$ and the remaining coordinates by $\underline{x}$, so that $x = \overline{x}^\frown \underline{x}$. The inpainting problem is defined as

$$Y = \overline{X} + \sigma_y \varepsilon\,, \quad \varepsilon \sim \mathcal{N}(0, \mathrm{I}_{\mathsf{d}_y})\,, \quad \sigma \geq 0\,, \tag{5.5}$$

where $\overline{X}$ are the first d_y coordinates of a random variable $X \sim \mathsf{p}_0$. The goal is then to recover the law of the complete state X given a realisation $\mathbf{y}$ of the incomplete observation $\mathbf{Y}$ and the model (5.5).

Noiseless case. We begin by the case $\sigma_y = 0$. As the first d_y coordinates are observed exactly, we aim at infering the remaining coordinates of X,which correspond to $\underline{X}$. As such, given an observation y, we aim at sampling from the posterior $\phi_0^y(\underline{x}_0) \propto \mathsf{p}_0(y^\frown \underline{x}_0)$ with integral form

$$\phi_0^y(\underline{x}_0) \propto \int \mathsf{p}_n(x_n) \left\{ \textstyle\prod_{s=1}^{n-1} p_s(x_s|x_{s+1}) \right\} p_0(y^\frown \underline{x}_0|x_1) \mathrm{d}x_{1:n}\,. \tag{5.6}$$

To solve this problem, we propose to use SMC algorithms Doucet et al. (2001); Cappé et al. (2005b); Chopin and Papaspiliopoulos (2020), where a set of N random samples, referred to as particles, is iteratively updated to approximate the posterior distribution. The updates involve, at iteration s, selecting promising particles from the pool of particles $\xi_{s+1}^{1:N} = (\xi_{s+1}^1, \ldots, \xi_{s+1}^N)$ based on a weight function $\widetilde{\omega}_s$, and then apply a Markov transition p_s^y to obtain the samples $\xi_s^{1:N}$. The transition $p_s^y(x_s|x_{s+1})$ is designed to follow the backward process while guiding the d_y top coordinates of the pool of particles $\xi_s^{1:N}$ towards the measurement y. Note that under the backward dynamics (5.3), $\overline{X}_t$ and $\underline{X}_t$ are independent conditionally on X_{t+1} with transition kernels respectively $\overline{p}_t(\overline{x}_t|x_{t+1}) := \mathcal{N}(\overline{x}_t; \overline{\mathsf{m}}_{t+1}(x_{t+1}), \sigma_{t+1}^2 \mathrm{I}_{\mathsf{d}_y})$ and $\underline{p}_t(\underline{x}_t|x_{t+1}) := \mathcal{N}(\underline{x}_t; \underline{\mathsf{m}}_{t+1}(x_{t+1}), \sigma_{t+1}^2 \mathrm{I}_{\mathsf{d}_x - \mathsf{d}_y})$ where $\overline{\mathsf{m}}_{t+1}(x_{t+1}) \in \mathbb{R}^{\mathsf{d}_y}$ and $\underline{\mathsf{m}}_{t+1}(x_{t+1}) \in \mathbb{R}^{\mathsf{d}_x - \mathsf{d}_y}$

are such that $\mathsf{m}_{t+1}(x_{t+1}) = \overline{\mathsf{m}}_{t+1}(x_{t+1}) \frown \underline{\mathsf{m}}_{t+1}(x_{t+1})$ and the above kernels satisfy $p_t(x_t|x_{t+1}) = \overline{p}_t(\overline{x}_t|x_{t+1})\underline{p}_t(\underline{x}_t|x_{t+1})$. We consider the following proposal kernels for $t \in [1:n-1]$,

$$p_t^y(x_t|x_{t+1}) \propto p_t(x_t|x_{t+1})\overline{q}_{t|0}(\overline{x}_t|y)\,, \quad \text{where} \quad \overline{q}_{t|0}(\overline{x}_t|y) := \mathcal{N}(\overline{x}_t; \bar{\alpha}_t^{1/2} y, (1-\bar{\alpha}_t)\mathrm{I}_{\mathsf{d}_y})\,. \tag{5.7}$$

For the final step, we define $\underline{p}_0^y(\underline{x}_0|x_1) = \underline{p}_0(\underline{x}_0|x_1)$. Using standard Gaussian conjugation formulas, we obtain

$$p_t^y(x_t|x_{t+1}) = \underline{p}_t(\underline{x}_t|x_{t+1}) \cdot \mathcal{N}\left(\overline{x}_t; \mathsf{K}_t \alpha_t^{1/2} y + (1-\mathsf{K}_t)\overline{\mathsf{m}}_{t+1}(x_{t+1}), (1-\bar{\alpha}_t)\mathsf{K}_t \cdot \mathrm{I}_{\mathsf{d}_y}\right)\,,$$

where $\mathsf{K}_t := \sigma_{t+1}^2/(\sigma_{t+1}^2 + 1 - \alpha_t)$. For this procedure to target the posterior ϕ_0^y, the weight function $\widetilde{\omega}_s$ is chosen as follows; we set $\widetilde{\omega}_{n-1}(x_n) := \int p_{n-1}(x_{n-1}|x_n)\overline{q}_{n-1|0}(\overline{x}_{n-1}|y)\mathrm{d}x_{n-1} = \mathcal{N}\left(\alpha_{n-1}^{1/2} y; \overline{\mathsf{m}}_n(x_n), \sigma_n^2 + 1 - \alpha_{n-1}\right)$ and for $t \in [1:n-2]$,

$$\widetilde{\omega}_t(x_{t+1}) := \frac{\int p_t(x_t|x_{t+1})\overline{q}_{t|0}(\overline{x}_t|y)\mathrm{d}x_t}{\overline{q}_{t+1|0}(\overline{x}_{t+1}|y)} = \frac{\mathcal{N}\left(\alpha_t^{1/2} y; \overline{\mathsf{m}}_{t+1}(x_{s+1}), (\sigma_{t+1}^2 + 1 - \alpha_t)\mathrm{I}_{\mathsf{d}_y}\right)}{\mathcal{N}\left(\alpha_{t+1}^{1/2} y; \overline{x}_{t+1}, (1-\alpha_{t+1})\mathrm{I}_{\mathsf{d}_y}\right)}\,. \tag{5.8}$$

For the final step, we set $\widetilde{\omega}_0(x_1) := \overline{p}_0(y|\overline{x}_1)/\overline{q}_{1|0}(\overline{x}_1|y)$. The overall SMC algorithm targeting ϕ_0^y using the instrumental kernel (5.7) and weight function (5.8) is summarized in Algorithm 1. We now

Algorithm 1: `MCGdiff` ($\sigma = 0$)

Input: Number of particles N
Output: $\xi_0^{1:N}$

$\xi_n^{1:N} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}_{\mathsf{d}_x}, \mathrm{I}_{\mathsf{d}_x})$;
// Operations involving index i are repeated for each $i \in [1:N]$
for $s \leftarrow n-1:0$ **do**
  if $s = n-1$ **then**
    $\widetilde{\omega}_{n-1}(\xi_n^i) = \mathcal{N}\big(\bar{\alpha}_n^{1/2} y; \overline{\mathsf{m}}_n(\xi_n^i), 2 - \bar{\alpha}_n\big)$;
  else
    $\widetilde{\omega}_s(\xi_{s+1}^i) = \mathcal{N}\left(\bar{\alpha}_s^{1/2} y; \overline{\mathsf{m}}_{s+1}(\xi_{s+1}^i), \sigma_{s+1}^2 + 1 - \bar{\alpha}_s\right) / \mathcal{N}\left(\bar{\alpha}_{s+1}^{1/2} y; \overline{\xi}_{s+1}^i, 1 - \bar{\alpha}_{s+1}\right)$;
  $I_{s+1}^i \sim \mathrm{Cat}\big(\{\widetilde{\omega}_s(\xi_{s+1}^j)/\sum_{k=1}^N \widetilde{\omega}_s(\xi_{s+1}^k)\}_{j=1}^N\big)$, $\quad \overline{z}_s^i \sim \mathcal{N}(\mathbf{0}_{\mathsf{d}_y}, \mathrm{I}_{\mathsf{d}_y})$, $\quad \underline{z}_s^i \sim \mathcal{N}(\mathbf{0}_{\mathsf{d}_x - \mathsf{d}_y}, \mathrm{I}_{\mathsf{d}_x - \mathsf{d}_y})$;
  $\overline{\xi}_s^i = \mathsf{K}_s \bar{\alpha}_s^{1/2} y + (1-\mathsf{K}_s)\overline{\mathsf{m}}_{s+1}(\xi_{s+1}^{I_{s+1}^i}) + (1-\alpha_s)^{1/2}\mathsf{K}_s^{1/2}\overline{z}_s^i$, $\quad \underline{\xi}_s^i = \underline{\mathsf{m}}_{s+1}(\xi_{s+1}^{I_{s+1}^i}) + \sigma_{s+1}\underline{z}_s^i$;
  Set $\xi_s^i = \overline{\xi}_s^i \frown \underline{\xi}_s^i$;

provide a justification to Algorithm 1. Let $\{g_s^y\}_{s=1}^n$ be a sequence of positive functions with $g_n^y \equiv 1$. Consider the sequence of distributions $\{\phi_s^y\}_{s=1}^n$ defined as follows; $\phi_n^y(x_n) \propto g_n^y(x_n)\mathsf{p}_n(x_n)$ and for $t \in [1:n-1]$

$$\phi_t^y(x_t) \propto \int g_{t+1}^y(x_{t+1})^{-1} g_t^y(x_t) p_t(x_t|x_{t+1})\phi_{t+1}^y(\mathrm{d}x_{t+1})\,. \tag{5.9}$$

By construction, the time t marginal (5.9) is $\phi_t^y(x_t) \propto \mathsf{p}_t(x_t)g_t^y(x_t)$ for all $t \in [1:n]$. Then, using ϕ_1^y and (5.6), we have that

$$\phi_0^y(\underline{x}_0) \propto \int g_1^y(x_1)^{-1}\overline{p}_0(y|\overline{x}_1)\underline{p}_0(\underline{x}_0|x_1)\phi_1^y(\mathrm{d}x_1)\,. \tag{5.10}$$

The recursion (5.9) suggests a way of obtaining a particle approximation of ϕ_0^y; by sequentially approximating each ϕ_t^y we can effectively derive a particle approximation of the posterior.To construct the intermediate particle approximations we use the framework of *auxiliary particle filters* (APF) (Pitt and Shephard, 1999). We focus on the case $g_t^y(x_t) = \overline{q}_{t|0}(\overline{x}_t|y)$ which corresponds to Algorithm 1.

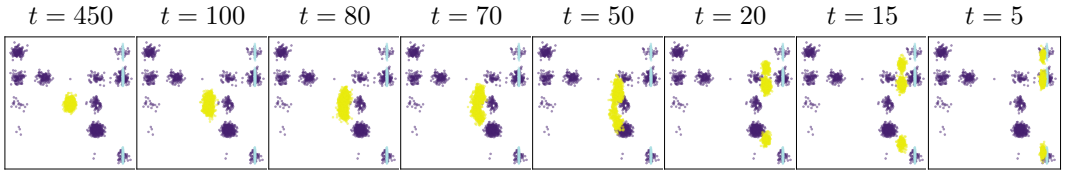


Figure 5.1: Display of samples from $\phi_t^y(x_t) \propto \mathsf{p}_t(x_t)\overline{q}_{t|0}(\overline{x}_t|y)$ for the GM prior. Samples from ϕ_t^y (yellow), those from the prior (purple) and those from the posterior ϕ_0^y (light blue) with $n = 500$.

The initial particle approximation ϕ_n^y is obtained by drawing N i.i.d. samples $\xi_n^{1:N}$ from p_n and setting $\phi_n^N = N^{-1}\sum_{i=1}^N \delta_{\xi_n^i}$ where δ_ξ is the Dirac mass at ξ. Assume that the empirical approximation of ϕ_{t+1}^y is $\phi_{t+1}^N = N^{-1}\sum_{i=1}^N \delta_{\xi_{t+1}^i}$, where $\xi_{t+1}^{1:N}$ are N random variables. Substituting ϕ_{t+1}^N into the recursion (5.9) and introducing the instrumental kernel (5.7), we obtain the mixture

$$\widehat{\phi}_t^N(x_t) = \textstyle\sum_{i=1}^N \widetilde{\omega}_t(\xi_{t+1}^i) p_t^y(x_t|\xi_{t+1}^i) / \sum_{j=1}^N \widetilde{\omega}_t(\xi_{t+1}^j)\,. \tag{5.11}$$

Then, a particle approximation of (5.11) is obtained by sampling N conditionally i.i.d. ancestor indices $I_{t+1}^{1:N} \overset{\text{i.i.d.}}{\sim} \mathrm{Cat}(\{\widetilde{\omega}_t(\xi_{t+1}^i)/\sum_{j=1}^N \widetilde{\omega}_t(\xi_{t+1}^j)\}_{i=1}^N)$, and then propagating each ancestor particle $\xi_{t+1}^{I_{t+1}^i}$ according to the instrumental kernel (5.7). The final particle approximation is given by $\phi_0^N = N^{-1}\sum_{i=1}^N \delta_{\xi_0^i}$, where $\xi_0^i \sim \underline{p}_0(\cdot|\xi_1^{I_1^i})$, $I_1^i \sim \mathrm{Cat}(\{\widetilde{\omega}_0(\xi_1^k)/\sum_{j=1}^N \widetilde{\omega}_0(\xi_1^j)\}_{k=1}^N)$. The sequence of distributions $\{\mathsf{p}_t\}_{t=0}^n$ approximating the marginals of the forward process initialized at p_0 defines a path that bridges between p_n and the prior p_0 such that the discrepancy between p_t and p_{t+1} is small. SMC samplers based on this path are robust to multi-modality and offer an interesting alternative to the geometric and tempering paths traditionally used in the SMC literature, see Dai et al. (2022). Our proposals $\phi_t^y(x_t) \propto \mathsf{p}_t(x_t)\overline{q}_{t|0}(\overline{x}_t|y)$ inherit the behavior of $\{\mathsf{p}_t\}_{t\in\mathbb{N}}$ and bridge the initial distribution ϕ_n^y and posterior ϕ_0^y. Indeed, as y is a noiseless observation of $X_0 \sim \mathsf{p}_0$, we may consider $\bar{\alpha}_t^{1/2} y + (1-\bar{\alpha}_t)^{1/2}\varepsilon_t$, with $\varepsilon_t \sim \mathcal{N}(\mathbf{0}_{\mathsf{d}_y}, \mathrm{I}_{\mathsf{d}_y})$, as a noisy observation of $X_t \sim \mathsf{p}_t$ and thus, ϕ_t^y is the associated posterior. We illustrate this intuition by considering the following Gaussian mixture (GM) example. We assume that $\mathsf{p}_0(x_0) = \sum_{i=1}^M w_i \cdot \mathcal{N}(x_0; \mu_i, \mathrm{I}_{\mathsf{d}_x})$ where $M > 1$ and $\{w_i\}_{i=1}^M$ are drawn uniformly on the simplex. The marginals of the forward process are available in closed form and are given by $\mathsf{p}_t(x_t) = \sum_{i=1}^M w_i \cdot \mathcal{N}(x_t; \bar{\alpha}_t^{1/2}\mu_i, \mathrm{I}_{\mathsf{d}_x})$, which shows that the discrepancy between p_t and p_{t+1} is small as long as $\bar{\alpha}_t^{1/2} - \bar{\alpha}_{t+1}^{1/2}$ is close to 0. The posteriors $\{\phi_t^y\}_{t\in[0:n]}$ are also available in closed form and displayed in Figure 5.1, which illustrates that our choice of potentials ensures that the discrepancy between consecutive posteriors is small. The idea of using the forward diffused observation to guide the observed part of the state, as we do here through $\overline{q}_t(\overline{x}_t|y)$, has been exploited in prior works but in a different way. For instance, in Song et al. (2021c, 2022) the observed part of the state is directly replaced by the forward noisy observation and, as it has been noted Trippe et al. (2023), this introduces an irreducible bias. Instead, `MCGdiff` weights the backward process by the density of the forward one conditioned on y, resulting in a natural and consistent algorithm.

We now establish the convergence of `MCGdiff` with a general sequence of *potentials* $\{g_s^y\}_{s=1}^n$. We consider the following assumption on the sequence of potentials $\{g_t^y\}_{t=1}^n$.

(A1) $\sup_{x\in\mathbb{R}^{\mathsf{d}_x}} \overline{p}_0(y|x)/g_1^y(x) < \infty$ and $\sup_{x\in\mathbb{R}^{\mathsf{d}_x}} \int g_t^y(x_t) p_t(x_t|x)\mathrm{d}x_t / g_{t+1}^y(x) < \infty$ for all $t \in [1:n-1]$.

The following exponential deviation inequality is standard and is a direct application of (Douc et al., 2014, Theorem 10.17). In particular, it implies a $\mathcal{O}(1/\sqrt{N})$ bound on the mean squared error $\|\phi_0^N(h) - \phi_0^y(h)\|_2$.

Proposition 32. *Assume (A1). There exist constants $c_{1,n}, c_{2,n} \in (0,\infty)$ such that, for all $N \in \mathbb{N}$, $\varepsilon > 0$ and bounded function $h : \mathbb{R}^{\mathsf{d}_x} \mapsto \mathbb{R}$, $\mathbb{P}\left[|\phi_0^N(h) - \phi_0^y(h)| \geq \varepsilon\right] \leq c_{1,n}\exp(-c_{2,n}N\varepsilon^2/|h|_\infty^2)$ where $|h|_\infty := \sup_{x\in\mathbb{R}^{\mathsf{d}_x}} |h(x)|$.*

We also furnish our estimator with an explicit non-asymptotic bound on its bias. Define $\Phi_0^N = \mathbb{E}[\phi_0^N]$ where $\phi_0^N = N^{-1}\sum_{i=1}^N \delta_{\underline{\xi}_0^i}$ is the particle approximation produced by Algorithm 1 and the expectation is with respect to the law of $(\underline{\xi}_{0:n}^{1:N}, I_{1:n}^{1:N})$. Define for all $t \in [1:n]$, $\phi_t^\star(x_t) \propto \mathsf{p}_t(x_t)\int \delta_y(\mathrm{d}\overline{x}_0)p_{0|t}(x_0|x_t)\mathrm{d}\underline{x}_0$, where $p_{0|t}(x_0|x_t) := \int \left\{\prod_{s=0}^{t-1} p_s(x_s|x_{s+1})\right\}\mathrm{d}x_{1:t-1}$.

Proposition 33. *It holds that*

$$\mathsf{KL}(\phi_0^y \parallel \Phi_0^N) \leq \mathsf{C}_{0:n}^y (N-1)^{-1} + \mathsf{D}_{0:n}^y N^{-2}\,, \tag{5.12}$$

where $\mathsf{D}_{0:n}^y > 0$, $\mathsf{C}_{0:n}^y := \sum_{t=1}^n \int \frac{\mathcal{Z}_t/\mathcal{Z}_0}{g_t^y(z_t)}\left\{\int \delta_y(\mathrm{d}\overline{x}_0)p_{0|t}(x_0|z_t)\mathrm{d}\underline{x}_0\right\}\phi_t^\star(\mathrm{d}z_t)$ *and* $\mathcal{Z}_t := \int g_t^y(x_t)\mathsf{p}_t(\mathrm{d}x_t)$ *for all* $t \in [1:n]$ *and* $\mathcal{Z}_0 := \int \delta_y(\mathrm{d}\overline{x}_0)\mathsf{p}_0(x_0)\mathrm{d}\underline{x}_0$. *If furthermore (**A1**) holds then both* $\mathsf{C}_{0:n}^y$ *and* $\mathsf{D}_{0:n}^y$ *are finite.*

The proof of Proposition 33 is postponed to Section D.2.1. (**A1**) is an assumption on the equivalent of the weights $\{\widetilde{\omega}_t\}_{t=0}^n$ with a general sequence of potentials $\{g_t^y\}_{t=1}^n$ and is not restrictive as it can be satisfied by setting for example $g_s^y(x_s) = \overline{q}_{s|0}(\overline{x}_s|y) + \delta$ where $\delta > 0$. The resulting algorithm is then only a slight modification of the one described above, see Section D.2.1 for more details. It is also worth noting that Proposition 33 combined with Pinsker's inequality implies that the bias of `MCGdiff` goes to 0 with the number of particle samples N for fixed n. We have chosen to present a bound in Kullback–Leibler (KL) divergence, inspired by Andrieu et al. (2018); Huggins and Roy (2019), as it allows an explicit dependence on the modeling choice $\{g_s^y\}_{s=1}^n$, see Lemma 67. Finally, unlike the theoretical guarantees established for `SMCdiff` in Trippe et al. (2023), proving the asymptotic exactness of our methodology w.r.t. to the generative model posterior does not require having $\mathsf{p}_{s+1}(x_{s+1})p_s(x_s|x_{s+1}) = \mathsf{p}_s(x_s)q_{s+1}(x_{s+1}|x_s)$ for all $s \in [0:n-1]$, which does not hold in practice.

Noisy case. We consider the case $\sigma_y > 0$. The posterior density is given by $\phi_0^y(x_0) \propto g_0^y(\overline{x}_0)\mathsf{p}_0(x_0)$, where $g_0^y(x_0) := \mathcal{N}(y; \overline{x}_0, \sigma_y^2 \mathrm{I}_{\mathsf{d}_y})$. In what follows, assume that there exists $\tau \in [1:n]$ such that $\sigma^2 = (1-\bar{\alpha}_\tau)/\bar{\alpha}_\tau$. We denote $\tilde{y}_\tau = \bar{\alpha}_\tau^{1/2} y$. We can then write that

$$g_0^y(\overline{x}_0) = \bar{\alpha}_\tau^{1/2}\cdot\mathcal{N}(\tilde{y}_\tau; \bar{\alpha}_\tau^{1/2}x_0, (1-\bar{\alpha}_\tau)\cdot\mathrm{I}_{\mathsf{d}_y}) = \bar{\alpha}_\tau^{1/2}\cdot\overline{q}_{\tau|0}(\tilde{y}_\tau|\overline{x}_0)\,, \tag{5.13}$$

which hints that the likelihood function g_0^y is closely related to the forward process (5.1). We may then write the posterior $\phi_0^y(x_0)$ as $\phi_0^y(x_0) \propto \overline{q}_{\tau|0}(\tilde{y}_\tau|\overline{x}_0)\mathsf{p}_0(x_0) \propto \int \delta_{\tilde{y}_\tau}(\mathrm{d}\overline{x}_\tau)q_{\tau|0}(x_\tau|x_0)\mathsf{p}_0(x_0)\mathrm{d}\underline{x}_\tau$. Next, assume that the forward process (5.1) is the reverse of the backward one (5.3), i.e. that

$$\mathsf{p}_t(x_t)q_{t+1}(x_{t+1}|x_t) = \mathsf{p}_{t+1}(x_{t+1})p_t(x_t|x_{t+1})\,, \quad \forall t \in [0:n-1]\,. \tag{5.14}$$

This is similar to the assumption made in `SMCdiff` Trippe et al. (2023). Then, it is easily seen that it implies $\mathsf{p}_0(x_0)q_{\tau|0}(x_\tau|x_0) = \mathsf{p}_\tau(x_\tau)p_{0|\tau}(x_0|x_\tau)$ and thus

$$\phi_0^y(x_0) = \int p_{0|\tau}(x_0|x_\tau)\delta_{\tilde{y}_\tau}(\mathrm{d}\overline{x}_\tau)\mathsf{p}_\tau(x_\tau)\mathrm{d}\underline{x}_\tau \Big/ \int \delta_{\tilde{y}_\tau}(\mathrm{d}\overline{z}_\tau)\mathsf{p}_\tau(z_\tau)\mathrm{d}\underline{z}_\tau = \int p_{0|\tau}(x_0|\tilde{y}_\tau{}^\frown\underline{x}_\tau)\phi_\tau^{\tilde{y}_\tau}(\mathrm{d}\underline{x}_\tau)\,, \tag{5.15}$$

where $\phi_\tau^{\tilde{y}_\tau}(\underline{x}_\tau) \propto \mathsf{p}_\tau(\tilde{y}_\tau{}^\frown\underline{x}_\tau)$. (5.15) highlights that solving the inverse problem (5.5) with $\sigma_y > 0$ is equivalent to solving an inverse problem on the intermediate state $X_\tau \sim \mathsf{p}_\tau$ with *noiseless* observation $\tilde{y}_\tau$ of the d_y top coordinates and then propagating the resulting posterior back to time 0 with the backward kernel $p_{0|\tau}$. The assumption (5.14) does not always holds in realistic settings.Therefore, while (5.15) also holds only approximately in practice, we can still use it as inspiration for designing potentials when the assumption is not valid. Consider then $\{g_t^y\}_{t=\tau}^n$ and sequence of probability measures $\{\phi_t^y\}_{t=\tau}^n$ defined for all $t \in [\tau:n]$ as $\phi_t^y(x_t) \propto g_t^y(x_t)\mathsf{p}_t(x_t)$, where $g_t^y(x_t) := \mathcal{N}(x_t; \bar{\alpha}_t^{1/2}y, (1-(1-\kappa)\bar{\alpha}_t/\bar{\alpha}_\tau)\mathrm{I}_{\mathsf{d}_y})$, $\kappa \geq 0$. In the case of $\kappa = 0$, we have $g_t^y(x_t) = \overline{q}_{t|\tau}(\overline{x}_t|\tilde{y}_\tau)$ for $t \in [\tau+1:n]$ and $\phi_\tau^y = \phi_\tau^{\tilde{y}_\tau}$. The recursion (5.9)

holds for $t \in [\tau : n]$ and assuming $\kappa > 0$, we find that $\phi_0^y(x_0) \propto g_0^y(x_0) \int g_\tau^y(x_\tau)^{-1} p_{0|\tau}(x_0|x_\tau) \phi_\tau^y(\mathrm{d}x_\tau)$, which resembles the recursion (5.15). In practice we take κ to be small in order to mimick the Dirac delta mass at $\overline{x}_\tau$ in (5.15). Having a particle approximation $\phi_\tau^N = N^{-1} \sum_{i=1}^N \delta_{\xi_\tau^i}$ of ϕ_τ^y by adapting Algorithm 1, we estimate ϕ_0^y with $\phi_0^N = \sum_{i=1}^N \omega_0^i \delta_{\xi_0^i}$ where $\xi_0^i \sim p_{0|\tau}(\cdot|\xi_\tau^i)$ and $\omega_0^i \propto g_0^y(\xi_0^i)/g_\tau^y(\xi_\tau^{I_i^i})$. In the next section we extend this methodology to general linear Gaussian observation models. Finally, (5.15) allows us to extend `SMCdiff` to handle noisy inverse problems in a principled manner which is detailed in Section D.1.

5.2.1 Extension to general linear inverse problems

Consider $Y = \mathrm{A}X + \sigma_y \varepsilon$ where $\mathrm{A} \in \mathbb{R}^{\mathsf{d}_y \times \mathsf{d}_x}$, $\varepsilon \sim \mathcal{N}(0_{\mathsf{d}_y}, \mathrm{I}_{\mathsf{d}_y})$ and $\sigma_y \geq 0$ and the singular value decomposition (SVD) $\mathrm{A} = \mathrm{U}\mathrm{S}\overline{\mathrm{V}}^T$, where $\overline{\mathrm{V}} \in \mathbb{R}^{\mathsf{d}_x \times \mathsf{d}_\mathsf{y}}$, $\mathrm{U} \in \mathbb{R}^{\mathsf{d}_y \times \mathsf{d}_\mathsf{y}}$ are two orthonormal matrices, and $\mathrm{S} \in \mathbb{R}^{\mathsf{d}_\mathsf{y} \times \mathsf{d}_\mathsf{y}}$ is diagonal. For simplicity, it is assumed that the singular values satisfy $s_1 > \cdots > s_{\mathsf{d}_\mathsf{y}} > 0$. Set $\mathsf{b} = \mathsf{d}_x - \mathsf{d}_\mathsf{y}$. Let $\underline{\mathrm{V}} \in \mathbb{R}^{\mathsf{d}_x \times \mathsf{b}}$ be an orthonormal matrix of which the columns complete those of $\overline{\mathrm{V}}$ into an orthonormal basis of $\mathbb{R}^{\mathsf{d}_x}$, i.e. $\underline{\mathrm{V}}^T \underline{\mathrm{V}} = \mathrm{I}_\mathsf{b}$ and $\underline{\mathrm{V}}^T \overline{\mathrm{V}} = \mathbf{0}_{\mathsf{b},\mathsf{d}_\mathsf{y}}$. We define $\mathrm{V} = [\overline{\mathrm{V}}, \underline{\mathrm{V}}] \in \mathbb{R}^{\mathsf{d}_x \times \mathsf{d}_x}$. In what follows, for a given $\mathbf{x} \in \mathbb{R}^{\mathsf{d}_x}$ we write $\overline{\mathbf{x}} \in \mathbb{R}^{\mathsf{d}_\mathsf{y}}$ for its top d_y coordinates and $\underline{\mathbf{x}} \in \mathbb{R}^\mathsf{b}$ for the remaining coordinates. Setting $\mathbf{X} := \mathrm{V}^T X$ and $\mathbf{Y} := \mathrm{S}^{-1}\mathrm{U}^T Y$ and multiplying the measurement equation by $\mathrm{S}^{-1}\mathrm{U}^T$ yields

$$\mathbf{Y} = \overline{\mathbf{X}} + \sigma_y \mathrm{S}^{-1} \widetilde{\varepsilon}, \quad \widetilde{\varepsilon} \sim \mathcal{N}(0, \mathrm{I}_{\mathsf{d}_\mathsf{y}}).$$

In this section, we focus on solving this linear inverse problem in the orthonormal basis defined by V using the methodology developed in the previous sections. This prompts us to define the diffusion based generative model in this basis. As V is an orthonormal matrix, the law of $\mathbf{X}_0 = \mathrm{V}^T X_0$ is $\mathfrak{p}_0(\mathbf{x}_0) := \mathsf{p}_0(\mathrm{V}\mathbf{x}_0)$. By definition of $\mathfrak{p}_0$ and the fact that $\|\mathrm{V}\mathbf{x}\|_2 = \|\mathbf{x}\|_2$ for all $\mathbf{x} \in \mathbb{R}^{\mathsf{d}_x}$ we have that

$$\mathfrak{p}_0(\mathbf{x}_0) = \int p_0(\mathrm{V}\mathbf{x}_0|x_1) \left\{ \prod_{s=1}^{n-1} p_s(\mathrm{d}x_s|x_{s+1}) \right\} \mathsf{p}_n(\mathrm{d}x_n) = \int \lambda_0(\mathbf{x}_0|\mathbf{x}_1) \left\{ \prod_{s=1}^{n-1} \lambda_s(\mathrm{d}\mathbf{x}_s|\mathbf{x}_{s+1}) \right\} \mathfrak{p}_n(\mathrm{d}\mathbf{x}_n)$$

where for all $s \in [1 : n]$, $\lambda_{s-1}(\mathbf{x}_{s-1}|\mathbf{x}_s) := \mathcal{N}(\mathbf{x}_{s-1}; \mathfrak{m}_s(\mathbf{x}_s), \sigma_s^2 \mathrm{I}_{\mathsf{d}_x})$, where $\mathfrak{m}_s(\mathbf{x}_s) := \mathrm{V}^T \mathsf{m}_s(\mathrm{V}\mathbf{x}_s)$. The transition kernels $\{\lambda_s\}_{s=0}^{n-1}$ thus define a diffusion based model in the basis V. In what follows we write $\overline{\mathfrak{m}}_s(\mathbf{x}_s)$ for the first d_y coordinates of $\mathfrak{m}_s(\mathbf{x}_s)$ and $\underline{\mathfrak{m}}_s(\mathbf{x}_s)$ the last b coordinates. We denote by $\mathfrak{p}_s$ the time s marginal of the backward process.

Noiseless. In this case the target posterior is $\phi_0^\mathbf{y}(\mathbf{x}_0) \propto \mathfrak{p}_0(\mathbf{y}^\frown \underline{\mathbf{x}}_0)$. The extension of algorithm 1 is straight forward; it is enough to replace y with $\mathbf{y}$ (=$\mathrm{S}^{-1}\mathrm{U}^T y$) and the backward kernels $\{p_t\}_{t=0}^{n-1}$ with $\{\lambda_t\}_{t=0}^{n-1}$.

Noisy. The posterior density is then $\phi_0^\mathbf{y}(\mathbf{x}_0) \propto g_0^\mathbf{y}(\overline{\mathbf{x}}_0)\mathfrak{p}_0(\mathbf{x}_0)$, where

$$g_0^\mathbf{y}(\overline{\mathbf{x}}_0) = \prod_{i=1}^{\mathsf{d}_\mathsf{y}} \mathcal{N}(\mathbf{y}[i]; \overline{\mathbf{x}}_0[i], (\sigma_y/s_i)^2).$$

As in Line 9, assume that there exists $\{\tau_i\}_{i=1}^{\mathsf{d}_\mathsf{y}} \subset [1 : n]$ such that $\bar{\alpha}_{\tau_i}\sigma_y^2 = (1 - \bar{\alpha}_{\tau_i})s_i^2$ and define for all $i \in [1 : \mathsf{d}_\mathsf{y}]$, $\tilde{\mathbf{y}}_i := \bar{\alpha}_{\tau_i}^{1/2} \mathbf{y}[i]$. Then we can write the potential $g_0^\mathbf{y}$ in a similar fashion to (5.13) as the product of forward processes from time 0 to each time step τ_i, i.e. $g_0^\mathbf{y}(\mathbf{x}_0) = \prod_{i=1}^{\mathsf{d}_\mathsf{y}} \bar{\alpha}_{\tau_i}^{1/2} \mathcal{N}(\tilde{\mathbf{y}}_i; \bar{\alpha}_{\tau_i}^{1/2} \mathbf{x}_0[i], (1 - \bar{\alpha}_{\tau_i}))$. Writing the potential this way allows us to generalize (5.15) as follows. Denote for $\ell \in [1 : \mathsf{d}_x]$, $\mathbf{x}^{\setminus \ell} \in \mathbb{R}^{\mathsf{d}_x - 1}$ the vector $\mathbf{x}$ with its ℓ-th coordinate removed. Define

$$\phi_{\tau_1:n}^{\tilde{\mathbf{y}}}(\mathrm{d}\mathbf{x}_{\tau_1:n}) \propto \left\{ \prod_{i=1}^{\mathsf{d}_\mathsf{y}-1} \lambda_{\tau_i|\tau_{i+1}}(\mathbf{x}_{\tau_i}|\mathbf{x}_{\tau_{i+1}}) \delta_{\tilde{\mathbf{y}}_i}(\mathrm{d}\mathbf{x}_{\tau_i}[i]) \mathrm{d}\mathbf{x}_{\tau_i}^{\setminus i} \right\} \mathfrak{p}_{\tau_{\mathsf{d}_\mathsf{y}}}(\mathbf{x}_{\tau_{\mathsf{d}_\mathsf{y}}}) \delta_{\tilde{\mathbf{y}}_{\mathsf{d}_\mathsf{y}}}(\mathrm{d}\mathbf{x}_{\tau_{\mathsf{d}_\mathsf{y}}}[\mathsf{d}_\mathsf{y}]) \mathrm{d}\mathbf{x}_{\tau_{\mathsf{d}_\mathsf{y}}}^{\setminus \mathsf{d}_\mathsf{y}},$$

which corresponds to the posterior of a noiseless inverse problem on the joint states $\mathbf{X}_{\tau_1:n} \sim \mathfrak{p}_{\tau_1:n}$ with noiseless observations $\tilde{\mathbf{y}}_{\tau_i}$ of $\mathbf{X}_{\tau_i}[i]$ for all $i \in [1 : \mathsf{d}_\mathsf{y}]$.

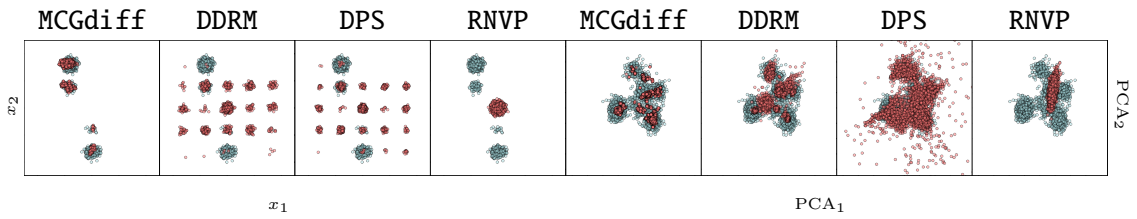


Figure 5.2: The first and last four columns correspond respectively to GM with $(\mathsf{d}_x, \mathsf{d}_\mathbf{y}) = (800, 1)$ *and FM with* $(\mathsf{d}_x, \mathsf{d}_\mathbf{y}) = (10, 1)$*. The blue and red dots represent respectively samples from the exact posterior and those generated by each of the algorithms used (names on top).*

d	d_y	MCGdiff	DDRM	DPS	RNVP
80	1	**1.39 ± 0.45**	5.64 ± 1.10	4.98 ± 1.14	6.86 ± 0.88
80	2	**0.67 ± 0.24**	7.07 ± 1.35	5.10 ± 1.23	7.79 ± 1.50
80	4	**0.28 ± 0.14**	7.81 ± 1.48	4.28 ± 1.26	7.95 ± 1.61
800	1	**2.40 ± 1.00**	7.44 ± 1.15	6.49 ± 1.16	7.74 ± 1.34
800	2	**1.31 ± 0.60**	8.95 ± 1.12	6.88 ± 1.01	8.75 ± 1.02
800	4	**0.47 ± 0.19**	8.39 ± 1.48	5.51 ± 1.18	7.81 ± 1.63

d	d_y	MCGdiff	DDRM	DPS	RNVP
6	1	**1.95 ± 0.43**	4.20 ± 0.78	5.43 ± 1.05	6.16 ± 0.65
6	3	**0.73 ± 0.33**	2.20 ± 0.67	3.47 ± 0.78	4.70 ± 0.90
6	5	**0.41 ± 0.12**	0.91 ± 0.43	2.07 ± 0.63	3.52 ± 0.93
10	1	**2.45 ± 0.42**	3.82 ± 0.64	4.30 ± 0.91	6.04 ± 0.38
10	3	**1.07 ± 0.26**	4.94 ± 0.87	5.38 ± 0.84	5.91 ± 0.64
10	5	**0.71 ± 0.12**	2.32 ± 0.74	3.74 ± 0.77	5.11 ± 0.69

Table 5.1: Sliced Wasserstein for the GM (left) and FM (right) case.

Proposition 34. *Assume that* $\mathfrak{p}_{s+1}(\mathbf{x}_{s+1})\lambda_s(\mathbf{x}_s|\mathbf{x}_{s+1}) = \mathfrak{p}_s(\mathbf{x}_s)q_{s+1}(\mathbf{x}_{s+1}|\mathbf{x}_s)$ *for all* $s \in [0 : n-1]$*. Then it holds that* $\phi_0^\mathbf{y}(\mathbf{x}_0) \propto \int \lambda_{0|\tau_1}(\mathbf{x}_0|\mathbf{x}_{\tau_1})\phi^{\tilde{\mathbf{y}}}_{\tau_1:n}(\mathrm{d}\mathbf{x}_{\tau_1:n})$.

The proof of Proposition 34 is given in Section D.2.2. We have shown that sampling from $\phi_0^\mathbf{y}$ is equivalent to sampling from $\phi^{\tilde{\mathbf{y}}}_{\tau_1:n}$ then propagating the final state $\mathbf{X}_{\tau_1}$ to time 0 according to $\lambda_{0|\tau_1}$. Therefore, as in (5.13), we define $\{g_t^y\}_{t=\tau}^n$ and $\{\phi_t^\mathbf{y}\}_{t=\tau}^n$ for all $t \in [\tau_1 : n]$ by $\phi_t^\mathbf{y}(\mathbf{x}_t) \propto g_t^\mathbf{y}(\mathbf{x}_t)\mathfrak{p}_t(\mathbf{x}_t)$ and $g_t^\mathbf{y}(\mathbf{x}_t) := \prod_{i=1}^{\tau(t)} \mathcal{N}(\mathbf{x}_t; \tilde{\mathbf{y}}_i, 1 - (1-\kappa)\bar{\alpha}_t/\bar{\alpha}_{\tau_i})$, $\kappa > 0$. We obtain a particle approximation of $\phi_{\tau_1}^\mathbf{y}$ using a particle filter with proposal kernel and weight function $\lambda_t^\mathbf{y}(\mathbf{x}_t|\mathbf{x}_{t+1}) \propto g_t^\mathbf{y}(\mathbf{x}_t)p_t(\mathbf{x}_t|\mathbf{x}_{t+1})$, $\widetilde{\omega}_t(\mathbf{x}_{t+1}) = \int g_t^\mathbf{y}(\mathbf{x}_t)p_t(\mathrm{d}\mathbf{x}_t|\mathbf{x}_{t+1})/g_{t+1}^\mathbf{y}(\mathbf{x}_{t+1})$, which are both available in closed form.

5.3 Numerics

The focus of this work is on providing an algorithm that consistently approximates the posterior distribution of a linear inverse problem with Gaussian noise. A prerequisite for quantitative evaluation in ill-posed inverse problems in a Bayesian setting is to have access to samples of the posterior distribution. This generally requires having at least an unnormalized proxy of the posterior density, so that one can run MCMC samplers such as the No U-turn sampler (NUTS) Hoffman and Gelman (2011). Therefore, this section focus on mixture models of two types of basis distribution, the Gaussian and the Funnel distributions. We then present a brief illustration of `MCGdiff` on image data. However, in this setting, the actual posterior distribution is unknown and the main goal is to explore the potentially multimodal posterior distribution, which makes a comparison with a "real image" meaningless. Therefore, metrics such as Fréchet Inception Distance (FID) and LPIPS score, which require comparison to a ground truth, are not useful for evaluating Bayesian reconstruction methods in such settings.[1]

Mixture Models: We refer to the Funnel mixture prior as FM prior (see section D.3 for the definition). For GM prior, we consider a mixture of 25 components with pre-established means and variances. For FM prior, we consider a mixture of 20 components consisting of rotated and translated funnel distributions. For a given pair $(\mathsf{d}_x, \mathsf{d}_\mathbf{y})$, we sample a prior distribution by randomly sampling the weights of the mixture and for the FM case the translation and rotation of each component. We then randomly

[1]The code for the experiments is available at https://github.com/gabrielvc/mcg_diff.

Figure 5.3: Illustration of the samples of `MCGdiff` *for different datasets and different inverse problems.*

sample measurement models $(y, \mathrm{A}, \sigma_y) \in \mathbb{R}^{\mathtt{d_y}} \times \mathbb{R}^{\mathtt{d_y} \times \mathtt{d}_x} \times [0, 1]$. For each pair of prior distribution and measurement model, we generate 10^4 samples from `MCGdiff`, `DPS`, `DDRM`, `RNVP`, and from the posterior either analytically (GM) or using NUTS (FM). We calculate for each algorithm the sliced Wasserstein (SW) distance between the resulting samples and the posterior samples. Table 5.1 shows the CLT 95% confidence intervals obtained over 20 seeds. Figure 5.2 illustrate the samples for the different algorithms for a given seed. We see that `MCGdiff` outperforms all the other algorithms in each setting tested. The complete details of the numerical experiments performed in this section is available in section D.3 as well as an additional visualisations.

Image datasets: Figure 5.3 shows samples of `MCGdiff` in different datasets (Celeb, Churches, Bedroom and Flowers) for different inverse problems, namely Inpaiting (Inp), super resolution (SR), Gaussian 2D deblur (G2Deb) and Colorization (Col). Visual comparison with competing algorithms and different datasets are shown in section D.3 as well as numerical details concerning figure 5.3.

Chapter 6

ECG-DIFF: Bayesian ECG Reconstruction using MCG-DIFF

6.1 Introduction

Electrocardiograms (ECGs) are essential tools for diagnosing various cardiac conditions. They record the heart's electrical activity using several electrodes placed on the chest, and highlight the different phases of the cardiac cycle, including the R peak, the QT segment and the ST segment. Myocardial infarction (MI), also known as heart attack, is an example of a critical diagnosis identified from ECG Jameson et al. (2018). An MI occurs when part of the heart muscle is deprived of oxygen, causing permanent damage. Accurate and rapid diagnosis of infarction is crucial, as treatment varies according to ECG morphology and must be carried out as quickly as possible Reed et al. (2017). For example, an infarct with ST-segment elevation may require invasive interventions such as percutaneous coronary intervention, which is not the case for intractus without ST-segment elevation Ibanez et al. (2017). However, the study of ECG morphology is complex and requires special expertise and attention. In particular, the morphology of each phase of the cardiac cycle, in each lead, as well as their coherence between leads, are crucial for assessing the electrical functioning of the heart. In addition, ECGs can be affected by noise and imperfect electrode placement, which affect recording quality. Therefore, methods that can accurately and impartially highlight morphological abnormalities, while denoising and reconstructing altered or missing signals, could be very useful for ECG analysis. In this article, we present a flexible method for addressing multiple challenges in ECG analysis: noise reduction, missing data reconstruction, and anomaly detection. To this end, we formulate these problems as inverse linear problems, meaning data reconstruction problems from incomplete and/or noisy observations. Our method relies on a trained model capable of generating ECGs, which is used as prior information to solve these inverse linear problems. This model is trained only once and is used for all tasks without requiring tuning for each of these tasks. Generative diffusion models have proven to be well-suited as priors in solving inverse problems Song et al. (2021a); Chung et al. (2023); Song et al. (2022); Kawar et al. (2022, 2021); Cardoso et al. (2023b); Wu et al. (2023). We adapt Cardoso et al. (2023b) for cases with unknown measurement noise levels, proposing a noise calibration strategy coupled with inverse problem resolution to simultaneously infer ECG noise levels and reconstruct missing data or detect anomalies. Finally, we demonstrate the effectiveness of our approach by comparing it to baseline methods and showcasing an innovative application: generating expected ECGs when a patient's heart rate increases. This application offers a promising alternative to the exercise stress test. Our contributions are the following.

- We introduce a flexible method that addresses multiple challenges in ECG analysis, including generating synthetic signals, noise reduction, missing data reconstruction, and anomaly detection

without having to re-train a model for each task.

- We adapt recent techniques for solving inverse problems with diffusion priors by including an estimation of unknown measurement noise levels.
- Our methods surpasses recent and classic existing approaches on multiple evaluation metrics specifically designed for ECGs, and offers novel applications.

6.2 Related Work

The use of generative models (Kingma et al., 2019; Kobyzev et al., 2020; Gui et al., 2021) as informative priors in solving Bayesian inverse problems has attracted significant interest Arjomand Bigdeli et al. (2017); Wei et al. (2022); Su et al. (2022); Kaltenbach et al. (2023); Shin and Choi (2023); Zhihang et al. (2023); Sahlström and Tarvainen (2023). In particular, DDMs have been demonstrated as a particularly suitable choice of prior for solving inverse problems Song et al. (2021a); Chung et al. (2023); Song et al. (2022); Kawar et al. (2022, 2021). DDMs are generative models that transform a simple reference distribution into the training data distribution through a denoising process called diffusion. These models are capable of generating high-quality realistic samples on par with the best Generative Adversarial Networks (GANs) Goodfellow et al. (2014) in terms of image and audio generation, without the intricacies of adversarial training (Sohl-Dickstein et al., 2015; Song et al., 2021c,a,b; Benton et al., 2022). In this article, we follow the approach proposed in Cardoso et al. (2023b); Wu et al. (2023), for sampling solutions to an inverse problem using a Sequential Monte Carlo (SMC) algorithm that guides the denoising process of a pretrained diffusion model. This method is accompanied by a series of theoretical guarantees in realistic scenarios. Generative modeling, denoising methods, and automatic anomaly detection algorithms are commonly used for ECG analysis. In particular, DDMs have been demonstrated to be capable of generating realistic ECGs: Adib et al. (2023) focuses on generating a single healthy beat for a single ECG lead, Alcaraz and Strodthoff (2023) generates a 10-second period conditioned on various complementary ECG information. Additionally, numerous methods address the denoising problem in ECGs Singh and Pradhan (2020); Li et al. (2023); Chiang et al. (2019). Classical approaches like Dower matrices Macfarlane et al. (2010) are used to reconstruct missing leads in ECGs. (Wen and Kang, 2021; Kang and Wen, 2022) rely on neural networks to detect anomalies, and Shan et al. (2022) use adversarial autoencoders for unsupervised anomaly detection. However, to our knowledge, there is no method that addresses all these problems with a single pretrained model.

6.3 Background

For all the ECG reconstruction tasks presented in our work, we used the same pre-trained DDM (section 6.3.1) as a prior for sampling solutions of these tasks with Monte Carlo guided diffusion (section 6.3.2).

6.3.1 Denoising Diffusion Generative Models (DDM):

We focus on the variance-exploding (VE) framework Song et al. (2021c), which transforms a reference distribution of the form $\lambda = \mathcal{N}(0, \upsilon_{\max}^2 \mathrm{I})$, with $\upsilon_{\max}^2 \gg 0$, into the data distribution. The training procedure involves denoising data that has been corrupted through a forward process as follow. The initial data state x_0 is sampled from $\mathsf{q}_{\mathrm{data}}$; independent noise with increasing variance is incrementally added to generate subsequent states $x_k = x_{k-1} + \rho_k \varepsilon_k$, where $k \in \mathbb{N}^*$, $\rho_k > 0$, and $\varepsilon_k \sim \mathcal{N}(0, \mathrm{I})$. The joint p.d.f. of the Markov chain is

$$\mathsf{q}_{0:K}(x_{0:K}) = \mathsf{q}_{\mathrm{data}}(x_0) \prod_{k=1}^{K} q_k(x_k | x_{k-1}) \,, \tag{6.1}$$

where $q_k(\cdot|x_{k-1}) = \mathcal{N}(x_{k-1}, \rho_k^2 \, \mathrm{I})$ and $K \in \mathbb{N}^*$. Hence, the conditional law at step k given x_s with $k > s \geq 0$ is

$$\mathsf{q}_{k|s}(\cdot|x_s) = \mathcal{N}(x_s, (\upsilon_k^2 - \upsilon_s^2) \, \mathrm{I}) \, , \tag{6.2}$$

with $\upsilon_k^2 = \sum_{j=1}^k \rho_j^2$ (and $\upsilon_0^2 = 0$). It is easy to see that for $K \in \mathbb{N}^*$ such that $\upsilon_K^2 = \upsilon_{\max}^2$ and $\upsilon_K^2 \gg \|x_0\|_\infty^2$, then $\mathsf{q}_{K|0}(\cdot|x_0)$ is close to the reference distribution $\lambda = \mathcal{N}(0, \upsilon_{\max}^2 \, \mathrm{I})$.

To infer real data from corrupted data, we introduce the *inference distribution* q^η, depending on hyper-parameter $\eta = \{\eta_k\}_{k \in \mathbb{N}}$ verifying for all $k \in \mathbb{N}$, $\eta_k^2 \leq \upsilon_k^2$. The p.d.f. of $x_{1:K}$ given the initial state x_0 is $q^\eta_{1:K|0}(x_{1:K}|x_0) := q^\eta_{K|0}(x_K|x_0) \prod_{k=K}^2 q^\eta_{k-1|k,0}(x_{k-1}|x_k, x_0)$ where

$$\begin{aligned}
q^\eta_{K|0}(\cdot|x_0) &:= \mathcal{N}(x_0, \upsilon_{\max}^2 \mathrm{I}) \approx \lambda, \\
q^\eta_{k-1|k,0}(\cdot|x_k, x_0) &:= \mathcal{N}\left(\boldsymbol{\mu}_{k-1}(x_k, x_0), \eta_{k-1}^2 \, \mathrm{I}_d\right), \\
\boldsymbol{\mu}_{k-1}(x_k, x_0) &:= x_0 + \sqrt{\upsilon_{k-1}^2/\upsilon_k^2 - \eta_{k-1}^2/\upsilon_k^2}(x_k - x_0),
\end{aligned}$$

for $k \in [1 : K]$; the backward induction is formulated as

$$q^\eta_{k-1|0}(\cdot|x_0) := \int q^\eta_{k-1|k,0}(\cdot|x, x_0) q^\eta_{k|0}(x|x_0) \mathrm{d}x \, .$$

In Lemma 70 we demonstrate that for $k \in [0 : K]$, $q^\eta_{k|0}(\cdot|x_0) = \mathsf{q}_{k|0}(\cdot|x_0)$. Since the states x_0 are not yet accessible as they are the ones we aim to model, all occurrences of x_0 are replaced by a denoised version of x_k, obtained with the subsequent network. Each corrupted state x_k is denoised, using the model $\mathcal{D}^\theta_{0|k}$ with parameters θ trained to minimize

$$\sum_{k=1}^K \gamma_k^2 \mathbb{E}_{\substack{X_0 \sim \mathsf{q}_{\mathrm{data}} \\ \epsilon \sim \mathcal{N}(0, \mathrm{I})}} \left[\|\mathcal{D}^\theta_{0|k}(X_0 + \upsilon_k \epsilon, \upsilon_k) - X_0\|^2 \right] \, , \tag{6.3}$$

where $\{\gamma_k\}_{k \in [1:K]}$ is a sequence of weighted coefficients.

After training, to generate $x_0 \sim \mathsf{q}_{\mathrm{data}}$ we start by sampling $x_K \sim \lambda$ and for $k = K$ to $k = 2$ we sample x_{k-1} given x_k with

$$p_{k-1|k}(\cdot|x_k) := q^\eta_{k-1|k,0}(\cdot|x_k, \mathcal{D}^\theta_{0|k}(x_k, \upsilon_k)) \, . \tag{6.4}$$

Finally, $x_0 \sim p_0(\cdot|x_1) := \mathcal{N}(\mathcal{D}^\theta_{0|1}(x_1, \upsilon_1), \eta_0^2 \, \mathrm{I})$. The p.d.f. of the sampled backward chain $x_{0:K}$ is

$$\mathsf{p}_{0:K}(x_{0:K}) := \lambda(x_K) \prod_{k=K}^1 p_{k-1|k}(x_{k-1}|x_k) \, .$$

This is equivalent to minimizing the Kullback-Leibler divergence between $\mathsf{q}_{\mathrm{data}}(x_0) q^\eta_{1:K|0}(x_{1:K}|x_0)$ and the joint backward $\mathsf{p}_{0:K}(x_{0:K})$, for a specific choice of $\{\gamma_k\}_{k \in [1:K]}$; see Lemma 71. For $k \in [0 : K-1]$, the marginal law of x_k is expressed with

$$\mathsf{p}_k(x_k) := \int \lambda(x_K) \prod_{s=K}^{k+1} p_{s-1|s}(x_{s-1}|x_s) \mathrm{d}x_{k+1:K}. \tag{6.5}$$

6.3.2 Monte Carlo Guided Diffusion

In Bayesian inverse problem, we aim to sample

$$\phi_0(x_0) := g_0(x_0) \mathsf{p}_0(x_0) / \mathcal{Z} \tag{6.6}$$

where g_0 is the likelihood function (typically depending on an observation) and $\mathcal{Z} := \int g_0(x) \mathsf{p}_0(x) \mathrm{d}x$ is the normalizing constant. The distribution ϕ_0 is often intractable, except for simple choices of g_0 and

p_0. A simple idea for sampling the posterior ϕ_0 is to use sampling importance resampling (SIR) Rubin (1987a), where p_0 is used as the instrumental distribution. However, this method may be inefficient since it neglects the potential g_0. It is imperative to construct an instrumental distribution that takes into account the likelihood function. For a given sequence, $x_{0:K}$, we define a distribution over the path space, using (6.5) and (6.6)

$$\phi_{0:K}(x_{0:K}) := \frac{g_0(x_0)}{\mathcal{Z}} \prod_{k=1}^{K} p_{k-1|k}(x_{k-1}|x_k)\lambda(x_K). \tag{6.7}$$

We introduce a sequence of potentials $\{g_k\}_{k\in[1:K]}$ with $g_K \equiv 1$, which aim is to lead the backward diffusion to regions of high values of g_0 is large. The path space distributions may be equivalently rewritten as

$$\begin{aligned}\phi_{0:K}(x_{0:K}) &\propto \lambda(x_K) \prod_{k=1}^{K} \frac{g_{k-1}(x_{k-1})p_{k-1|k}(x_{k-1}|x_k)}{g_k(x_k)} \\ &\propto \lambda(x_K) \prod_{k=1}^{K} \omega_k(x_k)p_{k-1|k}(x_{k-1}|x_k)\,,\end{aligned}$$

where, for $k \in [1:K]$, are defined

$$\begin{aligned}p_{k-1|k}(\cdot|x_k) &:= g^{k-1}(\cdot)p_{k-1|k}(\cdot|x_k)/\mathcal{Z}_k(x_k)\,, \\ \mathcal{Z}_k(x_k) &:= \int g^{k-1}(x')p_{k-1|k}(x'|x_k)\mathrm{d}x', \\ \omega_k(x_k) &:= \mathcal{Z}_k(x_k)/g_k(x_k)\,.\end{aligned} \tag{6.8}$$

We implicitly assume that these formulas have a closed form. By construction, for each $k \in [1:K]$ the marginal distribution of $\phi_{0:K}$ verifies

$$\begin{aligned}\phi_{k-1}(x_{k-1}) &\propto g_{k-1}(x_{k-1})\mathsf{p}_{k-1}(x_{k-1}) \\ &\propto \int \omega_k(x)p_{k-1|k}(x_{k-1}|x)\phi_k(x)\mathrm{d}x\,.\end{aligned} \tag{6.9}$$

Each ϕ_{k-1} thus has the same structure as ϕ_0: a product of a potential function and the marginal law at time $k-1$ of the backward diffusion. The original problem is replaced by a series of easier to solve problems.

It remains to approximate this sequence of distributions. For this purpose, we use Sequential Monte Carlo (SMC) Doucet et al. (2001); Chopin and Papaspiliopoulos (2020) to recursively build an empirical approximation from $k = K$ to $k = 0$. Suppose that we have at iteration k a *particle approximation* $\phi_k^M = M^{-1}\sum_{j=1}^{M} \delta_{\xi_k^j}$ of ϕ_k through a set of $M \in \mathbb{N}_{>0}$ particles $\xi_k^{1:M}$ Chopin and Papaspiliopoulos (2020), initialized with $\xi_K^{1:M} \sim \lambda^{\times M}$. Plugging this approximation into eq. (6.9) gives

$$\phi_{k-1} \propto \sum_{j=1}^{M} \omega_k(\xi_k^j)p_{k-1|k}(\cdot|\xi_k^j)\,. \tag{6.10}$$

Hence, to obtain $\xi_{k-1}^{1:M}$, we first sample M ancestors according to $I_{k-1}^{1:M} \sim \mathrm{Cat}\big(\{\omega_k(\xi_k^j)/\sum_{i=1}^{M}\omega_k(\xi_k^i)\}_{j=1}^{M}\big)^{\times M}$, then we sample new particles $\xi_{k-1}^{1:M} \sim \{p_{k-1|k}(\cdot|\xi_k^{I_{k-1}^j})\}_{j=1}^{M}$, leading to $\phi_{k-1}^M = M^{-1}\sum_{j=1}^{M}\delta_{\xi_{k-1}^j}$. Cf. algorithm 12 in section E.5.

6.4 Methods

Many fundamental problems in ECG analysis, such as noise suppression, reconstruction of missing leads, T-wave prediction, and anomaly detection, can be formulated as ill-posed linear inverse problems. For the sake of simplicity, we focus in this section on the problem of recovering/denoising the ECG signal in the presence of noise and/or missing samples. We discuss how section 6.3.2 can be employed to sample ECGs from partial observations using a pre-trained DDM as a prior. We also introduce the inference procedure for estimating the unknown level of noise in the observation.

6.4.1 ECG Linear Inverse Problem

ECGs are $L \times T$ matrices where L is the number of leads, and T is the number of samples. We assume that we have trained a DDM on ECG data and have access to the backward process: to generate a new ECG $x_0 \in \mathbb{R}^{L \times T}$, we first sample x_K from λ, then for k from K to 1, we sample $x_{k-1} \sim p_{k-1|k}(x_{k-1}|x_k)$ ((6.4)), as illustrated in figure 6.1.

We assume that we partially measure a new ECG through a subset of indices $\mathcal{I} = \{(\ell, t) \in [1 : L] \times [1 : T]\}$. For any $(\ell, t) \in \mathcal{I}$, the observation is written as

$$Y[\ell, t] = X_0[\ell, t] + \sigma_\ell \epsilon_{\ell,t}\,, \tag{6.11}$$

where $\epsilon_{\ell,t} \sim \mathcal{N}(0, 1)$, and $\sigma = \sigma_{1:L}$ are the measurement noise variances; we first assume that the variances are known; we describe below a method to estimate these parameters. Given an observation $y \sim Y$, we aim to sample x_0 from the posterior $X_0|y, \sigma$, with a p.d.f.

$$\phi_0^y(x_0) := g_0^y(x_0)\mathsf{p}_0(x_0)/\mathcal{Z} \tag{6.12}$$

where $\mathsf{p}_0(x_0)$ is the prior distribution defined in (6.5), $\mathcal{Z} = \int g_0^y(x)\mathsf{p}_0(x)\mathrm{d}x$ is the normalizing constant, and $g_0^y(x_0)$ is the likelihood of the observation, given by

$$g_0^y(x_0) := \textstyle\prod_{(\ell,t)\in\mathcal{I}} \mathcal{N}(x_0[\ell, t]; y[\ell, t], \sigma_\ell^2)\,.$$

We use the methods described in section 6.3.2 and adapt the choice of potentials $\{g_k^y\}_{k\in[0:K]}$ derived in Cardoso et al. (2023b) for the VE framework

$$g_k^y(x) = \prod_{(\ell,t)\in\mathcal{V}_k} \mathcal{N}(x[\ell, t]; y[\ell, t], \upsilon_k^2 - (1-\varepsilon)\sigma_\ell^2), \tag{6.13}$$

where $\mathcal{V}_k = \{(\ell, t) \in \mathcal{I} | \upsilon_k^2 \geq \sigma_\ell^2\}$ and ε is a positive hyper-parameter (see section E.6 for a heuristic introducing this choice of potential). By convention, if $\mathcal{V}_k = \emptyset$, we set $g_k^y(x) \equiv 1$. For this potential, $p_{k-1|k}^y$ and ω_k^y admit closed forms given in section E.7.

6.4.2 Estimation of Measurement Noise

We now discuss the estimation of the noise variance. We propose to use the MLE of σ, $\sigma^* = \mathrm{argmax}_{\sigma\in\mathbb{R}^{\mathrm{S}_y}} l(\sigma)$ where $l(\sigma) := \log \mathcal{Z}^{,\sigma} = \log \int g_0^{y,\sigma}(x)\mathsf{p}_0(x)\mathrm{d}x$. Note that we have explicitly specified the dependence of the potential $g_0^{y,\sigma}(x)$ on the noise variance. The gradient of l is approximated using $\xi_0^{1:M}$ obtained with algorithm 12

$$\begin{aligned}\nabla_\sigma l(\sigma) &= \textstyle\int \nabla_\sigma g_0^{y,\sigma}(x)\mathsf{p}_0(x)/\mathcal{Z}^{,\sigma}\mathrm{d}x \\ &= \textstyle\int \nabla_\sigma \log g_0^{y,\sigma}(x)\phi_0^{y,\sigma}(x)\mathrm{d}x \\ &\approx \textstyle M^{-1}\sum_{j=1}^M \nabla_\sigma \log g_0^{y,\sigma}(\xi_0^j)\,.\end{aligned}$$

We obtain the estimator σ through gradient ascent Cappé et al. (2005b)[Section 11] and enhance its robustness by running N_c parallel instances of algorithm 12 and averaging the resulting estimators, as outlined in algorithm 5.

6.5 Experiments

Our code to reproduce all experiment is available.[1]

[1] Anonymous code available at https://anonymous.4open.science/r/ecg_inpainting-7457

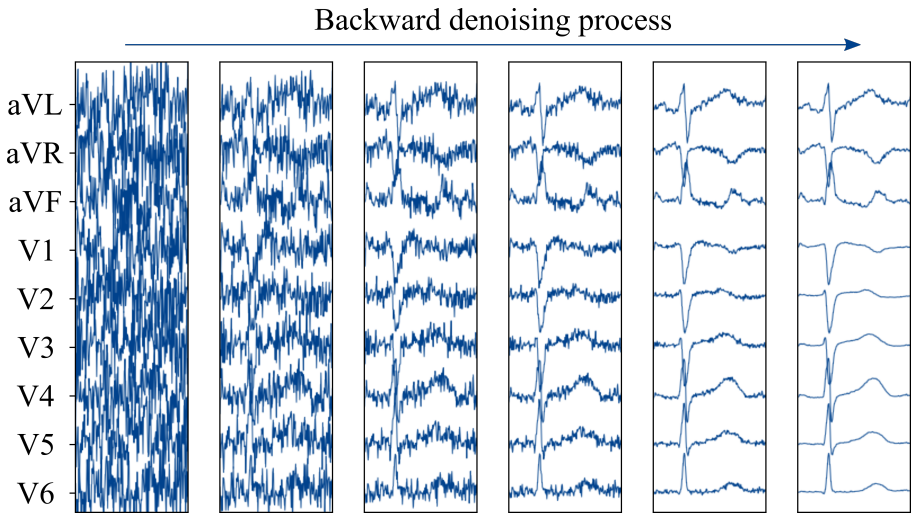


Figure 6.1: Example of healthy heartbeat generated with a denoising diffusion generative model, across multiple diffusion steps.

Algorithm 5 `NC-MCGdiff`

Input: number of steps N_{MLE}, initialization σ^0, number of parallel chains N_c
Parameters for SMC: observation y, number of diffusion steps K, number of particles M
for $i = 0$ **to** $N_{\text{MLE}} - 1$ **do**
  Sample $\xi_{0,i}^{1:N_c M}$ by running N_c parallel SMC with σ^i, y, K, M
  Update $\sigma^{i+1} := \sigma^i + \frac{\gamma}{(i+1)^{0.6}} \widehat{\nabla_\sigma l(\sigma^i)}[\xi_{0,i}^{1:N_c M}]$
end for
Output: $\sigma^{N_{\text{MLE}}}$

6.5.1 Dataset and Preprocessing

We utilize the PhysioNet Challenge dataset Goldberger et al. (e 13); Reyna et al. (2021, 2022), comprising 43,101 12-lead ECGs. Our preprocessing involves four steps, as described in section E.2: normalization of the sampling frequency to 250 Hz (resulting in time points separated by 4 ms), detection of R peaks to identify heartbeats, segmentation of the heartbeats within the window $[\text{R} - 192 \text{ ms}, \text{R} + 512 \text{ ms}]$, and amplitude normalization. This process generates 214,460 single-beat ECGs, each with a time length of 704 ms and leads (aVL, aVR, aVF, V1–V6), represented as an $L \times T$ matrix, where $L = 9$ and $T = 176$ (since $704 \text{ ms}/4 \text{ ms} = 176$), from a pool of 28,167 individuals with healthy profiles and 468 patients diagnosed with myocardial infarction (MI). Due to significant variability between patients compared to variability between heartbeats, we randomly select a single beat per patient from either the training, cross-validation (CV), test, or MI datasets for model evaluation. All analyses are conducted on single beats with normalized amplitudes, but our entire approach is also applicable to signals with multiple beats with non-normalized amplitudes.

6.5.2 Denoising Network for ECGs

Our ECG denoising model is based on two key insights: first, generative models perform better when additional information such as labels is incorporated during generation. Therefore, in addition to the ECG and noise level, we input time $\mathcal{T}$ and categorical patient data $\mathcal{P}$ into the network. The second insight is that the noise level varies with the diffusion step k, and we use the following reparameterization

(Karras et al., 2022)

$$\mathcal{D}^{\theta}_{0|k}(x, \upsilon_k, \mathcal{T}, \mathcal{P}) = \mathrm{c}_{\mathrm{skip}}(\upsilon_k)x + \mathrm{c}_{\mathrm{out}}(\upsilon_k)\,\mathrm{F}_{\theta}(x, \upsilon_k, \mathcal{T}, \mathcal{P})\,.$$

where x is a 9×176 matrix, $\mathrm{c}_{\mathrm{skip}}(\upsilon_k) = (\upsilon_k^2 + \sigma^2_{\mathrm{data}})^{-1}\sigma^2_{\mathrm{data}}$, $\mathrm{c}_{\mathrm{out}}(\upsilon_k) = \upsilon_k\sigma_{\mathrm{data}}(\upsilon_k^2 + \sigma^2_{\mathrm{data}})^{-1/2}$, and σ_{data} is the empirical standard deviation of $\mathsf{q}_{\mathrm{data}}$. For small υ_k, $\mathrm{c}_{\mathrm{skip}}(\upsilon_k) \approx 1$ and $\mathrm{c}_{\mathrm{out}}(\upsilon_k) \approx 0$, thus $\mathcal{D}^{\theta}_{0|k}(x, \upsilon_k, \mathcal{T}, \mathcal{P}) \approx x$, which is expected since x is already a good reconstruction of the original data. On the contrary, when υ_k is large, then $\mathrm{c}_{\mathrm{skip}}(\upsilon_k) \approx 0$ and $\mathrm{c}_{\mathrm{out}} \approx 1$, thus $\mathcal{D}^{\theta}_{0|k}(x, \upsilon_k, \mathcal{T}, \mathcal{P})$ relies heavily on the network F_{θ} to provide a good reconstruction.

The initial layers of F_{θ} aggregate the corrupted state x, the standard deviation of the exploration noise υ_k, the temporal information $\mathcal{T}$, and the categorical patient information $\mathcal{P}$ into a single matrix $\mathrm{e}_x + \mathrm{e}_{\upsilon_k} + \mathrm{e}_{\mathcal{T}} + \mathrm{e}_{\mathcal{P}}$. We now discuss how each component is encoded. First, to mitigate the impact of magnitude variability across different diffusion steps, x is rescaled by the normalization factor $\mathrm{c}_{\mathrm{in}}(\upsilon_k) = (\upsilon_k^2 + \sigma^2_{\mathrm{data}})^{-1/2}$. Subsequently, $\mathrm{c}_{\mathrm{in}}(\upsilon_k)x$ is fed into a 1D convolutional layer with a 1-size kernel and c channels, resulting in a $c \times 176$ matrix e_x. To incorporate the information of the noise and the time, we use positional encoding Vaswani et al. (2017) defined for $s, t \in [1 : c] \times [1 : T]$ as

$$\mathrm{Enc}(t)[s] = \begin{cases} \sin(1000^{-(r/96)}t) & \text{if} \quad \ell = 2r\,, \\ \cos(1000^{-(r/96)}t) & \text{if} \quad \ell = 2r + 1\,. \end{cases}$$

The noise operator is defined for $s \in [1 : c]$ as $\mathrm{e}_{\upsilon_k}[s] = \mathrm{Enc}(\frac{\log(\upsilon_k)}{4})[s]$, and the time operator is defined for $s, t \in [1 : c] \times [1 : T]$ as $\mathrm{e}_{\upsilon_k}[s] = \mathrm{Enc}(t)[s]$. Various factors, including age (A), sex (S), and the preceding R-R interval (RR), which is linked to the inverse of the heart-rate, affect the morphology Malik et al. (2013); Salama and Bett (2014); Ball et al. (2014). We normalize A and RR as $\tilde{A} = (A - 50)/50$ and $\tilde{\mathrm{RR}} = (\mathrm{RR} - 400)/400$. A one-hot encoding is applied to S to generate $\tilde{S} \in \{0, 1\}^2$. The concatenated vector $\tilde{S}, \tilde{A}, \tilde{\mathrm{RR}}$ is fed into a two-layer dense network, yielding a $c \times 1$ vector $\mathrm{e}_{\mathcal{P}}$.

After aggregating ECG, distortion, temporality, and patient information, F_{θ} adopts a U-Net architecture Ronneberger et al. (2015); Ho et al. (2020); Dhariwal and Nichol (2021). Each output from the U-Net blocks undergoes a multi-head attention layer Vaswani et al. (2017), with the number of heads equal to the original dimension divided by 64. The entire network $\mathcal{D}^{\theta}_{0|k}$ is trained to minimize eq. (6.3) through stochastic gradient descent on the healthy training set, and the best model is selected using the cross-validation set. For further details, refer to section E.3.

6.5.3 Evaluation of ECG Generation

We first evaluate the quality of the generated ECGs with the DDM trained as described in the previous section. To do so, we generate the same number of ECGs as in the test set (2864) using the same 2864 features $\mathcal{P} = (A, S, \mathrm{RR})$. We propose two metrics to assess the quality of the synthetically generated ECGs. These are: (1) a distance between the real and the generated ECG distribution, and (2) an out-of-distribution (OOD) score quantifying how likely a given ECG is outside the training healthy distribution. The Earth Mover's Distance (EMD) Genevay et al. (2016) measures the dissimilarity between the predicted and target distributions by calculating the minimal transport cost. The EMD is calculated from the generated set to both the test set and the training set. To obtain comparable orders of magnitude, the training set is divided into batches of the same size (2864). The transport cost is defined as the L^2-distance over concatenated ECGs with A, S, RR features to penalize the transport of an ECG to ECGs with different A, S, and RR features. With this metric, we compare the DDM with the WGAN model proposed in Adib et al. (2022). To ensure both models are comparable, we use the same training set for both models: heartbeats conditioned with $\mathcal{P} = (A, S, \mathrm{RR})$. As the WGAN model was originally

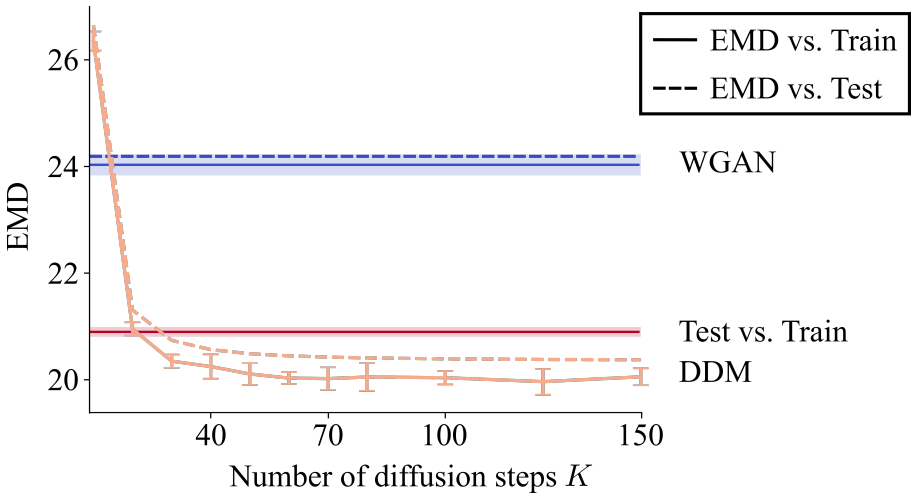


Figure 6.2: EMD between generated ECG distribution and real ECG distribution. EMD vs. test (resp. train) in dotted (resp. plain) line. EMD for DDM with different number of difffusion steps, in orange. DDM for WGAN model in blue. EMD between test and train distributions in red. Error bars correspond to different training batches of size 2864.

introduced for categorical conditioning, we adapted it to include scalar conditioning (RR) using two fully connected layers as detailed in section E.9. We also use the EMD to assess the influence of the number of diffusion steps K. Figure 6.2 shows the EMD with respect to the test and training sets for both the WGAN and DDM, with K varying in the interval $[2, 150]$. The EMD values show that few diffusion steps are sufficient to generate an accurate predictive distribution, and the DDM outperforms the WGAN in reproducing the real data distribution. The analysis in Section E.4 shows that using a more complex architecture does not improve the results, and conditioning on A, S, RR leads to a smaller EMD.

To quantify how unlikely each generated ECG is with respect to the training distribution, we used the OOD-score proposed by Ciosek et al. (2020). Their method involves using a randomly initialized network, which remains unchanged throughout the process, to produce a "random prior" by associating each training data point (images in the original paper, real or generated ECGs in our case) with a random pattern. Subsequently, a second network is trained to learn this random prior distribution, meaning that the output of the network for a training data point should be close (in terms of L2 distance) to the random pattern from the first network. After training the second network, the OOD-score for an input data point is the distance between the outputs of the two networks. The authors demonstrate the relevance of their score for out-of-distribution data detection by training on four classes of the CIFAR dataset and verifying that, at test time, the score effectively distinguishes test data with the same classes as the training data from those with different classes. In our case, we adopt the same residual network architectures proposed in Ciosek et al. (2020), but replace the 2D convolutions with 1D convolutions, as unidimensional residual networks are known for their efficiency in ECG classification Ribeiro et al. (2020). We use 10 bootstraps and train the corresponding networks for 100 epochs with the Adam optimizer (learning rate=0.001) on healthy patients from the training set. The OOD-score boxplots and the resulting classification ROC curve in figure 6.3 show that the OOD-scores of the generated ECGs are close to those of the test ECGs, and that the scores for MI ECGs are significantly higher than those for the test and generated ECG.

6.5.4 ECG Denoising

We now consider the application of `NC-MCGdiff` to solve various problems. In all our experiments, we do not perform additional fine-tuning; all our results are obtained solely by sampling the pre-trained

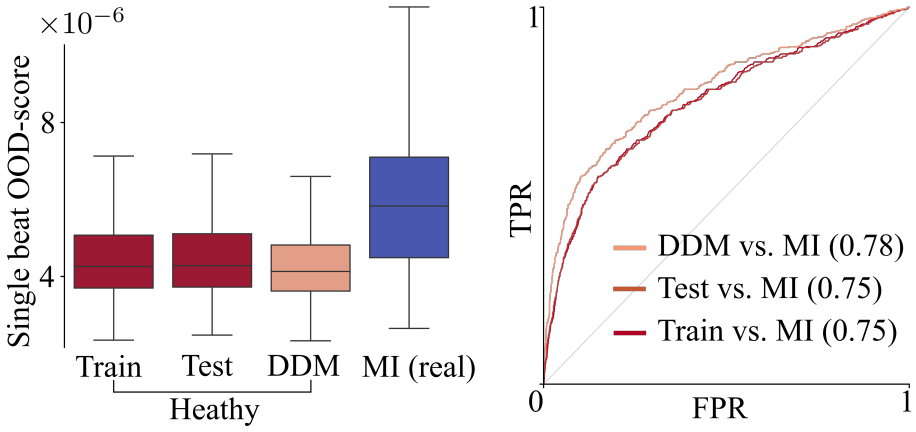


Figure 6.3: Out-of-distribution evaluation. ***Left.*** *Box-plot of OOD-score for train, test, generated (Gen) and MI heart beats.* ***Right.*** *ROC curves for classification between train/test/gen and MI based on OOD-score.*

DDM model as follow: for a given observation y we perform algorithm 5 with $N_{\text{MLE}} = 10$ (stops before when convergence is reached), $\sigma_0 = 1$, $N_c = 100$, $K = 50$, $M = 50$ (see section E.8) to estimate σ^*. Then we generate 100 ECGs from $X_0|y, \sigma^*$ (conditioning on $\mathcal{P} = A, S, \text{RR}$ is implicit) by running $N_c = 100$ parallel SMC (algorithm 12 in section E.5) with σ^*, y, $K = 50$, $M = 50$ as input. We first investigate the denoising of noisy ECG observations ($\mathcal{I} = [1 : L] \times [1 : T]$). In this experiment, all test samples are corrupted with per-lead Gaussian noise with standard deviation sampled from an exponential law $\sigma_\ell \sim \exp(0.2)$. The randomness of σ mimics real-world scenarios where some electrodes may be more corrupted than others. For each corrupted test ECG y, σ^* is estimated using algorithm 5, and $N_c = 100$ denoised samples are drawn from $X_0|y, \sigma^*$. We compare `NC-MCGdiff` with a Denoising Autoencoder (DAE) introduced for ECGs by Chiang et al. (2019), whose architecture we adapt to single heartbeats as described in section E.9. We trained the DAE to denoise ECGs corrupted with per-lead Gaussian noise with standard deviation sampled from an exponential law $\sigma_\ell \sim \exp(0.2)$, using the Adam optimizer. Figure 6.4 shows two examples of corrupted heartbeats denoised with `NC-MCGdiff` and the DAE. To assess the reconstruction quality, we measure the R^2-score between real and denoised ECGs. `NC-MCGdiff` outperforms the DAE with an R^2-score of 0.928 ± 0.002, whereas the DAE achieves a score of 0.855 ± 0.003. We also verify the accuracy of the estimated parameter σ^* by computing the absolute total deviation between the real $\sigma = \sigma_{1:S_y}$ and the derived σ^*, resulting in 0.03 ± 0.001.

6.5.5 Missing Leads Reconstruction

We evaluate `NC-MCGdiff` for reconstructing a missing lead ℓ while observing the other leads ($\mathcal{I} = [1 : \ell-1] \cup [\ell+1 : L] \times [1 : T]$). In this experiment, for each test ECG, a precordial lead $\ell \in [4 : L]$ is randomly removed; the reconstruction of missing leads aVL, aVR, or aVF is considered impractical because the absence of these leads implies the absence of a limb electrode, which prevents the measurement of any leads. Then, for each partial ECG y, complete beats are generated using `NC-MCGdiff` and with Dower matrices Macfarlane et al. (2010)[Chapter 11], which is a classical method notably used for missing lead reconstruction. Figure 6.5 shows three examples of reconstructed beats with both methods. To assess the quality of the reconstruction, we compute the R^2-score between the reconstructed and the ground-truth missing leads. The results in table E.2 show that our approach outperforms reconstruction with Dower matrices for all missing leads; the overall R^2-score is 0.987 ± 0.003 for `NC-MCGdiff` and 0.804 ± 0.023 for Dower matrices. This experiment opens up numerous possibilities for applications, such as managing errors in electrode placement and predicting complete ECGs from partial ECGs measured by devices such as the Apple Watch.

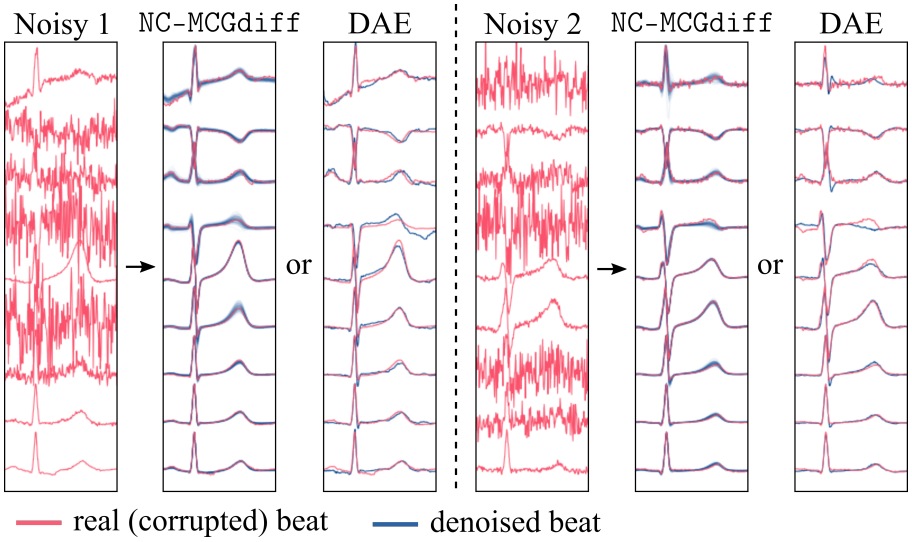


Figure 6.4: Denoising of two corrupted signals with Gaussian noise using `NC-MCGdiff` *and Denoising Auto-Encoder (DAE).*

Table 6.1: Comparative evaluation of `NC-MCGdiff` *against existing approaches for ECG generation (Gen.), denoising (Denois.), missing lead reconstruction (Recon.), anomaly detection (Anom.), with EMD for generated vs. test distribution, R^2 score between denoised (resp. reconstructed) and real ECG (resp. missing lead), AUC for anomaly score.*

TASK Metric	Gen. EMD	Denois. R^2	Recon. R^2	Anom. AUC
NC-MCGdiff	**21.26**	**0.928**	**0.987**	**0.83**
WGAN	24.16	-	-	-
DAE	-	0.855	-	0.72
Dower	-	-	0.804	-
AAE	-	0.685	-	0.81
OOD	-	-	-	0.75

6.5.6 Cardiac Anomaly Detection

In this section, we evaluate `NC-MCGdiff` for detecting cardiac abnormalities by addressing an inverse problem as follows. Given an ECG x that may exhibit morphological anomalies, we sample a new ECG $\hat{x}$ from the posterior $X_0|y, \sigma^*$, where y represents a partial observation of x with $\mathcal{I} = [4 : L] \times [1 : T]$. We condition on the augmented leads aVL, aVR, aVF since they are further from the heart and less likely to be affected by localized anomalies than the precordial leads V1-V6. The $1 - R^2$-score between $\hat{x}$ and x provides an anomaly score, and anomalies in the real ECG x can be highlighted by superimposing $\hat{x}$ on x. We applied our methodology to detect MI, as illustrated in figure 6.6. To evaluate the accuracy of our anomaly score, we compute the Area Under the Curve (AUC) for classifying control versus myocardial infarction (MI) based on the anomaly score as shown in figure 6.7. Our method performs better than a recent anomaly detection approach based on Adversarial AutoEncoder (AAE) Shan et al. (2022), achieving AUC values of 0.84 and 0.82 for females and males, respectively, compared to 0.78 and 0.81.

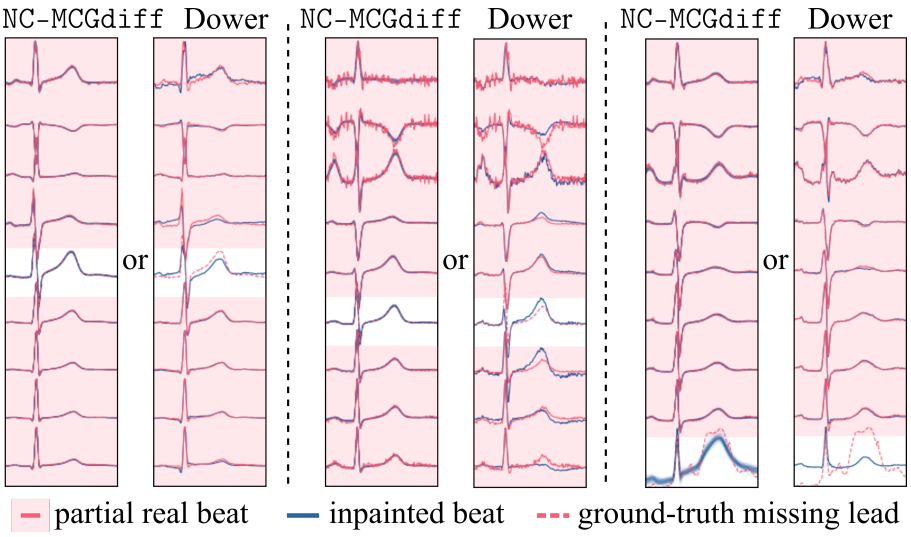


Figure 6.5: Reconstruction of 3 partially observed ECG using `NC-MCGdiff` *and Dower matrices.*

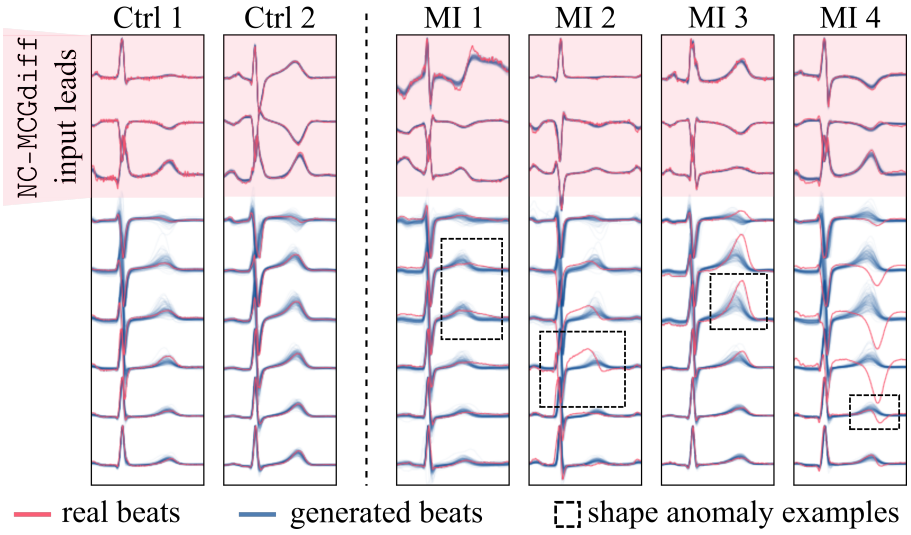


Figure 6.6: Illustration of ECG anomaly detection using `NC-MCGdiff` *in MI patients compared to control patients.*

6.5.7 Application: Prediction of Corrected QT

In the previous sections, we have demonstrated that for several classical ECG applications, `NC-MCGdiff` outperforms methods specifically designed for individual problems. To address these challenges, we pre-train a diffusion model once on a dataset of healthy ECGs and utilize it as a prior for all experiments. Table 6.1 provides a comparative summary with all the baselines mentioned in our experiments, also detailed in section E.9.

In this section, we introduce a new application that, to our knowledge, has not been numerically tackled before. The relationship between QT and heart rate (linked to the inverse of the RR) is well-documented in the medical literature and has been expressed in several formulas Bazett (1997); Fridericia (1921); Sagie et al. (1992). These formulas introduce coefficients called "corrected QT" denoted as QT_0^c and QT_1^c, which depend on the patient and are determined from ECGs measured during an exercise stress test. Using `NC-MCGdiff` we propose a numerical approach to avoid the need for an exercise stress test. Each test ECG is truncated to focus only on the QRS complex, i.e., we set $\mathcal{I} = [1 : L] \times [1 : 70]$. Then, for RR values ranging from 0.6 s to 1.2 s, or equivalently for heart rates ranging from 43 to 100 beats

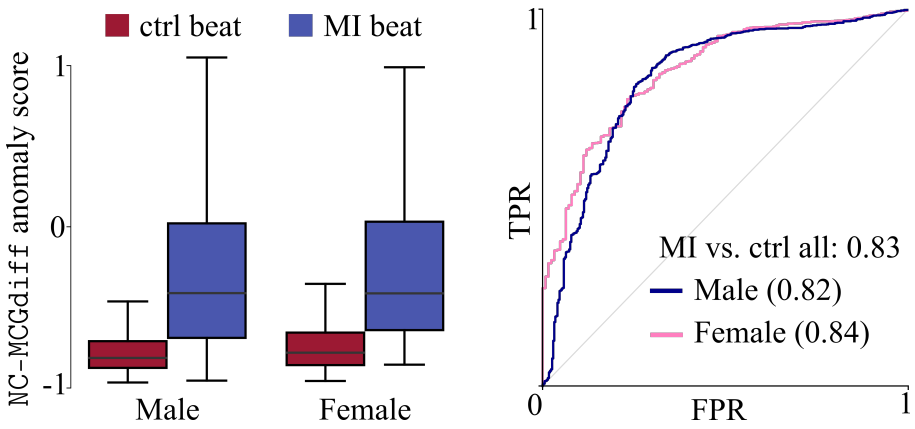


Figure 6.7: ***Left.*** *Distribution of* `NC-MCGdiff` *anomaly score for control (red) and MI (blue) ECGs.* ***Right.*** *ROC curve for classification between control and MI based on the anomaly score.*

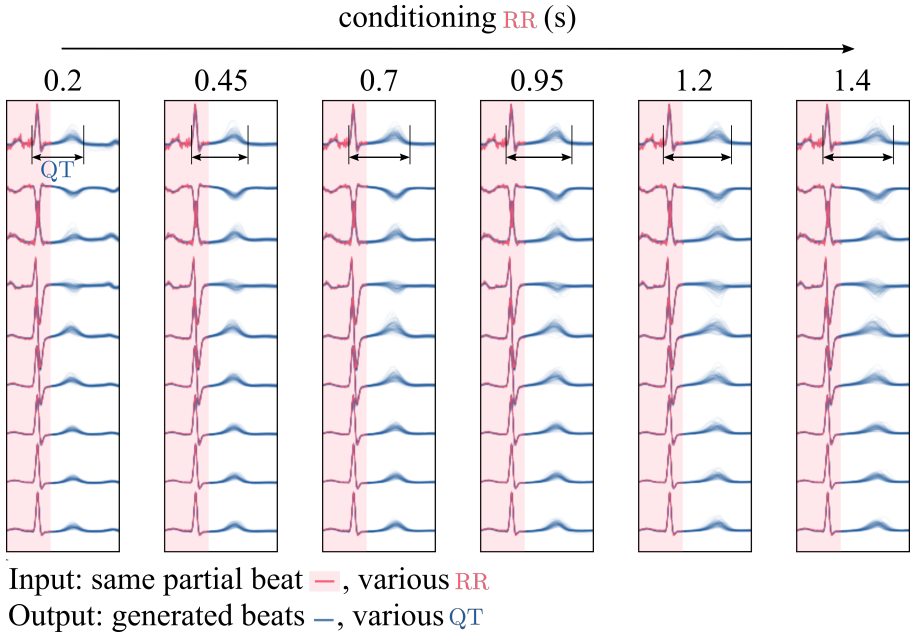


Figure 6.8: Example of T-wave prediction (blue) conditioned on Q-wave (red) for different value of RR.

per minute, we sample x from the conditional distribution $X_0|y, \sigma^*, \mathrm{RR}$ as illustrated in figure 6.8. We regress the intercept QT_0^c and slope QT_1^c of the Fridericia formula Fridericia (1921), which states that $\mathrm{QT} = \mathrm{QT}_0^c + \mathrm{QT}_1^c \sqrt[3]{\mathrm{RR}}$, from the generated curves. As shown in figure 6.9, we observe a consistent trend between the observed and regressed curves for five patients. Additionally, table E.3 indicates a high R^2-score of 0.98 between observed and expected QT curves.

This experiment illustrates the importance, when generating synthetic ECGs for a given patient, of conditioning on specific observations unique to that patient, such as their QRS complex and RR intervals, to capture their individual physiological differences compared to other patients. While the relationship between QT and RR has been observed in clinical settings, our model reproduces it without explicitly enforcing it during training or sampling. Furthermore, this experiment suggests that our model reliably predicts the T wave (ventricular repolarization) given the QRS (ventricular depolarization), opening up new applications such as the diagnosis of long QT syndrome or other diseases that specifically alter repolarization without altering the QRS.

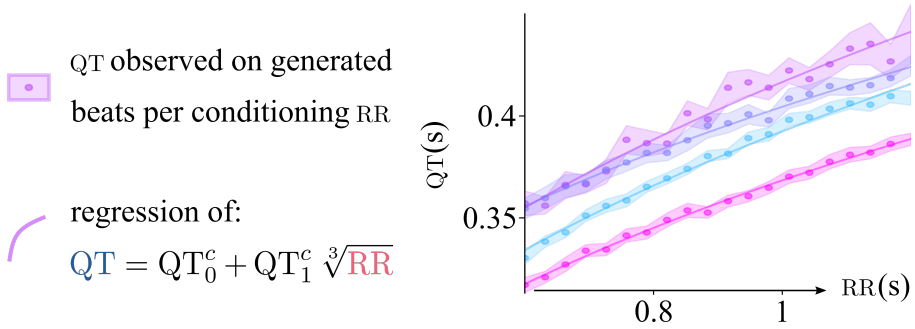


Figure 6.9: QT as a function of RR for 5 patients. QT measured in 100 generated samples (resp. regressed with Fridericia formula) displayed in dots with 95%-CLT bars (resp. curve).

6.6 Conclusion

In this paper, we described a flexible method that addresses various challenges in ECG analysis, including noise reduction, missing data reconstruction, and anomaly detection, all formulated as inverse linear problems. Our method leverages a DDM, pre-trained once to generate ECGs, as a prior for sampling solutions to inverse problems with SMC. We extended existing methods for solving inverse problems with a DDM prior, for cases with unknown measurement noise levels. The effectiveness of our approach is demonstrated against baselines through several evaluation metrics specifically designed for ECGs. Additionally, we introduced an innovative application of our method: generating expected ECGs when heart rate increases, offering an alternative to the exercise stress test. This contribution extends the utility of our approach beyond conventional ECG analysis tasks.

Besides, our approach opens up new applications such as completing ECGs measured by devices like the Apple Watch and diagnosing long QT syndrome or other diseases that specifically alter repolarization. In this paper, the DDM was trained only on healthy ECGs. Furthermore, this DDM could be replaced by a model trained on a dataset containing ECGs presenting pathologies, conditioned on the specific pathologies. A concrete application example would be training the model with ECGs from patients with left bundle branch block condition to detect ischemia in these patients for whom the criteria of ST segment elevation or depression are not valid.

6.7 Impact Statements

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Appendices

Appendix A

Appendix of Chapter 2

A.1 Proofs

A.1.1 i-SIR Algorithm

We analyze a slightly modified version of the i-SIR algorithm, with an extra randomization of the state position. The k-th iteration is defined as follows. Given a state $Y_k \in \mathbb{X}$,

(i) draw $I_{k+1} \in \{1, \dots, N\}$ uniformly at random and set $X_{k+1}^{I_{k+1}} = Y_k$;
(ii) draw $X_{k+1}^{1:N \setminus \{I_{k+1}\}}$ independently from the proposal distribution λ;
(iii) compute, for $i \in \{1, \dots, N\}$, the normalized importance weights

$$\omega_{N,k+1}^i = w(X_{k+1}^i) / \sum_{\ell=1}^N w(X_{k+1}^\ell);$$

(iv) select Y_{k+1} from the set $X_{k+1}^{1:N}$ by choosing X_{k+1}^i with probability $\omega_{N,k+1}^i$.

Thus, compared to the simplified i-SIR algorithm given in the introduction, the state is inserted uniformly at random into the list of candidates instead of being inserted at the first position. Of course, this change has no impact as long as we are interested in integrating functions that are permutation invariant with respect to candidates, which is the case throughout our work. Still, this randomization makes the analysis much more transparent.

A.1.2 Proof of Theorem 2

We write

$$\boldsymbol{\varphi}_N(\mathrm{d}(y, x_{1:N})) = \frac{1}{N} \sum_{i=1}^N \pi(\mathrm{d}y) \delta_y(\mathrm{d}x^i) \prod_{j \neq i} \lambda(\mathrm{d}x^j) \tag{A.1}$$

$$= \frac{1}{N\lambda(w)} \sum_{i=1}^N w(x^i) \lambda(\mathrm{d}x^i) \delta_{x^i}(\mathrm{d}y) \prod_{j \neq i} \lambda(\mathrm{d}x^j) \tag{A.2}$$

$$= \frac{1}{\lambda(w)} \prod_{j=1}^N \lambda(\mathrm{d}x^j) \Gamma_N \mathbb{1}_{\mathbb{X}}(x^{1:N}) \sum_{i=1}^N \frac{w(x^i)}{\sum_{\ell=1}^N w(x^\ell)} \delta_{x^i}(\mathrm{d}y), \tag{A.3}$$

where we recognize, and after having recalled definitions (2.5) and (2.6) of $\boldsymbol{\pi}_N$ and Π_N, respectively, the right-hand side as $\boldsymbol{\pi}_N(\mathrm{d}x^{1:N}) \Pi_N(x^{1:N}, \mathrm{d}y)$. This completes the proof.

A.1.3 Proof of Theorem 3

Using (2.6) we get

$$\int \boldsymbol{\pi}_N(\mathrm{d}x^{1:N})\Pi_N f(x^{1:N}) = \int \frac{1}{N\lambda(w)} \sum_{\ell=1}^{N} w(x^\ell)\Pi_N f(x^{1:N}) \prod_{j=1}^{N} \lambda(\mathrm{d}x^j) \tag{A.4}$$

$$= \frac{1}{N\lambda(w)} \int \sum_{i=1}^{N} w(x^i) f(x^i) \prod_{j=1}^{N} \lambda(\mathrm{d}x^j) = \pi(f), \tag{A.5}$$

and the proof is complete.

A.1.4 Proof of Theorem 6

Proof. We first check that $\boldsymbol{\varphi}_N$ is an invariant distribution for $\mathbf{P}_N$. For every $A \in \mathcal{X}^{\otimes(N+1)}$, using that π is the marginal of $\boldsymbol{\varphi}_N$ with respect to the state and applying Theorem 2 yields

$$\int \boldsymbol{\varphi}_N(\mathrm{d}(y, x^{1:N}))\mathbf{P}_N(y, x^{1:N}, A) = \int \pi(\mathrm{d}y) \iint \boldsymbol{\Lambda}_N(y, \mathrm{d}\bar{x}^{1:N})\Pi_N(\bar{x}^{1:N}, \mathrm{d}\bar{y})\mathbb{1}_A(\bar{y}, \bar{x}^{1:N}) \tag{A.6}$$

$$= \iiint \boldsymbol{\pi}_N(\mathrm{d}\bar{x}_{1:N})\Pi_N(\bar{x}^{1:N}, \mathrm{d}y)\Pi_N(\bar{x}^{1:N}, \mathrm{d}\bar{y})\mathbb{1}_A(\bar{y}, \bar{x}^{1:N}) \tag{A.7}$$

$$= \boldsymbol{\varphi}_N(A), \tag{A.8}$$

which establishes invariance. We now show that P_N is reversible with respect to π. For this purpose, let g and h be two nonnegative measurable functions and write, using Theorem 2 twice,

$$\iint \pi(\mathrm{d}y)\mathsf{P}_N(y, \mathrm{d}\bar{y})g(y)h(\bar{y}) = \int \pi(\mathrm{d}y)\boldsymbol{\Lambda}_N(y, \mathrm{d}x^{1:N})\Pi_N(x^{1:N}, \mathrm{d}\bar{y})g(y)h(\bar{y}) \tag{A.9}$$

$$= \int \boldsymbol{\pi}_N(\mathrm{d}x^{1:N})\Pi_N(x^{1:N}, \mathrm{d}y)\Pi_N(x^{1:N}, \mathrm{d}\bar{y})g(y)h(\bar{y}) \tag{A.10}$$

$$= \int \pi(\mathrm{d}\bar{y})\boldsymbol{\Lambda}_N(\bar{y}, \mathrm{d}x^{1:N})\Pi_N(x^{1:N}, \mathrm{d}y)g(y)h(\bar{y}) \tag{A.11}$$

$$= \iint \pi(\mathrm{d}\bar{y})\mathsf{P}_N(\bar{y}, \mathrm{d}y)g(y)h(\bar{y}). \tag{A.12}$$

□

A.1.5 Proof of Theorem 7

For completeness, we repeat the arguments in Lindsten et al. (2015); Andrieu et al. (2018). Under **A**1, we have, for $(x, \mathsf{A}) \in \mathbb{X} \times \mathcal{X}$,

$$\begin{aligned}
\mathsf{P}_N(x, \mathsf{A}) &= \int \delta_x(\mathrm{d}x^1) \sum_{i=1}^{N} \frac{w(x^i)}{\sum_{j=1}^{N} w(x^j)} \mathbb{1}_{\mathsf{A}}(x^i) \prod_{j=2}^{N} \lambda(\mathrm{d}x^j) \\
&= \int \frac{w(x)}{w(x) + \sum_{j=2}^{N} w(x^j)} \mathbb{1}_{\mathsf{A}}(x) \prod_{j=2}^{N} \lambda(\mathrm{d}x^j) + \int \sum_{i=2}^{N} \frac{w(x^i)}{w(x) + \sum_{j=2}^{N} w(x^j)} \mathbb{1}_{\mathsf{A}}(x^i) \prod_{j=2}^{N} \lambda(\mathrm{d}x^j) \\
&\geq \sum_{i=2}^{N} \int \frac{w(x^i)}{w(x) + w(x^i) + \sum_{j=2, j\neq i}^{N} w(x^j)} \mathbb{1}_{\mathsf{A}}(x^i) \prod_{j=2}^{N} \lambda(\mathrm{d}x^j) \\
&\geq \sum_{i=2}^{N} \int \pi(\mathrm{d}x^i)\mathbb{1}_{\mathsf{A}}(x^i) \int \frac{\lambda(w)}{w(x) + w(x^i) + \sum_{j=2, j\neq i}^{N} w(x^j)} \prod_{j=2, j\neq i}^{N} \lambda(\mathrm{d}x^j).
\end{aligned}$$

Finally, since the function $f\colon z \mapsto (z+a)^{-1}$ is convex on $\mathbb{R}_+$ and $a > 0$, we get for $i \in \{2, \dots, N\}$,

$$\int \frac{\lambda(w)}{w(x) + w(x^i) + \sum_{j=2, j\neq i}^{N} w(x^j)} \prod_{j=2, j\neq i}^{N} \lambda(\mathrm{d}x^j) \tag{A.13}$$

$$\geq \frac{\lambda(w)}{\int w(x) + w(x^i) + \sum_{j=2, j\neq i}^{N} w(x^j) \prod_{j=2, j\neq i}^{N} \lambda(\mathrm{d}x^j)} \tag{A.14}$$

$$\geq \frac{1}{w(x)/\lambda(w) + w(x^i)/\lambda(w) + N - 2} \geq \frac{1}{2\omega + N - 2}. \tag{A.15}$$

We finally obtain the inequality

$$\mathsf{P}_N(x, \mathsf{A}) \geq \pi(\mathsf{A}) \times \frac{N-1}{2\omega + N - 2} = \epsilon_N \pi(\mathsf{A}). \tag{A.16}$$

This means that the whole space $\mathbb{X}$ is $(1, \epsilon_N \pi)$-small (see (Douc et al., 2018, Definition 9.3.5)). Since $\mathsf{P}_N(x, \cdot)$ and π are probability measures, (A.16) implies

$$\|\mathsf{P}_N(x, \cdot) - \pi\|_{\mathrm{TV}} = \sup_{\mathsf{A} \in \mathcal{X}} |\mathsf{P}_N(x, \mathsf{A}) - \pi(\mathsf{A})| \leq 1 - \epsilon_N = \kappa_N. \tag{A.17}$$

Now the statement follows from (Douc et al., 2018, Theorem 18.2.4) applied with $m = 1$.

A.1.6 Proof of Theorem 4

Proof of (ii). Using the identity $(a+b)^2 \leq (1+\epsilon^2)a^2 + (1+\epsilon^{-2})b^2$ we obtain the decomposition $\{\Pi_N f(X_k^{1:N}) - \pi(f)\}^2 \leq (1 + (N-1)^{-1/2})\mathrm{I}^{(1)} + (1 + (N-1)^{1/2})\mathrm{I}^{(2)}$, with

$$\mathrm{I}^{(1)} = \{\Pi_N f(X_k^{1:N}) - a_N(Y_{k-1})/b_N(Y_{k-1})\}^2, \tag{A.18}$$

$$\mathrm{I}^{(2)} = \{a_N(Y_{k-1})/b_N(Y_{k-1}) - \pi(f)\}^2, \tag{A.19}$$

where $a_N(Y_{k-1}) = \mathbf{\Lambda}_N \Gamma_N f(Y_{k-1})$ and $b_N(Y_{k-1}) = \mathbf{\Lambda}_N \Gamma_N \mathbb{1}_{\mathbb{X}}(Y_{k-1})$.

Using the identity $a/b - c/d = (1/d)[(a/b)(d-b) - (c-a)]$, we obtain

$$\begin{aligned} &\Pi_N f(X_k^{1:N}) - a_N(Y_{k-1})/b_N(Y_{k-1}) \\ &= b_N(Y_{k-1})^{-1} \left[\Pi_N f(X_k^{1:N})(b_N(Y_{k-1}) - \Gamma_N \mathbb{1}_{\mathbb{X}}(X_k^{1:N})) - (a_N(Y_{k-1}) - \Gamma_N f(X_k^{1:N}))\right]. \end{aligned} \tag{A.20}$$

Therefore, using the trivial bound $(a+b)^2 \leq 2(a^2+b^2)$, we get

$$\mathrm{I}^{(1)} \leq \frac{2}{b_N(Y_{k-1})^2} [\Pi_N f(X_k^{1:N})^2 \{\Gamma_N \mathbb{1}_{\mathbb{X}}(X_k^{1:N}) - b_N(Y_{k-1})\}^2 + \{\Gamma_N f(X_k^{1:N}) - a_N(Y_{k-1})\}^2]. \tag{A.21}$$

Since $\Pi_N f(X_k^{1:N})^2 \leq 1$, $\mathbb{P}_{\boldsymbol{\xi}}$-a.s., and $b_N(y) \geq (N-1)/N\lambda(w)$, it holds, $\mathbb{P}_{\boldsymbol{\xi}}$-a.s.,

$$\mathrm{I}^{(1)} \leq \frac{2N^2}{(N-1)^2 \lambda(w)^2} \left[\{\Gamma_N \mathbb{1}_{\mathbb{X}}(X_k^{1:N}) - b_N(Y_{k-1})\}^2 + \{\Gamma_N f(X_k^{1:N}) - a_N(Y_{k-1})\}^2\right]. \tag{A.22}$$

Therefore, using Lemma 8,

$$\mathbb{E}_{\boldsymbol{\xi}}[\{\Pi_N f(X_k^{1:N}) - a_N(Y_{k-1})/b_N(Y_{k-1})\}^2] \tag{A.23}$$

$$= \mathbb{E}_{\boldsymbol{\xi}}\left[\mathbb{E}_{\boldsymbol{\xi}}\left[\{\Pi_N f(X_k^{1:N}) - a_N(Y_{k-1})/b_N(Y_{k-1})\}^2 \,\middle|\, Y_{k-1}\right]\right] \tag{A.24}$$

$$\leq \frac{2N^2}{(N-1)^2\lambda(w)^2} \left[(N-1)/N^2 \lambda(\{w - \lambda(w)\}^2) + (N-1)/N^2 \lambda(\{wf - \lambda(wf)\}^2)\right] \tag{A.25}$$

$$\leq 4(N-1)^{-1}\kappa[\pi, \lambda]. \tag{A.26}$$

We turn to $\mathrm{I}^{(2)}$ and note that (2.20) implies that $\mathrm{I}^{(2)} \leq 4N^{-2}(1+\omega)^2$, which completes the proof. □

Proof of (iii). Note that

$$\mathrm{I}^{(3)} = \mathbb{E}_{\boldsymbol{\xi}}\left[\{\Pi_N f(X_k^{1:N}) - \pi(f)\}\{\Pi_N f(X_{k+\ell}^{1:N}) - \pi(f)\}\right] \tag{A.27}$$

$$= \mathbb{E}_{\boldsymbol{\xi}}[\{\Pi_N f(X_k^{1:N}) - \pi(f)\}\mathbb{E}_{\boldsymbol{\xi}}\left[\Pi_N f(X_{k+\ell}^{1:N}) - \pi(f) \,\middle|\, Y_{k+\ell-1}\right]]. \tag{A.28}$$

As $\mathbb{E}_{\boldsymbol{\xi}}\left[\Pi_N f(X_{k+\ell}^{1:N}) \,\middle|\, Y_{k+\ell-1}\right] = \Phi_N(Y_{k+\ell-1})$ $\mathbb{P}_{\boldsymbol{\xi}}$-a.s., it holds that

$$\mathrm{I}^{(3)} = \mathbb{E}_{\boldsymbol{\xi}}[\{\Pi_N f(X_k^{1:N}) - \pi(f)\}\{\Phi_N(Y_{k+\ell-1}) - \pi(f)\}] \tag{A.29}$$

$$= \mathbb{E}_{\boldsymbol{\xi}}[\{\Pi_N f(X_k^{1:N}) - \pi(f)\}\{\mathbb{E}_{\boldsymbol{\xi}}\left[\Phi_N(Y_{k+\ell-1}) \,|\, Y_k\right] - \pi(f)\}]. \tag{A.30}$$

By the Markov property,

$$\mathbb{E}_{\boldsymbol{\xi}}\left[\Phi_N(Y_{k+\ell-1}) \,|\, Y_k\right] = \mathsf{P}_N^{\ell-1}\Phi_N(Y_k) = \delta_{Y_k}\mathsf{P}_N^{\ell-1}\Phi_N, \quad \mathbb{P}_{\boldsymbol{\xi}}\text{-a.s.}, \tag{A.31}$$

which, combined with (2.14), implies that

$$\|\mathsf{P}_N^{\ell-1}\Phi_N - \pi(f)\|_\infty \leq \varsigma^{bias}(N-1)^{-1}\kappa_N^{\ell-1}. \tag{A.32}$$

Combining the results above, we finally establish that

$$|\mathrm{I}^{(3)}| \leq \varsigma^{bias}(N-1)^{-1}\kappa_N^{\ell-1}\mathbb{E}_{\boldsymbol{\xi}}[\{\Pi_N f(X_k^{1:N}) - \pi(f)\}^2]^{1/2} \tag{A.33}$$

$$\leq \varsigma^{bias}(N-1)^{-1}\kappa_N^{\ell-1}\left(\sum_{i=0}^{2}\varsigma_i^{mse}(N-1)^{-1-i/2}\right)^{1/2}. \tag{A.34}$$

□

A.1.7 Proof of Theorem 5

We first consider the bias term, which can be bounded according to

$$\left|\mathbb{E}_{\boldsymbol{\xi}}[\Pi_{(K_0,K),N}(f)] - \pi(f)\right| \leq (K-K_0)^{-1}\sum_{\ell=K_0+1}^{K}\left|\mathbb{E}_{\boldsymbol{\xi}}[\Pi_N f(X_\ell^{1:N})] - \pi(f)\right| \tag{A.35}$$

$$\leq (K-K_0)^{-1}(N-1)^{-1}\varsigma^{bias}\sum_{\ell=K_0+1}^{K}\kappa_N^{\ell-1}. \tag{A.36}$$

Thus, the claimed bias bound can be established by noting that

$$\sum_{\ell=K_0+1}^{K}\kappa_N^{\ell-1} \leq \frac{\kappa_N^{K_0}}{1-\kappa_N} \leq \frac{4\tau_{mix,N}(1/4)^{K_0/\tau_{mix,N}}}{3}. \tag{A.37}$$

We now turn to the MSE, and make the decomposition

$$\mathbb{E}_{\boldsymbol{\xi}}[\{\Pi_{(K_0,K),N}(f) - \pi(f)\}^2] \leq (K-K_0)^{-2}\left(\sum_{\ell=K_0+1}^{K}\mathbb{E}_{\boldsymbol{\xi}}[\Pi_N f(X_\ell^{1:N})] - \pi(f)\right)^2 \tag{A.38}$$

$$+2\sum_{\ell=K_0+1}^{K}\sum_{j=\ell+1}^{K}\mathbb{E}_{\boldsymbol{\xi}}[\{\Pi_N f(X_\ell^{1:N}) - \pi(f)\}\{\Pi_N f(X_j^{1:N}) - \pi(f)\}]. \tag{A.39}$$

Using the MSE bound in Theorem 38, we obtain that

$$\sum_{\ell=K_0+1}^{K} \mathbb{E}_{\boldsymbol{\xi}}[\{\Pi_N f(X_\ell^{1:N}) - \pi(f)\}^2] \leq (K-K_0)(N-1)^{-1} \sum_{i=0}^{2} \varsigma_i^{mse} (N-1)^{-i/2}. \tag{A.40}$$

In addition, the covariance bound of Theorem 38 yields

$$\sum_{\ell=K_0+1}^{K} \sum_{j=\ell+1}^{K} \mathbb{E}_{\boldsymbol{\xi}}[\{\Pi_N f(X_\ell^{1:N}) - \pi(f)\}\{\Pi_N f(X_j^{1:N}) - \pi(f)\}]$$
$$\leq \sum_{i=0}^{2} \varsigma_i^{cov} (N-1)^{-(3-i/2)/2} \left(\sum_{\ell=K_0+1}^{K} \sum_{j=\ell+1}^{K} \kappa_N^{(j-\ell)-1} \right). \tag{A.41}$$

As $\sum_{\ell=K_0+1}^{K} \sum_{j=\ell+1}^{K} \kappa_N^{(j-\ell)-1} \leq (K-K_0)(4/3)\tau_{mix,N}$, we may write

$$\mathbb{E}_{\boldsymbol{\xi}}[(\Pi_{(K_0,K),N}(f) - \pi(f))^2] \leq ((K-K_0)(N-1))^{-1} \left(\sum_{i=0}^{2} \varsigma_i^{mse} (N-1)^{-i/2} \right) \tag{A.42}$$
$$+(8/3)(K-K_0)^{-1}(N-1)^{-3/2} \left(\sum_{i=0}^{2} \varsigma_i^{cov} (N-1)^{-i/4} \right), \tag{A.43}$$

and the MSE bound may now be established by noting that $(K-K_0)(N-1) = \upsilon M$.

Establishing the high-probability bound requires more complex derivations. More precisely, we will apply the decomposition

$$\Pi_{(K_0,K),N}(f) - \pi(f) = (K-K_0)^{-1} \sum_{k=K_0+1}^{K} \Pi_N f(X_k^{1:N}) - \Phi_N(Y_{k-1})$$
$$+ (K-K_0)^{-1} \sum_{k=K_0+1}^{K-1} \Phi_N(Y_{k-1}) - \pi(\Phi_N), \tag{A.44}$$

where we used that $\pi(f) = \pi(\Phi_N)$. Therefore, for every $t \geq 0$ it holds that

$$\mathbb{P}_{\boldsymbol{\xi}}(|\Pi_{(K_0,K),N}(f) - \pi(f)| \geq t) \leq \mathbb{P}_{\boldsymbol{\xi}} \left((K-K_0)^{-1} \left| \sum_{k=K_0+1}^{K} \Pi_N f(X_k^{1:N}) - \Phi_N(Y_{k-1}) \right| \geq t/2 \right)$$
$$+ \mathbb{P}_{\boldsymbol{\xi}} \left((K-K_0)^{-1} \left| \sum_{k=K_0+1}^{K-1} \Phi_N(Y_{k-1}) - \pi(\Phi_N) \right| \geq t/2 \right). \tag{A.45}$$

We will show that for all $t > 0$, and for some absolute constants $\zeta^{(1)}$ and $\zeta^{(2)}$,

$$\mathrm{I}^{(1)} = \mathbb{P}_{\boldsymbol{\xi}} \left((K-K_0)^{-1} \left| \sum_{k=K_0+1}^{K} \Pi_N f(X_k^{1:N}) - \Phi_N(Y_{k-1}) \right| \geq t \right) \leq 2\exp(-t^2 \upsilon M/(4\zeta^{(1)})), \tag{A.46}$$

$$\mathrm{I}^{(2)} = \mathbb{P}_{\boldsymbol{\xi}} \left((K-K_0)^{-1} \left| \sum_{k=K_0+1}^{K-1} \Phi_N(Y_{k-1}) - \pi(\Phi_N) \right| \geq t \right) \tag{A.47}$$
$$\leq 2\exp(-t^2 \zeta^{(2)} (K-K_0)(N-1)^2/\tau_{mix,N}). \tag{A.48}$$

We first consider $\mathrm{I}^{(1)}$ and note that

$$\mathrm{I}^{(1)} = \mathbb{E}_{\boldsymbol{\xi}}\left[\mathbb{P}_{\boldsymbol{\xi}}\left((K-K_0)^{-1}\left|\sum_{k=K_0+1}^{K}\Pi_N f(X_k^{1:N}) - \Phi_N(Y_{k-1})\right| \geq t \mid Y_{K_0:K-1}\right)\right]. \tag{A.49}$$

By Theorem 6, the random elements $(X_k^{1:N})_{k=K_0+1}^{K}$ are independent conditionally to $(Y_k)_{k=K_0}^{K-1}$. Thus, using the generalized Hoeffding inequality (see (Vershynin, 2018, Theorem 2.6.2) or (Wainwright, 2019, Proposition 2.1)) we get, with $\Delta_{N,k} = \Pi_N f(X_k^{1:N}) - \Phi_N(Y_{k-1})$, that, $\mathbb{P}_{\boldsymbol{\xi}}$-a.s.,

$$\mathbb{P}_{\boldsymbol{\xi}}\left((K-K_0)^{-1}\left|\sum_{k=K_0+1}^{K}\Delta_{N,k}\right| \geq t \mid Y_{K_0:K-1}\right) \leq 2\exp\left(-\frac{t^2(K-K_0)^2}{4\sum_{k=K_0+1}^{K}\|\Delta_{N,k}\|_{\psi_2,Y_k}^2}\right), \tag{A.50}$$

where $\psi_2 : x \mapsto \exp(x^2) - 1$ and

$$\|\Delta_{N,k}\|_{\psi_2,Y_{k-1}} = \inf\left\{\lambda > 0 : \mathbb{E}_{\boldsymbol{\xi}}\left[\psi_2(|\Delta_{N,k}|/\lambda) \,|\, Y_{k-1}\right] \leq 1\right\}.$$

In order to bound $\|\Delta_{N,k}\|_{\psi_2,Y_{k-1}}$ we use the decomposition $\Delta_{N,k} = \Delta_{N,k}^{(1)} + \Delta_{N,k}^{(2)}$, where

$$\Delta_{N,k}^{(1)} = \frac{\Gamma_N f(X_k^{1:N})}{\Gamma_N \mathbb{1}_{\mathbb{X}}(X_k^{1:N})} - \frac{a_N(Y_{k-1})}{b_N(Y_{k-1})}, \tag{A.51}$$

$$\Delta_{N,k}^{(2)} = \frac{a_N(Y_{k-1})}{b_N(Y_{k-1})} - \Phi_N(Y_{k-1}), \tag{A.52}$$

combined with Lemma 36 with $\phi = \chi = \psi_2$ and (Vershynin, 2018, Proposition 2.6.1). By (2.18) and by (Vershynin, 2018, Equation 2.17) it holds that, $\mathbb{P}_{\boldsymbol{\xi}}$-a.s.,

$$\|\Delta_{N,k}^{(2)}\|_{\psi_2,Y_{k-1}} \leq 2(\log 2)^{-1/2}(N-1)^{-1}\kappa[\lambda,\pi]. \tag{A.53}$$

Using Lemma 36 with $\phi = \chi = \psi_2$ and the fact that $b_N(y) \geq (1-1/N)\lambda(w)$ we obtain, $\mathbb{P}_{\boldsymbol{\xi}}$-a.s.,

$$\begin{aligned}&\|\Delta_{N,k}^{(1)}\|_{\psi_2,Y_{k-1}} \\ &\leq \frac{2}{(1-1/N)\lambda(w)}\left(\|\Gamma_N f(X_k^{1:N}) - a_N(Y_{k-1})\|_{\psi_2,Y_{k-1}} + 2\|\Gamma_N \mathbb{1}_{\mathbb{X}}(X_k^{1:N}) - b_N(Y_{k-1})\|_{\psi_2,Y_{k-1}}\right).\end{aligned} \tag{A.54}$$

Furthermore, using (Vershynin, 2018, Proposition 2.6.1, Eq 2.17) we get, $\mathbb{P}_{\boldsymbol{\xi}}$-a.s.,

$$\|\Gamma_N f(X_{k-1}^{1:N}) - a_N(Y_{k-1})\|_{\psi_2,Y_{k-1}}^2 \tag{A.55}$$

$$\leq (64\mathrm{e}/\log 2)N^{-1}\left\|w(X_k^1)f(X_k^1) - \mathbb{E}_{\boldsymbol{\xi}}\left[w(X_k^1)f(X_k^1)\,\middle|\, Y_{k-1}\right]\right\|_{\psi_2,Y_{k-1}}^2, \tag{A.56}$$

$$\leq (256\mathrm{e}/(\log 2)^2)N^{-1}\|w\|_\infty^2. \tag{A.57}$$

The same bound applies to $\|\Gamma_N \mathbb{1}_{\mathbb{X}}(X_k^{1:N}) - b_N(Y_{k-1})\|_{\psi_2,Y_{k-1}}^2$, and we may write

$$\|\Delta_{N,k}^{(1)}\|_{\psi_2,Y_{k-1}} \leq 96\mathrm{e}^{1/2}(\log 2)^{-1}(N-1)^{-1/2}\omega. \tag{A.58}$$

We can now finalize the bound on $\mathrm{I}^{(1)}$ by writing

$$\|\Delta_{N,k}\|_{\psi_2,Y_{k-1}}^2 \leq 2(\|\Delta_{N,k}^{(1)}\|_{\psi_2,Y_{k-1}}^2 + \|\Delta_{N,k}^{(2)}\|_{\psi_2,Y_{k-1}}^2) \tag{A.59}$$

$$\leq (N-1)^{-1}(\zeta^{(1,1)}\omega^2 + \zeta^{(1,2)}\kappa[\lambda,\pi]^2(N-1)^{-1}), \tag{A.60}$$

where $\zeta^{(1,1)} = 18432\mathrm{e}(\log 2)^{-2}$ and $\zeta^{(1,2)} = 8(\log 2)^{-1}$ are universal constants, which implies that

$$\|\Delta_{N,k}\|^2_{\psi_2, Y_{k-1}} \leq \zeta^{(1)}(N-1)^{-1}, \tag{A.61}$$

with $\zeta^{(1)} = 1.1 \cdot 10^5 \omega^2$. This finally yields that $\mathrm{I}^{(1)} \leq 2\exp(-t^2 \upsilon M / 4\zeta^{(1)})$.

We treat $\mathrm{I}^{(2)}$ using Lemma 39 with $g_i = \Phi_N(Y_{K_0+i-1}) - \pi(\Phi_N)$. As $\|g_i\|_\infty \leq \mathrm{osc}(\Phi_N) \leq (N-1)^{-1}\varsigma^{bias}$, we obtain

$$\mathrm{I}^{(2)} \leq 2\exp\left(-t^2 \zeta^{(2)}(K-K_0)(N-1)^2/\tau_{mix,N}\right), \tag{A.62}$$

where $\zeta^{(2)} = 2/(3\varsigma^{bias})^2$. Finally, we obtain

$$\begin{aligned}\mathbb{P}_{\boldsymbol{\xi}}(|\Pi_{(K_0,K),N}(f) - \pi(f)| \geq t) \\ \leq 2\exp\left(-t^2\upsilon M/4\zeta^{(1)}\right)\left[1 + \exp\left(-t^2\upsilon M\{\zeta^{(2)}(N-1)/\tau_{mix,N} - (4\zeta_I)^{-1}\}\right)\right].\end{aligned} \tag{A.63}$$

We conclude by noting that for every $\delta \in (0,1)$ and $N - 1 \geq \tau_{mix,N}(4\zeta^{(1)}\zeta^{(2)})^{-1}$ it holds that

$$\mathbb{P}_{\boldsymbol{\xi}}(|\Pi_{(K_0,K),N}(f) - \pi(f)| \geq t) \leq 4\exp\left(-t^2\upsilon M/4\zeta^{(1)}\right) \leq \delta \tag{A.64}$$

for all $t \geq 2\zeta_I^{1/2}(\upsilon M)^{-1/2}\log(4/\delta)^{1/2}$. Letting $\varsigma^{hpd} = 2\zeta_I^{1/2}$ concludes the proof.

A.1.8 High-probability inequality for SNIS

Theorem 35. *Assume that $\omega = \|w\|_\infty/\lambda(w) < \infty$. For all bounded measurable functions f on $(\mathbb{X}, \mathcal{X})$ such that $\|f\|_\infty \leq 1$, it holds that for every $M \in \mathbb{N}^*$ and $\delta \in (0,1)$,*

$$|\widehat{\pi}_M(f) - \pi(f)| \leq 12\omega(M\log 2)^{-1/2}\log(2/\delta)^{1/2} \tag{A.65}$$

with probability larger than $1-\delta$.

Proof. Let $\alpha_M = M^{-1}\sum_{i=1}^M w(X^i)f(X^i)$, $\beta_M = M^{-1}\sum_{i=1}^M w(X^i)$, $a = \mathbb{E}[\alpha_M] = \lambda(wf)$, and $b = \mathbb{E}[\beta_M] = \lambda(w)$. Note that $\widehat{\pi}_M(f) = \alpha_M/\beta_M$ and $\pi(f) = a/b$. Using Lemma 36 with ϕ and χ equal to the mapping $x \mapsto \exp(x^2) - 1$ we obtain that

$$\|\widehat{\pi}_M(f) - \pi(f)\|_{\psi_2} \leq 2\lambda(w)^{-1}\left(\|\alpha_M - a\|_{\psi_2} + 2\|\beta_M - b\|_{\psi_2}\right). \tag{A.66}$$

Moreover, using (Vershynin, 2018, Eq 2.17) yields, $\mathbb{P}_{\boldsymbol{\xi}}$-a.s.,

$$\|\alpha_M - a\|^2_{\psi_2} \leq M^{-1}\|w(X^i)f(X^i) - \lambda(wf)\|^2_{\psi_2} \leq 4(M\log 2)^{-1}\|w\|^2_\infty. \tag{A.67}$$

In the same way, $\|\beta_M - b\|^2_{\psi_2} \leq 4(M\log 2)^{-1}\|w\|^2_\infty$. Therefore, we may conclude that

$$\|\widehat{\pi}_M(f) - \pi(f)\|^2_{\psi_2} \leq (12\omega)^2(M\log 2)^{-1}. \tag{A.68}$$

Combining the previous bound with (Vershynin, 2018, Proposition 2.5.2) provides

$$\mathbb{P}(|\widehat{\pi}_M(f) - \pi(f)| \geq t) \leq 2\exp(-t^2\zeta^{snis}M), \tag{A.69}$$

where $\zeta^{snis} = (12\omega)^{-2}\log 2$. The high-probability inequality of the theorem follows directly. □

A.2 Moments and high-probability bounds for ratio statistics

Let $(U_i, V_i)_{i\in\{1,\dots,n\}}$ be (possibly dependent) random variables defined on some probability space $(\Omega, \mathcal{F}, \mathbb{P})$. Assume that $U_i \geq 0$ $\mathbb{P}$-a.s. Moreover, let $\alpha_n = n^{-1}\sum_{i=1}^n U_i V_i$, $\beta_n = n^{-1}\sum_{i=1}^n U_i$, and $\rho_n = \alpha_n/\beta_n$ as well as $a = \mathbb{E}[\alpha_n]$, $b = \mathbb{E}[\beta_n]$, and $r = a/b$.

A continuous, even, convex function $\phi : \mathbb{R}^+ \to [0, +\infty]$ is a Young function if ϕ is monotonically increasing for $x > 0$, $\phi(0) = 0$, $\lim_{x\to\infty} \phi(x)/x = \infty$, and $\lim_{x\to 0^+} \phi(x)/x = 0$. We denote by ϕ^* the Fenchel-Legendre conjugate of ϕ. Let X be a random variable and ϕ a Young function. Then the *Orlicz norm* of X is

$$\|X\|_\phi = \inf\left\{\lambda > 0 : \mathbb{E}\left[\phi\left(|X|/\lambda\right)\right] \leq 1\right\}, \tag{A.70}$$

with the convention that $\inf\emptyset = \infty$. The Orlicz space $\mathcal{L}_\phi(\Omega)$ of random variables is the family of equivalence classes of random variables X such that $\|X\|_\phi < \infty$. Here $\mathcal{L}_\phi(\Omega)$ is a Banach space. If $\phi_p(x) = |x|^p$ for $p \geq 1$, then $\mathcal{L}_\phi(\Omega) = \mathcal{L}^p(\Omega)$ and we denote $\|\cdot\|_p = \|\cdot\|_{\phi_p}$. If $X \in \mathcal{L}_\phi(\Omega)$, then, for every $x > 0$,

$$\mathbb{P}(|X| \geq x) \leq 1/\phi(x/\|X\|_\phi) \quad \text{and} \quad \|\mathbb{1}_{\{|X|\geq x\}}\|_\phi = 1/\phi^{-1}(1/\mathbb{P}(|X| \geq x)).$$

Lemma 36. *Let ϕ and χ be Young functions. If $\max_i \|V_i\|_\infty \leq c|r|$, then*

$$\|\rho_n - r\|_\phi/|r| \leq 2\|\alpha_n - a\|_\phi/b + 2\|\beta_n - b\|_\phi/b + c/\{(\phi^{-1}\circ\chi)(b/2\|(\beta_n - b)_-\|_\chi)\}. \tag{A.71}$$

Proof. We decompose the computation in two parts: first, when $\beta_n > b/2$, we have

$$|\rho_n - r| = \left|\frac{\alpha_n - a}{\beta_n} + a\left(\frac{1}{\beta_n} - \frac{1}{b}\right)\right| \leq \frac{|\alpha_n - a|}{b/2} + \frac{|a||\beta_n - b|}{(b/2)b} = \frac{2|\alpha_n - a|}{b} + \frac{2|r||\beta_n - b|}{b}.$$

Then, when $\beta_n \leq b/2$,

$$|\rho_n - r| \leq |\rho_n| + |r| \leq |\rho_n| + \frac{2|r||\beta_n - b|}{b} \leq \max_i |V_i| + \frac{2|r||\beta_n - b|}{b}, \tag{A.72}$$

where the second inequality follows from $|\beta_n - b| \geq b/2$. Combining the two previous inequalities yields

$$|\rho_n - r| \leq \frac{2|\alpha_n - a|}{b} + \frac{2|r||\beta_n - b|}{b} + \max_i |V_i|\mathbb{1}_{\{\beta_n \leq b/2\}}. \tag{A.73}$$

Recall that if $|X| \leq |Y|$ $\mathbb{P}$-a.s., then $\|X\|_\phi \leq \|Y\|_\phi$; hence, we may proceed like

$$\|\rho_n - r\|_\phi \leq \left\|\frac{2|\alpha_n - a|}{b} + \frac{2r|\beta_n - b|}{b} + \max_i |V_i|\mathbb{1}_{\{\beta_n \leq b/2\}}\right\|_\phi \tag{A.74}$$

$$\leq \frac{2\|\alpha_n - a\|_\phi}{b} + \frac{2|r|\|\beta_n - b\|_\phi}{b} + c|r|\|\mathbb{1}_{\{\beta_n \leq b/2\}}\|_\phi \tag{A.75}$$

$$= \frac{2\|\alpha_n - a\|_\phi}{b} + \frac{2|r|\|\beta_n - b\|_\phi}{b} + c|r|/\phi^{-1}\left(1/\mathbb{P}(\beta_n \leq b/2)\right). \tag{A.76}$$

Finally, we obtain the desired result by noting that for any Young function χ, $\mathbb{P}(\beta_n \leq b/2) = \mathbb{P}(|(\beta_n - b)_-| \geq b/2) \leq 1/\chi(b/2\|(\beta_n - b)_-\|_\chi)$. □

Theorem 37. *Let $p \geq 1$. If $\max_i \|V_i\|_\infty \leq c|r|$, then*

$$\frac{\|\rho_n - r\|_p}{|r|} \leq \frac{2\|\alpha_n - a\|_p}{b} + \frac{2(1+c)\|\beta_n - b\|_p}{b}. \tag{A.77}$$

Proof. Apply Lemma 36 with $\chi(x) = \phi(x) = x^p$. □

Theorem 38. *If* $|\alpha_n/\beta_n| \leq 1$ $\mathbb{P}$*-a.s., then*

$$|\mathbb{E}[\rho_n] - r| \leq (2b^2)^{-1}\{3\mathbb{E}[(\beta_n - b)^2] + \mathbb{E}[(\alpha_n - a)^2]\}. \tag{A.78}$$

Proof. Using the identity

$$\frac{\alpha_n}{\beta_n} - \frac{a}{b} = \frac{\alpha_n}{\beta_n}\frac{(b-\beta_n)^2}{b^2} + \frac{(\alpha_n - a)(b-\beta_n)}{b^2} + \frac{a(b-\beta_n)}{b^2} + \frac{\alpha_n - a}{b}, \tag{A.79}$$

yields

$$\mathbb{E}[\rho_n] - r = \mathbb{E}\left[\frac{\alpha_n}{\beta_n}\frac{(b-\beta_n)^2}{b^2}\right] + \frac{\mathbb{E}[(\alpha_n - a)(b-\beta_n)]}{b^2},$$

which completes the proof. □

We conclude with a lemma that gives the concentration of a uniformly ergodic Markov chain. We think that this Lemma is of independent interest, and we give it under general conditions.

Lemma 39. *Let* $(\mathbb{Z}, \mathcal{Z})$ *be a state-space and* Q *a Markov kernel on* $(\mathbb{Z}, \mathcal{Z})$ *which is uniformly ergodic with mixing time* t_{mix} *and stationary distribution* π*. Let* $(g_i)_{i=1}^n$ *be a family of* $\mathbb{R}^d$*-valued measurable functions on* $\mathbb{Z}$ *such that* $\|g\|_\infty = \max_{i\in\{1,\dots,n\}} \|g_i\|_\infty < \infty$ *and* $\pi(g_i) = 0$ *for all* $i \in \{1, \dots, n\}$*. Then for every initial probability* ξ *on* $(\mathbb{Z}, \mathcal{Z})$*,* $n \in \mathbb{N}$*, and* $t \geq 0$*,*

$$\mathbb{P}_\xi\left(\left\|\sum_{i=1}^n g_i(Z_i)\right\| \geq t\right) \leq 2\exp\left(-\frac{2t^2}{u_n^2}\right), \tag{A.80}$$

where $u_n = 3\|g\|_\infty\sqrt{nt_{mix}}$*.*

Proof. The function $\varphi(x_1^{1:N}, \dots, x_n^{1:N}) = \|\sum_{i=1}^n g_i(x_i^{1:N})\|$ on $\mathbb{Z}^n$ satisfies the bounded differences property. Applying (Paulin, 2015, Corollary 2.10), we get, for $t \geq \mathbb{E}_\xi[\|\sum_{i=1}^n g_i(Z_i)\|]$,

$$\mathbb{P}_\xi\left(\left\|\sum_{i=1}^n g_i(Z_i)\right\| \geq t\right) \leq \exp\left\{-\frac{2(t - \mathbb{E}_\xi[\|\sum_{i=1}^n g_i(Z_i)\|])^2}{9n\|g\|_\infty^2 t_{mix}}\right\}. \tag{A.81}$$

It remains to bound $\mathbb{E}_\xi\left[\|\sum_{i=1}^n g_i(Z_i)\|\right]$ from above. For this purpose, note that

$$\mathbb{E}_\xi\left[\left\|\sum_{i=1}^n g_i(Z_i)\right\|^2\right] = \sum_{i=1}^n \mathbb{E}_\xi\left[\|g_i(Z_i)\|^2\right] + 2\sum_{k=1}^{n-1}\sum_{\ell=1}^{n-k}\mathbb{E}_\xi[g_k(Z_k)^\intercal g_{k+\ell}(Z_{k+\ell})], \tag{A.82}$$

where, using that $\pi(g_{k+\ell}) = 0$,

$$\left|\mathbb{E}_\xi[g_k(Z_k)^\intercal g_{k+\ell}(Z_{k+\ell})]\right| = \left|\int g_k(z)^\intercal\left(\mathrm{Q}^\ell\, g_{k+\ell}(z) - \pi(g_{k+\ell})\right)\xi\,\mathrm{Q}^k(\mathrm{d}z)\right| \leq \|g\|_\infty^2(1/4)^{\lceil \ell/t_{mix}\rceil}, \tag{A.83}$$

which implies that

$$\sum_{k=1}^{n-1}\sum_{\ell=1}^{n-k}\left|\mathbb{E}_\xi[g_k(Z_k)^\intercal g_{k+\ell}(Z_{k+\ell})]\right| \leq \sum_{k=1}^{n-1}\|g\|_\infty^2(1/4)^{\lceil \ell/t_{mix}\rceil} \leq (4/3)\|g\|_\infty^2 t_{mix} n. \tag{A.84}$$

Combining the bounds above, we obtain the upper bound

$$\mathbb{E}_\xi\left[\left\|\sum_{i=1}^n g_i(Z_i)\right\|\right] \leq \left(\mathbb{E}_\xi\left[\left\|\sum_{i=1}^n g_i(Z_i)\right\|^2\right]\right)^{1/2} \leq 2\sqrt{n}\|g\|_\infty\sqrt{t_{mix}} \overset{\text{not.}}{=} v_n. \tag{A.85}$$

By plugging this result into (A.80), we obtain that

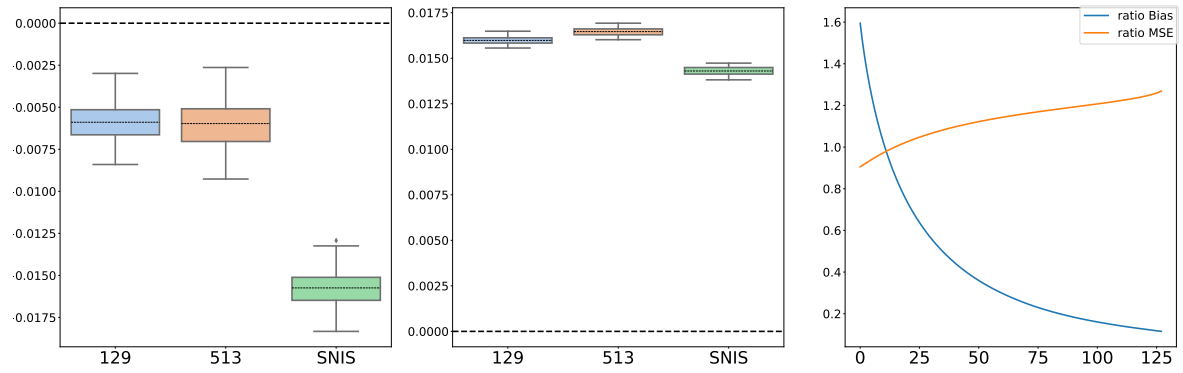
$$\mathbb{P}_\xi\left(\left\|\sum_{i=1}^n g_i(Z_i)\right\| \geq t\right) \leq \begin{cases} 1, & t < v_n, \\ \exp\left(-\frac{2(t-v_n)^2}{9v_n^2}\right), & t \geq v_n. \end{cases} \tag{A.86}$$

Now, since the right-hand side of (A.86) is, for every $t \geq 0$, upper bounded by $2\exp(-2t^2/(9v_n^2))$, the statement of the lemma follows. □

A.3 Experiments

A.3.1 Gaussian Mixture

Bias MSE trade-off: We display in Figures A.1a and A.1b the bias and the MSE of the BR-SNIS estimators for the same configuration as in Figure 2.2 but with $k_0 = \lfloor 0.625 k_{max} \rfloor$. We observe 3 times less bias than the SNIS estimators but only with a 10% increase of the MSE for the $N = 129$ setting. This can be also seen in Figure A.1c, where we show the ratio between BR-SNIS and SNIS for bias and MSE with $N = 129$.



(a) Bias *(b) MSE* *(c) Ratios as k_0*

*Figure A.1: Comparison between SNIS and BR-SNIS for the same budget. In each boxplot the dotted line represents the **mean** value of the samples. In Figure A.1c we display the ratio between BR-SNIS and SNIS for bias and MSE with $N = 129$.*

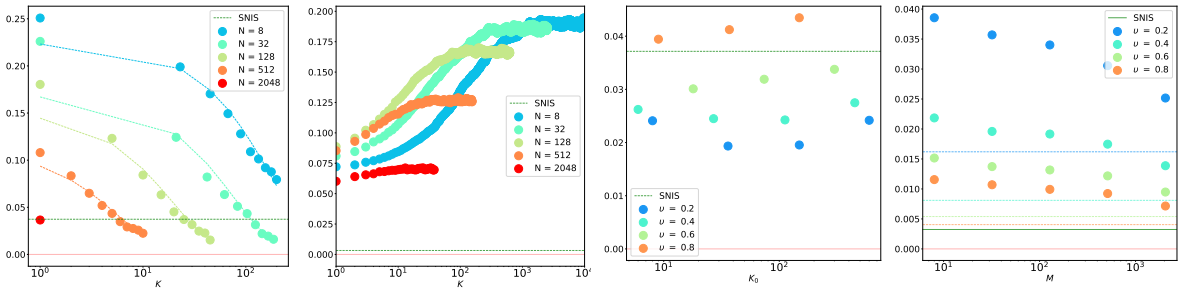
Parameters Gaussian mixture: The π in Section 2.3 is a Mixture of two Gaussians in dimension 7 with mean vectors $\boldsymbol{\mu}_1 = (1, \dots, 1)^\intercal$ and $\boldsymbol{\mu}_2 = (-2, 0, \dots, 0)^\intercal$ and covariance matrices $\boldsymbol{\Sigma}_1 = d^{-1}\mathbf{I}$ and $\boldsymbol{\Sigma}_2 = d^{-1}\mathbf{I}$, where $p = 1/3$ and $\mathbf{I}$ is the identity matrix In this setting, the quantities $\kappa[\pi, \lambda]$ and ω can be estimated by Monte Carlo and Gradient ascent respectively. Their values are approximately $7 \cdot 10^2$ and $1 \cdot 10^4$, respectively.

The sets A and B used to define the function f are the following:

$$A := [-2, 6] \times [-1, 1]^6, \quad B := [0.75, 1.25] \times [1, 2] \times [-0.1, 0.1]^5. \tag{A.87}$$

We used this example to illustrate numerically the bounds in Theorems 4 and 5, where each expectation was calculated by Monte Carlo using $2 \cdot 10^4$ samples. We displayed in each figure the equivalent SNIS estimation in a green dashed line. For all the bias related bounds(Theorem 4(i) in Figure A.2a, Theorem 5(i) in Figure A.2c), we fixed a total budget of $M = 6 \cdot 10^3$. For Figure A.2a we added a fit of the type $y = \exp(ak + b)$ to illustrate the exponential decay w.r.t. k.

We then increased the budget to $M = 8 \cdot 10^4$ for the MSE and covariance bounds, in order to fully observe the stabilisation of the MSE in Figure A.2b for all the minibatch sizes N. For the true value of $\pi(f)$ needed for calculating the MSE, we use an estimation obtained by Monte Carlo (sampling directly from π) with $4 \cdot 10^7$ samples. In Figure A.2d we added dashed lines with the theoretical value of the $\mathsf{MSE}^{is}_{\upsilon M}$ with the same color as υ.



(a) Bias *(b) MSE* *(c) Rolling bias* *(d) Rolling MSE*

Figure A.2: Visualization of the theoretical bounds from Theorems 4 and 5.

Comparison with zero bias SNIS methods: There exists estimators based on SNIS that have no bias, such as the estimator proposed in Middleton et al. (2019) and refered to as Unbiased-PIMH . One of the main differences between such estimator and BR-SNIS is that BR-SNIS works under a pre-established budget of samples, whereas in Unbiased-PIMH the number of samples used to produce an estimate varies due to the accept-reject procedure. Even though the two estimators have different goals, it can be of interest to compare both of them in the case where there is a restriction in the total number of samples available.

We proceed to a fixed-budget (M) comparison between BR-SNIS and the "Rao Blackwellized" version of the algorithm proposed at Middleton et al. (2019) in the Gaussian Mixture example. In order to do so, it's necessary to impose the fixed-budget constraint to the Unbiased-PIMH estimator. A single iteration of the estimator from Unbiased-PIMH with batch-size N needs rN samples where $r \in \mathbb{N}$ is a random number satisfying $r \geq 2$. Therefore, there are two ways of applying the constraint to Unbiased-PIMH :

- **Soft**: For a given N, generate estimations using Unbiased-PIMH until the number of samples is larger than M and **keep** the last estimation. Therefore, all the estimators from Unbiased-PIMH will have used **at least** M samples. All the estimations generated are averaged to generate a single estimate.
- **Hard**: For a given N, generate estimations using Unbiased-PIMH until the number of total samples used is larger than M and **discard** the last estimation. Therefore, all the estimators from Unbiased-PIMH will have used **at most** M samples. **If no estimations were produced under the budget cap (first iteration used more than M samples), then we consider it a miss**. All the estimations generated are averaged to create a single estimate.

The code used to run the experiments is available at [1]. For both cases, the following values of M are used

[1] https://github.com/gabrielvc/br_snis/blob/master/notebooks/Comparison_Unbiased-PIMH.ipynb

N	k	algorithm	Bias	std	average M
65536		SNIS	-0.0029	0.0605	65536.0
65	1024	BR-SNIS	-0.0010	0.0658	65536.0
129	512	BR-SNIS	-0.0006	0.0689	65536.0
257	256	BR-SNIS	0.0003	0.0678	65536.0
513	128	BR-SNIS	0.0019	0.0670	65536.0
16384		Unbiased-PIMH	0.0065	0.1005	71904.0
8192		Unbiased-PIMH	0.0058	0.1066	71040.0
4096		Unbiased-PIMH	0.0082	0.1139	69316.0
2048		Unbiased-PIMH	0.0053	0.1174	67764.0

*Table A.1: $M = 2^{16}$ in the **Soft** framework.*

in the comparison: $2^{16}, 2^{12}, 2^9$. For each estimator, a total of 1024 Monte Carlo replications are used to estimate the mean and the standard deviation of the estimator. Note that in the **Hard** framework, **it can happen that less than 1024 replications are used for the Unbiased-PIMH estimator**. The number of failed estimations is reported in the tables for the framework **Hard** for each configuration.

For each configuration of the BR-SNIS estimator (defined by N, k_{max}), we have used 90% burn-in period ($k_0 = \lfloor 0.9 k_{max} \rfloor$) and k_{max} rounds of bootstrap (k_{max} permutations of the input samples).

The following values were calculated:

- **Bias**: The mean of the estimations minus ref over 1024 replications
- **Std**: The standard deviation of the estimations over 1024 replications.
- **Fails**: The number of replications that failed to produce a single estimation for a given budget M. This is only applicable for the Unbiased-PIMH estimator and in the **Hard** framework.
- **average M**: The average (over the 1024 replications) total cost of the estimator. For BR-SNIS and SNIS this is always M. For Unbiased-PIMH in the **Soft** framework it is larger than M. In the **Hard** framework it is smaller than M.

Algorithm 6 Unbiased-PIMH

Data: $N \geq 0$

$e_1, \mathrm{lwav}_1 \leftarrow \mathrm{SNIS}(N)$; `/* SNIS also returning the average log weights */`

$e_2, \mathrm{lwav}_2 \leftarrow \mathrm{SNIS}(N)$ **if** $\mathrm{lwav}_1 < \mathrm{lwav}_2$ **then**

  $\mathrm{swap}(e_1, \mathrm{lwav}_1; e_2, \mathrm{lwav}_2)$

$u = \log \mathrm{rand}()$ **if** $u < \mathrm{lwav}_1$ *and* $u < \mathrm{lwav}_2$ **then**

  $\tau = 1$

$t \leftarrow 1$ $\tau = \infty$ **while** $\tau = \infty$ **do**

  $e_1 = e_1 + (e_1 - e_2)$ $e_p, \mathrm{lwav}_p = \mathrm{SNIS}(N)$ $t = t + 1$ $u = \log \mathrm{rand}()$; **if** $u < \mathrm{lwav}_p - \mathrm{lwav}_1$ **then**

    $e_1, \mathrm{lwav}_1 = e_p, \mathrm{lwav}_p$

  if $u < \mathrm{lwav}_p - \mathrm{lwav}_1$ **then**

    $e_2, \mathrm{lwav}_1 = e_p, \mathrm{lwav}_p$

  if $u < \mathrm{lwav}_1$ *and* $u < \mathrm{lwav}_2$ **then**

    $\tau = t$

We have compared both estimators in two different frameworks (**Hard** and **Soft**) with three different budgets $M = 2^{16}$ (tables A.1 and A.4), $M = 2^{12}$ (tables A.2 and A.5) and $M = 2^9$ (tables A.3 and A.6).

N	k	algorithm	Bias	std	average M
4096		SNIS	-0.0365	0.1946	4096.0
65	64	BR-SNIS	-0.0314	0.2211	4096.0
129	32	BR-SNIS	-0.0358	0.2214	4096.0
257	16	BR-SNIS	-0.0281	0.2282	4096.0
513	8	BR-SNIS	-0.0296	0.2351	4096.0
1024		Unbiased-PIMH	0.0587	0.6073	5388.0
512		Unbiased-PIMH	0.0678	0.8086	5027.5
256		Unbiased-PIMH	0.1258	1.1492	4730.0
128		Unbiased-PIMH	0.2364	1.9521	4629.6

Table A.2: $M = 2^{12}$ *in the* ***Soft*** *framework.*

N	k	algorithm	Bias	std	average M
512		SNIS	-0.1458	0.2420	512.0
65	8	BR-SNIS	-0.1537	0.2468	512.0
129	4	BR-SNIS	-0.1543	0.2444	512.0
257	2	BR-SNIS	-0.1426	0.2600	512.0
128		Unbiased-PIMH	-0.0048	1.3924	841.5
64		Unbiased-PIMH	0.1997	2.5677	796.4
32		Unbiased-PIMH	0.2365	4.1642	708.1
16		Unbiased-PIMH	0.3670	5.1533	685.3

Table A.3: $M = 2^{9}$ *in the* ***Soft*** *framework.*

N	k	algorithm	Bias	std	average M	Fails
65536		SNIS	-0.0029	0.0605	65536.0	
65	1024	BR-SNIS	-0.0006	0.0650	65536.0	
129	512	BR-SNIS	-0.0023	0.0645	65536.0	
257	256	BR-SNIS	-0.0024	0.0657	65536.0	
513	128	BR-SNIS	0.0000	0.0693	65536.0	
16384		Unbiased-PIMH	-0.0028	0.0885	57520.0	7
8192		Unbiased-PIMH	-0.0008	0.1029	59264.0	0
4096		Unbiased-PIMH	-0.0014	0.1026	61956.0	0
2048		Unbiased-PIMH	0.0008	0.1106	63244.0	0

Table A.4: $M = 2^{16}$ *in the* ***Hard*** *framework.*

N	k	algorithm	Bias	std	average M	Fails
4096		SNIS	-0.0365	0.1946	4096.0	
65	64	BR-SNIS	-0.0252	0.2270	4096.0	
129	32	BR-SNIS	-0.0296	0.2221	4096.0	
257	16	BR-SNIS	-0.0338	0.2218	4096.0	
513	8	BR-SNIS	-0.0486	0.2243	4096.0	
1024		Unbiased-PIMH	-0.0901	0.2353	2922.0	103
512		Unbiased-PIMH	-0.0833	0.3368	3343.0	24
256		Unbiased-PIMH	-0.0547	0.4815	3554.8	9
128		Unbiased-PIMH	-0.0634	0.4433	3683.1	4

Table A.5: $M = 2^{12}$ *in the* ***Hard*** *framework.*

N	k	algorithm	Bias	std	average M	Fails
512		SNIS	-0.1458	0.2420	512.0	
65	8	BR-SNIS	-0.1376	0.2636	512.0	
129	4	BR-SNIS	-0.1456	0.2565	512.0	
257	2	BR-SNIS	-0.1358	0.2585	512.0	
128		Unbiased-PIMH	-0.1962	0.2200	306.9	210
64		Unbiased-PIMH	-0.1947	0.3200	367.8	73
32		Unbiased-PIMH	-0.1999	0.4001	398.0	36
16		Unbiased-PIMH	-0.2057	0.7366	423.2	16

*Table A.6: $M = 2^9$ in the **Hard** framework.*

We observed that in general the BR-SNIS estimator has smaller standard deviation, with the difference of standard deviation being important for the smaller budgets (3 times less for $M = 2^{12}$ and 10 times less for $M = 2^9$ in the **Soft** framework).

For the **Hard** framework, we can see that the empirical bias of BR-SNIS is always at most equal to the empirical bias of Unbiased-PIMH . For the **Soft** framework, we observed that for $M = 2^{16}$ that both methods have similar performance, with BR-SNIS having negligible bias in this setting. For $M = 2^{12}$ and $M = 2^9$, BR-SNIS has in general a smaller empirical biais and the standard deviation of Unbiased-PIMH is considerably higher.

A.3.2 Bayesian Logistic regression

The importance distribution used in the Bayesian logistic regression example is given by the mean-field variational distribution Blei et al. (2017). More precisely, given the target π given in Section 2.3, the proposal λ is a Gaussian distribution with mean μ and diagonal covariance $\mathrm{diag}(\sigma)$, where μ, σ are learnt by maximization of the Evidence Lower Bound (ELBO):

$$\mathcal{L}(\mu, \sigma) = \int \log(\pi(\theta)/\lambda(\theta))\lambda(\theta)\mathrm{d}\theta. \tag{A.88}$$

In both Figures A.3 and 2.3, the optimal k for a given budget M was chosen by grid search over all the factors of M. The final settings are shown in Table A.7.

A.3.3 Importance Weighted Auto-Encoders

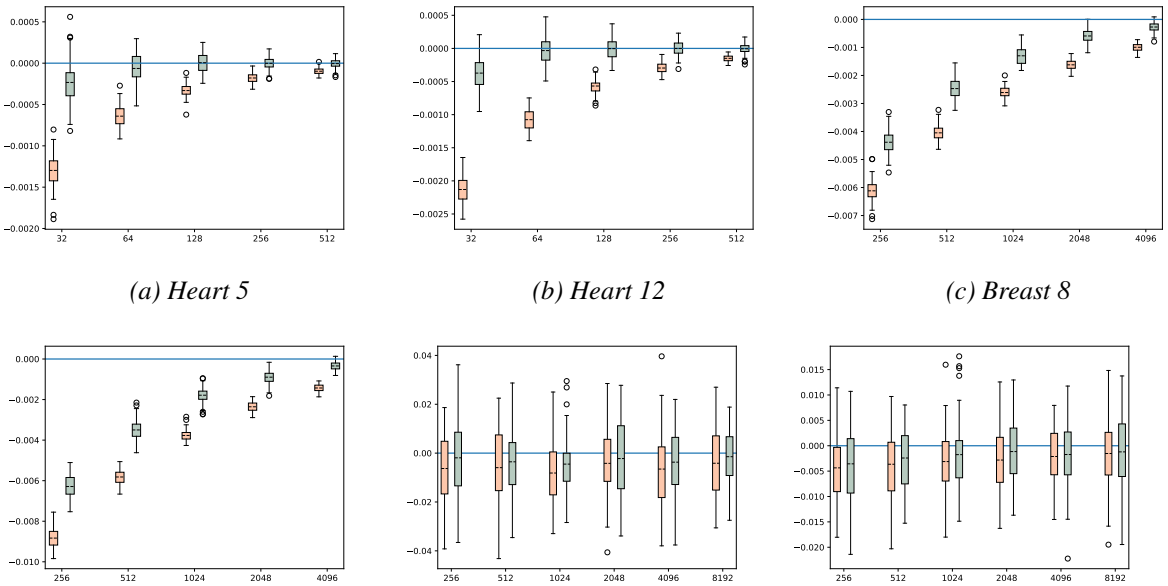
We trained each network for a total of 100 epochs, using 512 batch samples for the gradient calculations, with learning rate equals 10^{-4}. For IWAE and BR-IWAE , 64 samples were used for estimating the gradient. For BR-IWAE, we used $k = 8$. The architecture used is described in table A.8 where by conv layer we mean a convolutional layer followed by batch norm and the ReLU activation function. The train ELBO for each latent dimension is shown in Figure A.4. For the log likelihood comparison in Table 2.2, we use SNIS with the variational posterior as importance distribution and a total of $2 \cdot 10^3$ samples for a subset of 3232 samples from the validation set. Therefore, the estimation of the log likelihood is:

$$\hat{\mathcal{L}} = T^{-1} \sum_{j=1}^{T} \sum_{i=1}^{M} \omega_{\theta,\phi,x_j} \log p_\theta(x_j \mid z_i^j) \tag{A.89}$$

with $\omega_{\theta,\phi,x}(z) = p_\theta(x)/q_\phi(z \mid x)$ where z_i^j is sampled from $q_\phi(\cdot \mid x_j)$.

Dataset	component	M	k_{max}	N
breast	8	256	4	65
breast	8	512	8	65
breast	8	1024	16	65
breast	8	2048	16	129
breast	8	4096	64	65
breast	11	256	4	65
breast	11	512	8	65
breast	11	1024	16	65
breast	11	2048	32	65
breast	11	4096	64	65
breast	14	256	4	65
breast	14	512	8	65
breast	14	1024	16	65
breast	14	2048	32	65
breast	14	4096	64	65
heart	5	32	4	9
heart	5	64	8	9
heart	5	128	8	17
heart	5	256	32	9
heart	5	512	4	129
heart	8	32	4	9
heart	8	64	8	9
heart	8	128	8	17
heart	8	256	16	17
heart	8	512	32	17
heart	12	32	4	9
heart	12	64	8	9
heart	12	128	16	9
heart	12	256	4	65
heart	12	512	32	17
covertype	6	512	4	129
covertype	6	1024	8	129
covertype	6	2048	16	129
covertype	6	4096	2	2049
covertype	6	8192	4	2049
covertype	17	512	2	257
covertype	17	1024	2	513
covertype	17	2048	2	1025
covertype	17	4096	2	2049
covertype	17	8192	4	2049
covertype	23	512	2	257
covertype	23	1024	2	513
covertype	23	2048	4	513
covertype	23	4096	16	257
covertype	23	8192	32	257

Table A.7: Optimal configurations for Figures A.3 and 2.3



(a) Heart 5 *(b) Heart 12* *(c) Breast 8*

(d) Breast 14 *(e) Covertype 17* *(f) Covertype 23*

Figure A.3: Visualisation of the distribution of the bias for the Heart Failure and Breast cancer dataset for other components of θ

A.3.4 Resources

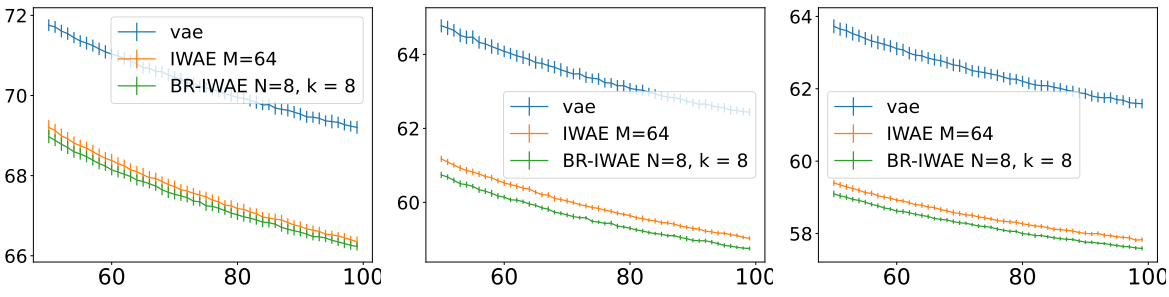
All the simulations were done using a server with the following configuration:

- GPUs: two Tesla V100-PCIE (32Gb RAM)
- CPU: 71 Intel(R) Xeon(R) Gold 6154 CPU @ 3.00GHz
- RAM: 377Gb

locally hosted. We estimate the total number of computing hours for the results presented in this paper to be inferior to 200 hours of GPU usage (All the calculations were done in the GPU).

Name	kernel size	stride	padding	out channels
Encoder conv 1	3	1	1	8
Encoder conv 2	3	1	1	16
Encoder conv 3	3	1	1	32
Encoder MaxPool2d 1	2	2	0	
Encoder conv 4	3	1	1	64
Encoder conv 5	3	1	1	32
Encoder MaxPool2d 2	2	2	0	
Encoder Linear + ReLU				2048
Encoder Linear				$2 * d$
Decoder Linear				$32 * 7 * 7$
Decoder conv transpose 1	2	1	0	64
Decoder conv transpose 2	2	1	1	128
Decoder conv transpose 3	3	2	1 (output padding = 1)	64
Decoder conv transpose 4	3	2	1 (output padding = 1)	32
Decoder conv transpose 5	2	1	0	16
Decoder final convolutional layer	2	1	0	1
Sigmoid activation				

Table A.8: Convolutional neural network architecture.



(a) Dimension 10 *(b) Dimension 20* *(c) Dimension 40*

Figure A.4: Per epoch training loss (ELBO) for the last 40 epochs. Confidence intervals are calculated as $1.96\sigma/\sqrt{n}$ over 10 ($n = 10$) different seeds.

Appendix B

Appendix of Chapter 3

B.1 Additional numerical results

B.1.1 LGSSM

B.1 and B.2 display the matrices A, B, $RR^\intercal$, and $SS^\intercal$ used for all experiments in the LGSSM model context. In B.1a,B.2a,B.3a we display boxplots of bias estimates, where each estimate is obtained by averaging 10^4 independent runs of the corresponding algorithm and each box is based on 10^3 replications of this bias estimator. The `PARIS` is compared to the `PPG` for different algorithmic configurations (N, k, k_0) and for different computational budgets $C = kN$ of sizes 10^3 (B.1), 2.5×10^3 (B.2), and 5×10^3 (B.3). Each experiment is carried through for each of the different designs $k_0 = \lfloor 2^{-1}k \rfloor$, $k_0 = \lfloor (3/4)C/N \rfloor$, and $k_0 = k - 1$ of the burn-in.

/	1	2	3	4	5
1	-0.4193	0.00182	0.00183	0.00184	0.00185
2	0.2145	0.63952	0.63953	0.63954	0.63955
3	0.3449	0.60202	0.60203	0.60204	0.60205
4	0.2572	-0.26932	-0.26933	-0.26934	-0.26935
5	0.7505	-0.36332	-0.36333	-0.36334	-0.36335

/	1	2	3	4	5
1	-0.2078	0.27752	0.27753	0.27754	0.27755
2	0.0984	0.45172	0.45173	0.45174	0.45175
3	0.7050	-0.04502	-0.04503	-0.04504	-0.04505
4	0.1684	-0.15152	-0.15153	-0.15154	-0.15155
5	-0.0320	0.50612	0.50613	0.50614	0.50615

Table B.1: The A (left) and B (right) matrices in the LGSSM.

/	1	2	3	4	5
1	0.0026	-0.00062	-0.00063	-0.00064	-0.00065
2	-0.0004	0.00122	0.00123	0.00124	0.00125
3	-0.0001	-0.00062	-0.00063	-0.00064	-0.00065
4	0.0007	0.00012	0.00013	0.00014	0.00015
5	-0.0006	0.00282	0.00283	0.00284	0.00285

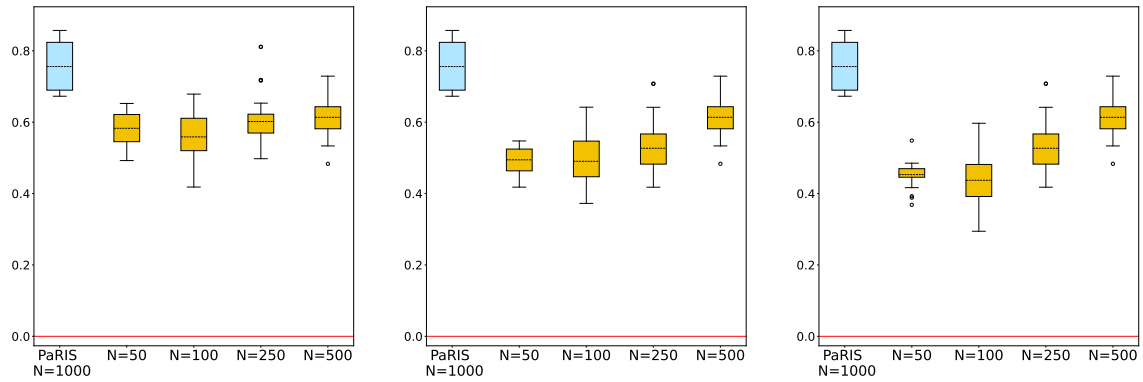
/	1	2	3	4	5
1	0.0157	-0.00072	-0.00073	-0.00074	-0.00075
2	0.0014	0.00072	0.00073	0.00074	0.00075
3	-0.0027	0.00592	0.00593	0.00594	0.00595
4	0.0064	-0.01052	-0.01053	-0.01054	-0.01055
5	-0.0007	0.02072	0.02073	0.02074	0.02075

Table B.2: The covariance matrices $RR^\intercal$ (left) and $SS^\intercal$ (right) for the state and measurement noises, respectively, in the LGSSM.

B.1.2 Stochastic volatility

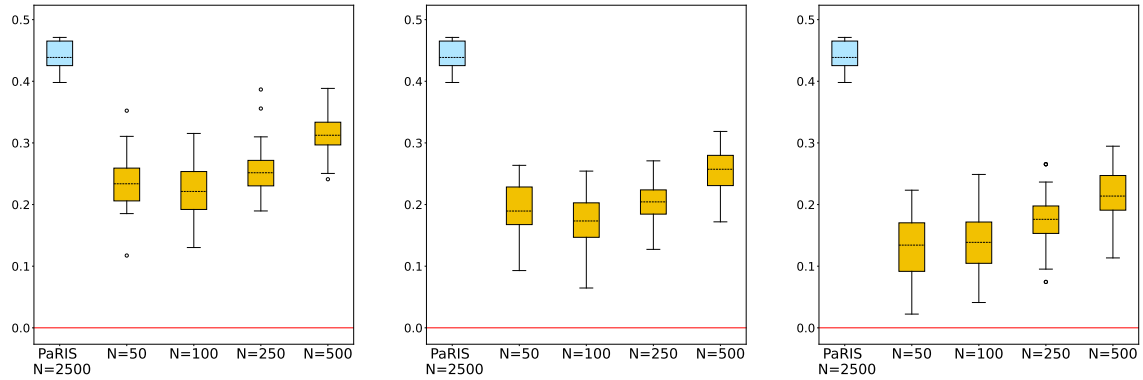
In this section we repeat the same experiments in B.1.1 in the context of the StoVol model described in 3.5. B.4–B.6 display boxplots of bias estimates for the `PARIS` and the `PPG` for different algorithmic configurations (N, k, k_0) and different computational budgets $C = kN$ of sizes 10^2 (B.4), 5×10^2 (B.5), and 10^3 (B.6). The bias of each algorithm is estimated by averaging 10^3 independent runs of the same, and each box is based on 10^3 independent replications of this bias estimator. Again, in each plot, the `PARIS` and `PPG` share the same computational budget (regardless configuration of the `PPG`).

Choice of (N, k, k_0). Designing the configuration (N, k, k_0) is challenging, since the upper bound $\kappa_{N,t}$ on the mixing rate is known to be conservative. As clear from B.4–B.6, the best configuration also



(a) $k_0 = \lfloor 2^{-1}C/N \rfloor$ *(b)* $k_0 = \lfloor (3/4)C/N \rfloor$ *(c)* $k_0 = k - 1$

Figure B.1: PARIS and PPG outputs for the LGSSM with $C = 10^3$ *and different designs of the burn-in* k_0.



(a) $k_0 = \lfloor 2^{-1}C/N \rfloor$ *(b)* $k_0 = \lfloor (3/4)C/N \rfloor$ *(c)* $k_0 = k - 1$

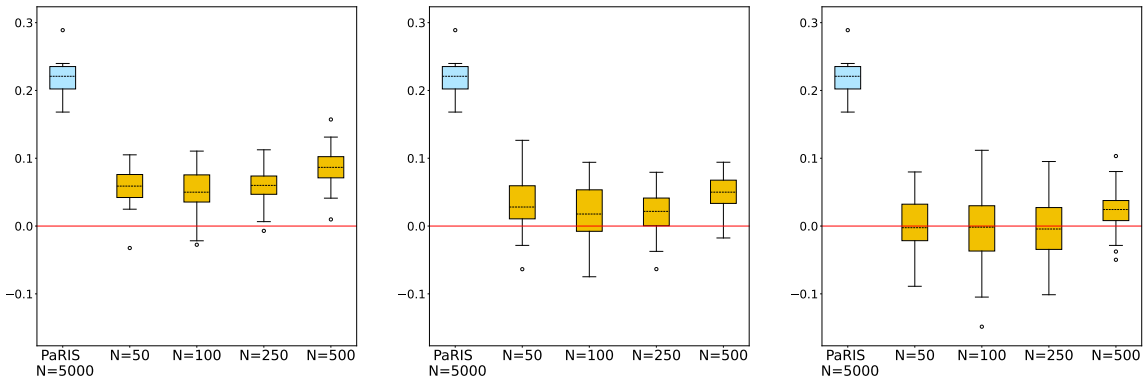
Figure B.2: PARIS and PPG outputs for the LGSSM with $C = 2.5 \times 10^3$ *and different designs of the burn-in* k_0.

depends on C; indeed, we see that for a smaller budget it is better to let the particle sample size N be large. Nevertheless, for more generous budgets it seems to be better to use a large number k of iterations at the expense of N.

Concerning the burn-in parameter k_0, the choice depends mainly on the bias–variance trade-off. In applications where minimising the bias is important one would choose $k_0 = k - 1$, which gives the smallest possible bias. Otherwise, a trade-off that provides an improvement in bias at the cost of an increase in MSE over the PARIS by only a factor of 2 is to choose $k_0 = \lfloor k/2 \rfloor$; recall the discussion in 3.4.2.

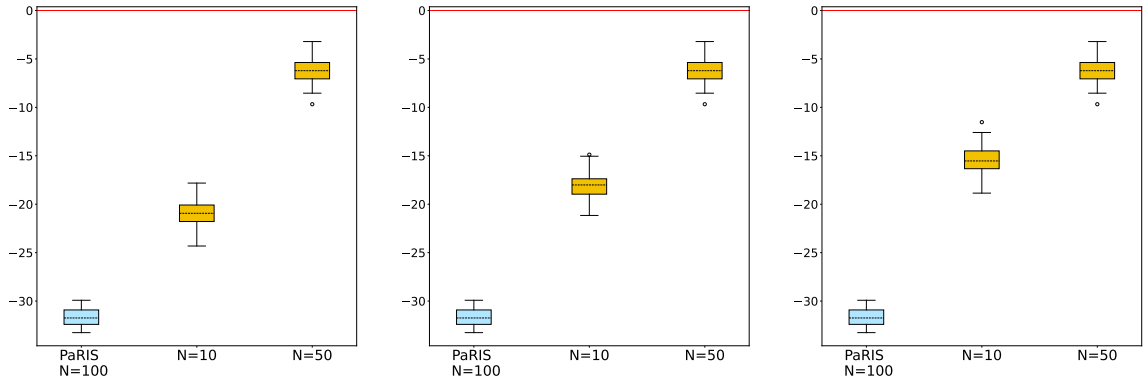
B.1.2.1 Comparison with the Rhee–Glynn-type estimator of Jacob et al. (2020a)

We now compare the proposed PPG estimator with the unbiased Rhee–Glynn-type smoothing estimator $\mathrm{H}_{k_0:k,N}$ defined in (Jacob et al., 2020a, Eq. 2), where the parameter k_0 is the burn-in phase length, k the minimum number of Gibbs iterations, and N the number of particles used in the coupled conditional particle filter. This estimator is based on the *coupled conditional particle filter* with ancestor sampling proposed in Jacob et al. (2020a); see 10 for details. Since the number of particles used in the algorithm is itself a random variable, we first perform 3×10^3 independent runs of the same and report the average



(a) $k_0 = \lfloor 2^{-1}C/N \rfloor$ *(b)* $k_0 = \lfloor (3/4)C/N \rfloor$ *(c)* $k_0 = k - 1$

Figure B.3: `PARIS` and `PPG` outputs for the LGSSM with $C = 5 \times 10^3$ *and different designs of the burn-in* k_0.



(a) $k_0 = \lfloor 2^{-1}C/N \rfloor$ *(b)* $k_0 = \lfloor (3/4)C/N \rfloor$ *(c)* $k_0 = k - 1$

Figure B.4: `PARIS` and `PPG` outputs for the stovol model with $C = 10^2$ *and different designs on the burn-in* k_0.

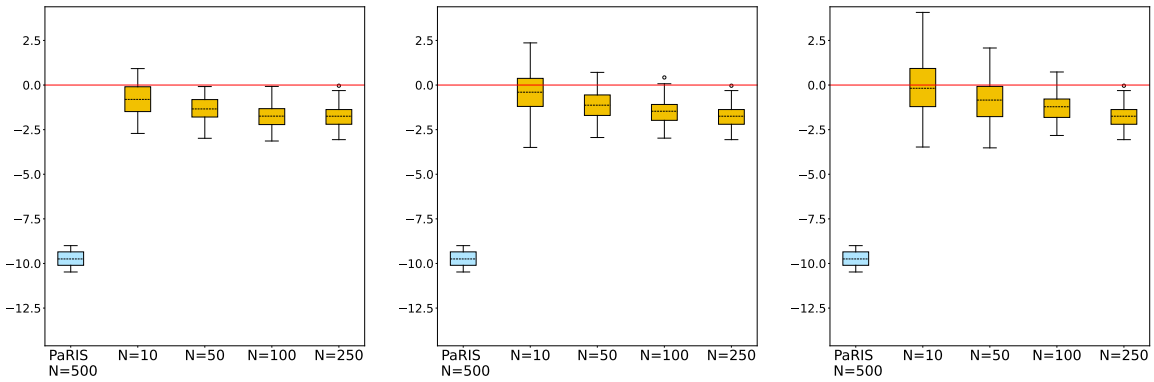
meeting time (*i.e.*, number of iterations of 10 until the conditional paths $\zeta_{0:t}$ and $\zeta'_{0:t}$ become identical) for three different choices of the hyperparameters in B.3. We deduce from B.3 that the average total number

N	k_0	k	Meeting time
100	5	10	30.4
250	2	4	12.6
500	1	2	7.1

Table B.3: Coupled conditional particle filter meeting times for three different configurations with $Nk = 10^3$.

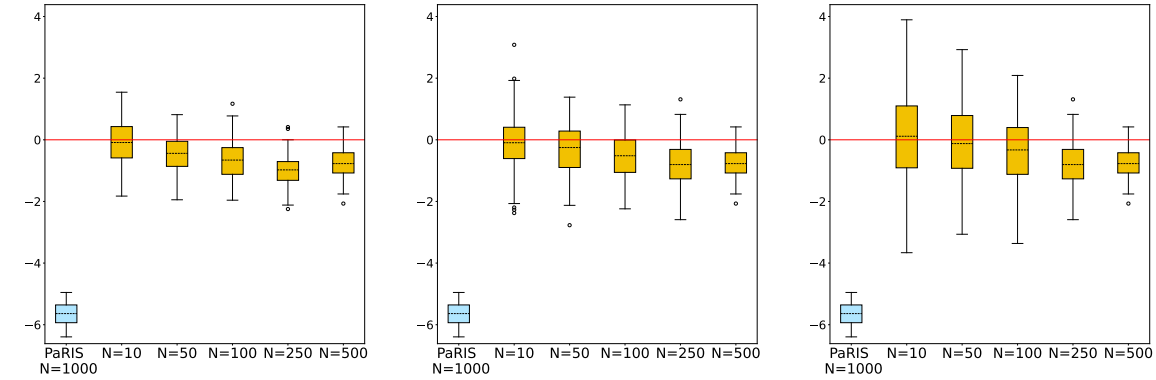
of particles generated is about 3×10^3. Therefore, we compare the Rhee–Glynn estimator induced by the coupled conditional particle filter with the `PPG` estimator with $(N, k_0, k) = (10, 150, 300)$. B.7 shows histograms of estimates produced using the Rhee–Glynn-type procedure, for the three different configurations, along with histograms of the estimates produced by the `PPG`. Each histogram is based on 3×10^3 independent replications. We find that the variance and empirical bias of the Rhee–Glynn-type estimator is about 10 and 20 times larger, respectively, than for the `PPG` for the same computational effort.

Another way of obtaining Rhee–Glynn-type smoothing estimator would be to consider the coupling of the conditional backward sampling particle filter, as proposed in Lee et al. (2020). In the case of the bootstrap



(a) $k_0 = \lfloor 2^{-1}C/N \rfloor$ *(b)* $k_0 = \lfloor (3/4)C/N \rfloor$ *(c)* $k_0 = k - 1$

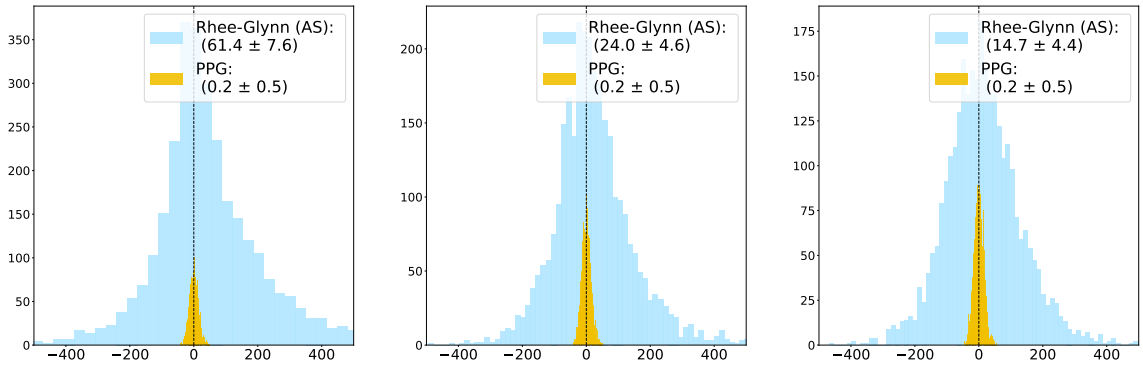
Figure B.5: `PARIS` and `PPG` outputs for the stovol model with $C = 5 \times 10^2$ *and different designs of the burn-in* k_0.



(a) $k_0 = \lfloor 2^{-1}C/N \rfloor$ *(b)* $k_0 = \lfloor (3/4)C/N \rfloor$ *(c)* $k_0 = k - 1$

Figure B.6: `PARIS` and `PPG` outputs for the stovol model with $C = 10^3$ *and different designs of the burn-in* k_0.

particle filter, the conditional particle filter with backward sampling is probabilistically equivalent to the conditional particle filter with ancestor sampling. Furthermore, (Lee et al., 2020, Section 7) also show that for $t = 10^3$, both the conditional particle filter with backward sampling and the conditional particle filter with ancestor sampling have similar performance. Thus, we expect the results in this section to translate to the estimators proposed in Lee et al. (2020).



(a) $(N, k) = (100, 10)$ *(b)* $(N, k) = (250, 4)$ *(c)* $(N, k) = (500, 2)$

Figure B.7: Histograms of estimates produced using the Rhee–Glynn-type smoothing estimator of Jacob et al. (2020a) for three different configurations and the PPG *estimator with* $(N, k_0, k) = (10, 150, 300)$. *Each box is based on* 3000 *independent replications. The plot also provides the corresponding* 95% *coverage asymptotic confidence intervals.*

B.2 Algorithms

The following section provides pseudocode for the algorithms discussed in 3.3, namely: the original PARIS algorithm (7) proposed in Olsson and Westerborn (2017), the conditional PARIS update (8), and the PPG (9). In addition, we provide a pseudocode for the coupled conditional conditional particle filter with *ancestor sampling* (10), being the key ingredient of the unbiased Rhee–Glynn-type estimator proposed in Jacob et al. (2020a) against which the PPG is benchmarked in B.1.2.1. Note that the conditional PARIS update described in 8 differs somewhat from that described in 3.3 in the way the underlying conditional dual process $\{\boldsymbol{\xi}_m\}_{m\in\mathbb{N}}$ is propagated; more precisely, in 8, each conditional dual process update $\boldsymbol{\xi}_{m+1} \sim \boldsymbol{M}_m\langle\zeta_{m+1}\rangle(\boldsymbol{\xi}_m, \cdot)$, where the value of ζ_{m+1} is inserted into a randomly chosen position in $\boldsymbol{\xi}_{m+1}$ (whereas the remaining elements of $\boldsymbol{\xi}_{m+1}$ are sampled independently from $\Phi_m(\mu(\boldsymbol{\xi}_m))$) is replaced by deterministic assignment of ζ_{m+1} to ξ^N_{m+1}. Of course, this change has no impact as long as we are interested in integrating functions that are permutation invariant with respect to the produced many-body systems, which is the case throughout our work. Still, as this derandomization technique simplifies somewhat the implementation of the PPG, we have chosen to include it in our pseudocode.

Algorithm 7 One update of the PARIS.

Data: $\{(\xi^i_t, \beta^i_t)\}_{i=1}^N$
Result: $\{(\xi^i_{t+1}, \beta^i_{t+1})\}_{i=1}^N$
for $i \leftarrow 1$ **to** N **do**
  draw $I^i_{t+1} \sim \mathsf{cat}(\{g_t(\xi^\ell_t)\}_{\ell=1}^N)$ draw $\xi^i_{t+1} \sim M_t(\xi^{I^i_{t+1}}_t, \cdot)$ **for** $j \leftarrow 1$ **to** M **do**
    draw $J^{(i,j)}_{t+1} \sim \mathsf{cat}(\{q_t(\xi^\ell_t, \xi^i_{t+1})\}_{\ell=1}^N)$
  set $\beta^i_{t+1} \leftarrow \frac{1}{M}\sum_{j=1}^M \left(\beta_t^{J^{(i,j)}_{t+1}} + \tilde{h}_t(\xi_t^{J^{(i,j)}_{t+1}}, \xi^i_{t+1})\right)$

Algorithm 8 One conditional PARIS update, expressed in a short form as "$\boldsymbol{v}_{t+1} \leftarrow \mathsf{CondPaRIS}(\boldsymbol{v}_t, \zeta_{t+1})$".

Data: $\boldsymbol{v}_t, \zeta_{t+1}$
Result: $\boldsymbol{v}_{t+1}$
for $i \leftarrow 1$ **to** $N-1$ **do**
  draw $I^i_{m+1} \sim \mathsf{cat}(\{g_m(\xi^\ell_{m|m})\}_{\ell=1}^N)$ draw $\xi^i_{m+1|m+1} \sim M_m(\xi^{I^i_{m+1}}_{m|m}, \cdot)$
set $\xi^N_{m+1|m+1} \leftarrow \zeta_{m+1}$ **for** $i \leftarrow 1$ **to** N **do**
  for $j \leftarrow 1$ **to** M **do**
    draw $J^{(i,j)}_{m+1} \sim \mathsf{cat}(\{q_m(\xi^\ell_{m|m}, \xi^i_{m+1|m+1})\}_{\ell=1}^N)$
  set $\beta^i_{m+1} \leftarrow \frac{1}{M}\sum_{j=1}^M \left(\beta_m^{J^{(i,j)}_{m+1}} + \tilde{h}_m(\xi_{m|m}^{J^{(i,j)}_{m+1}}, \xi^i_{m+1|m+1})\right)$ set $\xi^i_{0:m+1|m+1} \leftarrow (\xi^{J^{(i,1)}_{m+1}}_{0:m|m}, \xi^i_{m+1|m+1})$
set $\boldsymbol{v}_{t+1} \leftarrow ((\xi^1_{0:t+1|t+1}, \beta^1_{t+1}), \ldots, (\xi^N_{0:t+1|t+1}, \beta^N_{t+1}))$

Coupling algorithms. 10 provides a more detailed description of (the predictive variant of) the coupled conditional particle filter proposed in (Jacob et al., 2020a, Algorithm 1), and we focus here on the version of this algorithm where the iteratively produced particle paths underlying the resulting estimator are generated by means of ancestor sampling Lindsten et al. (2014a). If $\{\omega_\ell\}_{\ell=1}^N$ and $\{\omega'_\ell\}_{\ell=1}^N$ are possibly unnormalized event probabilities, we denote by $\mathsf{M}(\{\omega_\ell\}_{\ell=1}^N, \{\omega'_\ell\}_{\ell=1}^N)$ the *maximal coupling* between the distributions $\mathsf{cat}(\{\omega_\ell\}_{\ell=1}^N)$ and $\mathsf{cat}(\{\omega'_\ell\}_{\ell=1}^N)$. In our implementations, we used the maximum coupling

Algorithm 9 One iteration of the Parisian particle Gibbs (PPG)

Data: $\zeta_{0:t}$
Result: $\boldsymbol{\upsilon}_t$, $\zeta'_{0:t}$
draw $(\xi^1_{0|0}, \ldots, \xi^{N-1}_{0|0}) \sim \eta_0^{\otimes(N-1)}$ set $\xi^N_{0|0} \leftarrow \zeta_0$ set $\boldsymbol{\beta}_0 \leftarrow (0, \ldots, 0)$ **for** $m \leftarrow 0$ **to** $t-1$ **do**
  run $((\xi^1_{m+1|m+1}, \beta^1_{m+1}), \ldots, (\xi^N_{m+1|m+1}, \beta^N_{m+1})) \leftarrow \mathsf{CondPaRIS}((\xi^1_{m|m}, \beta^1_m), \ldots, (\xi^N_{m|m}, \beta^N_m), \zeta_{m+1})$
set $\boldsymbol{\upsilon}_t \leftarrow ((\xi^1_{t|t}, \beta^1_t), \ldots, (\xi^N_{t|t}, \beta^N_t))$ draw $J \sim \mathsf{cat}(\{1\}_{\ell=1}^N)$ set $\zeta'_{0:t} \leftarrow \xi^J_{0:t|t}$

given in (Jacob et al., 2020b, Algorithm 2). In order to couple two conditional particle filters, we assume, following (Jacob et al., 2020a, Algorithm 1), that for every $m \in \mathbb{N}$ we are able to simulate a random variable ε_m, defined on some measurable space $(\mathsf{S}_m, \mathcal{S}_m)$ and distributed according $\mu_m \in \mathsf{M}_1(\mathcal{S}_m)$, such that there exists some measurable function ϕ on $(\mathsf{X}_m \times \mathsf{S}_m, \mathcal{X}_m \otimes \mathcal{S}_m)$ such that for every $x_m \in \mathsf{X}_m$, $\mu_m \circ \phi_m^{-1}(x_m, \cdot)$ (the pushforward of μ_m through $\phi_m(x_m, \cdot)$) equals $M_m(x_m, \cdot)$.

Algorithm 10 Coupled conditional particle filters Jacob et al. (2020a).

Data: $\zeta_{0:t}$, $\tilde{\zeta}_{0:t}$
Result: $\zeta'_{0:t}$, $\tilde{\zeta}'_{0:t}$
set $(\xi^1_0, \ldots, \xi^{N-1}_0) \sim \eta_0^{\otimes(N-1)}$ set $(\tilde{\xi}^1_0, \ldots, \tilde{\xi}^{N-1}_0) \leftarrow (\xi^1_0, \ldots, \xi^{N-1}_0)$ set $(\xi^N_0, \tilde{\xi}^N_0) \leftarrow (\zeta_0, \tilde{\zeta}_0)$ **for** $m \leftarrow 0$ **to** $t-1$ **do**
  for $i \leftarrow 1$ **to** $N-1$ **do**
    draw $(I^i_{m+1}, \tilde{I}^i_{m+1}) \sim \mathsf{M}(\{g_m(\xi^\ell_m)\}_{\ell=1}^N, \{g_m(\tilde{\xi}^\ell_m)\}_{\ell=1}^N)$
  draw $(I^N_{m+1}, \tilde{I}^N_{m+1}) \sim \mathsf{M}(\{q_m(\xi^\ell_m, \zeta_{m+1})\}_{\ell=1}^N, \{q_m(\tilde{\xi}^\ell_m, \tilde{\zeta}_{m+1})\}_{\ell=1}^N)$ **for** $i \leftarrow 1$ **to** N **do**
    draw $\varepsilon_m \sim \mu_m$ set $(\xi^i_{m+1}, \tilde{\xi}^i_{m+1}) \leftarrow (\phi_m(\xi_m^{I^i_{m+1}}, \varepsilon_m), \phi_m(\tilde{\xi}_m^{\tilde{I}^i_{m+1}}, \varepsilon_m))$
draw $J_t \sim \mathsf{cat}(\{1\}_{\ell=1}^N)$ set $\tilde{J}_t \leftarrow J_t$ set $(\zeta_t, \tilde{\zeta}_t) \leftarrow (\xi_t^{J_t}, \tilde{\xi}_t^{\tilde{J}_t})$ **for** $m \leftarrow t-1$ **to** 0 **do**
  set $(J_m, \tilde{J}_m) \leftarrow (I^{J_{m+1}}_{m+1}, \tilde{I}^{\tilde{J}_{m+1}}_{m+1})$ set $(\zeta_m, \tilde{\zeta}_m) \leftarrow (\xi_m^{J_m}, \tilde{\xi}_m^{\tilde{J}_m})$

B.3 Additional proofs

B.3.1 Proof of 11

First, note that, by definitions (3.19) and (3.20),

$$\begin{aligned} H_t(\boldsymbol{x}_{0:t}) &\coloneqq \int \mathbb{S}_t(\boldsymbol{x}_{0:t}, \mathrm{d}\boldsymbol{y}_t)\, \mu(\boldsymbol{x}_{0:t|t}) h \\ &= \int \cdots \int \left(\frac{1}{N} \sum_{j_t=1}^{N} h(x_{0:t-1|t}^{j_t}, x_t^{j_t}) \right) \\ &\quad \times \prod_{m=0}^{t-1} \prod_{i_{m+1}=1}^{N} \int \sum_{j_m=1}^{N} \frac{q_m(x_m^{j_m}, x_{m+1}^{i_{m+1}})}{\sum_{j'_m=1}^{N} q_m(x_m^{j'_m}, x_{m+1}^{i_{m+1}})} \delta_{x_{0:m|m}^{j_m}}(\mathrm{d}x_{0:m|m+1}^{i_{m+1}}), \end{aligned}$$

where $x_{0:-1|0}^{i} = \emptyset$ for all $i \in [\![1, N]\!]$ by convention. We will show that for every $k \in [\![0, t]\!]$, $H_{k,t} \equiv H_t$, where

$$H_{k,t}(\boldsymbol{x}_{0:t}) \coloneqq \frac{1}{N} \sum_{j_t=1}^{N} \cdots \sum_{j_k=1}^{N} \prod_{\ell=k}^{t-1} \frac{q_\ell(x_\ell^{j_\ell}, x_{\ell+1}^{j_{\ell+1}})}{\sum_{j'_\ell=1}^{N} q_\ell(x_\ell^{j'_\ell}, x_{\ell+1}^{j_{\ell+1}})} a_{k,t}(\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{k-1}, x_k^{j_k}, \ldots, x_t^{j_t})$$

with

$$\begin{aligned} &a_{k,t}(\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{k-1}, x_k^{j_k}, \ldots, x_t^{j_t}) \\ &\quad = \int \prod_{m=0}^{k-1} \prod_{i_{m+1}=1}^{N} \sum_{j_m=1}^{N} \frac{q_m(x_m^{j_m}, x_{m+1}^{i_{m+1}})}{\sum_{j'_m=1}^{N} q_m(x_m^{j'_m}, x_{m+1}^{i_{m+1}})} \delta_{x_{0:m|m}^{j_m}}(\mathrm{d}x_{0:m|m+1}^{i_{m+1}}) h(x_{0:k-1|k}^{j_k}, x_k^{j_k}, \ldots, x_t^{j_t}). \end{aligned}$$

Since, by convention, $\prod_{\ell=t}^{t-1} \ldots = 1$, $H_{t,t}(\boldsymbol{x}_{0:t}) = N^{-1} \sum_{j_t=1}^{N} a_{t,t}(\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{t-1}, x_t^{j_t})$, and we note that $H_t \equiv H_{t,t}$. We now show that $H_{k,t} \equiv H_{k-1,t}$ for every $k \in [\![1, t]\!]$; for this purpose, note that

$$\begin{aligned} &a_{k,t}(\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{k-1}, x_k^{j_k}, \ldots, x_t^{j_t}) \\ &\qquad = \int \prod_{m=0}^{k-2} \prod_{i_{m+1}=1}^{N} \sum_{j_m=1}^{N} \frac{q_m(x_m^{j_m}, x_{m+1}^{i_{m+1}})}{\sum_{j'_m=1}^{N} q_m(x_m^{j'_m}, x_{m+1}^{i_{m+1}})} \delta_{x_{0:m|m}^{j_m}}(\mathrm{d}x_{0:m|m+1}^{i_{m+1}}) \\ &\quad \times \int \prod_{i_k=1}^{N} \sum_{j_{k-1}=1}^{N} \frac{q_{k-1}(x_{k-1}^{j_{k-1}}, x_k^{i_k})}{\sum_{j'_{k-1}=1}^{N} q_{k-1}(x_{k-1}^{j'_{k-1}}, x_k^{i_k})} \delta_{x_{0:k-1|k-1}^{j_{k-1}}}(\mathrm{d}x_{0:k-1|k}^{i_k})\, h(x_{0:k-1|k}^{j_k}, x_k^{j_k}, \ldots, x_t^{j_t}), \end{aligned}$$

and since $x_{0:k-1|k-1}^{j_{k-1}} = (x_{0:k-2|k-1}^{j_{k-1}}, x_{k-1}^{j_{k-1}})$, it holds that

$$\begin{aligned} &\int \prod_{i_k=1}^{N} \sum_{j_{k-1}=1}^{N} \frac{q_{k-1}(x_{k-1}^{j_{k-1}}, x_k^{i_k})}{\sum_{j'_{k-1}=1}^{N} q_{k-1}(x_{k-1}^{j'_{k-1}}, x_k^{i_k})} \delta_{x_{0:k-1|k-1}^{j_{k-1}}}(\mathrm{d}x_{0:k-1|k}^{i_k})\, h(x_{0:k-1|k}^{j_k}, x_k^{j_k}, \ldots, x_t^{j_t}) \\ &\qquad = \sum_{j_{k-1}=1}^{N} \frac{q_{k-1}(x_{k-1}^{j_{k-1}}, x_k^{j_k})}{\sum_{j'_{k-1}=1}^{N} q_{k-1}(x_{k-1}^{j'_{k-1}}, x_k^{j_k})} h(x_{0:k-2|k-1}^{j_{k-1}}, x_{k-1}^{j_{k-1}}, x_k^{j_k}, \ldots, x_t^{j_t}). \end{aligned}$$

Therefore, we obtain

$$
\begin{aligned}
&a_{k,t}(\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{k-1}, x_k^{j_k}, \ldots, x_t^{j_t}) \\
&\qquad = \int \prod_{m=0}^{k-2} \prod_{i_{m+1}=1}^{N} \sum_{j_m=1}^{N} \frac{q_m(x_m^{j_m}, x_{m+1}^{i_{m+1}})}{\sum_{j'_m=1}^{N} q_m(x_m^{j'_m}, x_{m+1}^{i_{m+1}})} \delta_{x_{0:m|m}^{j_m}}(\mathrm{d}x_{0:m|m+1}^{i_{m+1}}) \\
&\qquad\qquad \times \sum_{j_{k-1}=1}^{N} \frac{q_{k-1}(x_{k-1}^{j_{k-1}}, x_k^{j_k})}{\sum_{j'_{k-1}=1}^{N} q_{k-1}(x_{k-1}^{j'_{k-1}}, x_k^{j_k})} h(x_{0:k-2|k-1}^{j_{k-1}}, x_{k-1}^{j_{k-1}}, x_k^{j_k}, \ldots, x_t^{j_t}).
\end{aligned}
$$

Now, changing the order of summation with respect to j_{k-1} and integration on the right hand side of the previous display yields

$$
\begin{aligned}
&a_{k,t}(\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{k-1}, x_k^{j_k}, \ldots, x_t^{j_t}) \\
&\qquad = \sum_{j_{k-1}=1}^{N} \frac{q_{k-1}(x_{k-1}^{j_{k-1}}, x_k^{j_k})}{\sum_{j'_{k-1}=1}^{N} q_{k-1}(x_{k-1}^{j'_{k-1}}, x_k^{j_k})} a_{k-1,t}(\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{k-2}, x_{k-1}^{j_{k-1}}, \ldots, x_t^{j_t}).
\end{aligned}
$$

Thus,

$$
\begin{aligned}
&H_{k,t}(\boldsymbol{x}_{0:t}) \\
&= \frac{1}{N} \sum_{j_t=1}^{N} \cdots \sum_{j_k=1}^{N} \prod_{\ell=k}^{t-1} \frac{q_\ell(x_\ell^{j_\ell}, x_{\ell+1}^{j_{\ell+1}})}{\sum_{j'_\ell=1}^{N} q_\ell(x_\ell^{j'_\ell}, x_{\ell+1}^{j_{\ell+1}})} \\
&\quad \times \sum_{j_{k-1}=1}^{N} \frac{q_{k-1}(x_{k-1}^{j_{k-1}}, x_k^{j_k})}{\sum_{j'_{k-1}=1}^{N} q_{k-1}(x_{k-1}^{j'_{k-1}}, x_k^{j_k})} a_{k-1,t}(\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{k-2}, x_{k-1}^{j_{k-1}}, \ldots, x_t^{j_t}) \\
&= \frac{1}{N} \sum_{j_t=1}^{N} \cdots \sum_{j_{k-1}=1}^{N} \prod_{\ell=k-1}^{t-1} \frac{q_\ell(x_\ell^{j_\ell}, x_{\ell+1}^{j_{\ell+1}})}{\sum_{j'_\ell=1}^{N} q_\ell(x_\ell^{j'_\ell}, x_{\ell+1}^{j_{\ell+1}})} a_{k-1,t}(\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{k-2}, x_{k-1}^{j_{k-1}}, \ldots, x_t^{j_t}) \\
&= H_{k-1,t}(\boldsymbol{x}_{0:t}),
\end{aligned}
$$

which establishes the recursion. Therefore, $H_t \equiv H_{0,t}$ and we may now conclude the proof by noting that $\mathbb{B}_t h \equiv H_{0,t}$.

B.3.2 Proof of 15

In order to establish 15 we will prove the following more general result, of which 15 is a direct consequence.

Proposition 40. *For every $t \in \mathbb{N}$ and $M \in \mathbb{N}^*$ there exist $\mathsf{c}_t > 0$ and $\mathsf{d}_t > 0$ such that for every $N \in \mathbb{N}^*$, $z_{0:t} \in \mathsf{X}_{0:t}$, $(f_t, \tilde{f}_t) \in \mathsf{F}(\mathcal{X}_t)^2$, and $\varepsilon > 0$,*

$$
\begin{aligned}
&\int \mathbb{C}_t \mathbb{S}_t(z_{0:t}, \mathrm{d}\boldsymbol{b}_t) \\
&\qquad \times \mathbb{1}\left\{ \left| \frac{1}{N} \sum_{i=1}^{N} \{ b_t^i f_t(x_{t|t}^i) + \tilde{f}_t(x_{t|t}^i) \} - \eta_t\langle z_{0:t} \rangle (f_t B_t \langle z_{0:t-1} \rangle h_t + \tilde{f}_t) \right| \geq \varepsilon \right\} \\
&\qquad\qquad \leq \mathsf{c}_t \exp\left(-\frac{\mathsf{d}_t N \varepsilon^2}{2\kappa_t^2} \right),
\end{aligned}
$$

where

$$
\kappa_t \coloneqq \|f_t\|_\infty \sum_{m=0}^{t-1} \|\tilde{h}_m\|_\infty + \|\tilde{f}_t\|_\infty. \tag{B.1}
$$

To prove 40 we need the following technical lemma.

Lemma 41. *For every $t \in \mathbb{N}$, $(f_{t+1}, \tilde{f}_{t+1}) \in \mathsf{F}(\mathcal{X}_{t+1})^2$, $z_{0:t+1} \in \mathsf{X}_{0:t+1}$, and $N \in \mathbb{N}^*$,*

$$\begin{aligned}
&\gamma_{t+1}\langle z_{0:t+1}\rangle(f_{t+1}B_{t+1}\langle z_{0:t}\rangle h_{t+1} + \tilde{f}_{t+1}) \\
&\qquad = \left(1 - \frac{1}{N}\right)\gamma_t\langle z_{0:t}\rangle\{Q_t f_{t+1}B_t\langle z_{0:t-1}\rangle h_t + Q_t(\tilde{h}_t f_{t+1} + \tilde{f}_{t+1})\} \\
&\qquad\qquad + \frac{1}{N}\gamma_t\langle z_{0:t}\rangle g_t\left(f_{t+1}(z_{t+1})B_{t+1}\langle z_{0:t}\rangle h_{t+1}(z_{t+1}) + \tilde{f}_{t+1}(z_{t+1})\right).
\end{aligned}$$

Proof. Since 21 holds also for the Feynman–Kac model with a frozen path, we obtain

$$\begin{aligned}
&\gamma_{t+1}\langle z_{0:t+1}\rangle(f_{t+1}B_{t+1}\langle z_{0:t}\rangle h_{t+1} + \tilde{f}_{t+1}) \\
&\qquad = \gamma_t\langle z_{0:t}\rangle\{Q_t\langle z_{t+1}\rangle f_{t+1}B_t\langle z_{0:t}\rangle h_t + Q_t\langle z_{t+1}\rangle(\tilde{h}_t f_{t+1} + \tilde{f}_{t+1})\}.
\end{aligned}$$

Thus, the proof is concluded by noting that for every $x_t \in \mathsf{X}_t$ and $h \in \mathsf{F}(\mathcal{X}_{t:t+1})$,

$$Q_t\langle z_{t+1}\rangle h(x_t) = \left(1 - \frac{1}{N}\right)Q_t h(x_t) + \frac{1}{N}g(x_t)h(x_t, z_{t+1}).$$

□

Finally, before proceeding to the proof of 40, we introduce the law of the `PARIS` evolving conditionally on a frozen path $z = \{z_m\}_{m\in\mathbb{N}}$. Define, for $m \in \mathbb{N}$ and $z_{m+1} \in \mathsf{X}_{m+1}$,

$$\boldsymbol{P}_m\langle z_{m+1}\rangle : \mathbf{Y}_m \times \boldsymbol{\mathcal{Y}}_{m+1} \ni (\boldsymbol{y}_m, A) \mapsto \int \boldsymbol{M}_m\langle z_{m+1}\rangle(\boldsymbol{x}_{m|m}, \mathrm{d}\boldsymbol{x}_{m+1})\, \boldsymbol{S}_m(\boldsymbol{y}_m, \boldsymbol{x}_{m+1}, A).$$

For any given initial distribution $\boldsymbol{\psi}_0 \in \mathsf{M}_1(\boldsymbol{\mathcal{Y}}_0)$, let $\mathbb{P}^{\boldsymbol{P},z}_{\boldsymbol{\psi}_0}$ be the distribution of the canonical Markov chain induced by the Markov kernels $\{\boldsymbol{P}_m\langle z_{m+1}\rangle\}_{m\in\mathbb{N}}$ and the initial distribution $\boldsymbol{\psi}_0$. By abuse of notation we write $\mathbb{P}^{\boldsymbol{P},z}_{\boldsymbol{\eta}_0}$ instead of $\mathbb{P}^{\boldsymbol{P},z}_{\boldsymbol{\psi}_0[\boldsymbol{\eta}_0\langle z_0\rangle]}$, where the extension $\boldsymbol{\psi}_0[\boldsymbol{\eta}_0]$ is defined in 3.6.3.

Proof of 40. We proceed by forward induction over t. Let the σ-fields $\tilde{\mathcal{F}}_t$ and $\mathcal{F}_t$ be defined as in the proof of 13, but for the conditional `PARIS` dual process. Then, under the law $\mathbb{P}^{\boldsymbol{P},z}_{\boldsymbol{\eta}_0}$, reusing (3.43),

$$\begin{aligned}
&\mathbb{E}^{\boldsymbol{P},z}_{\boldsymbol{\eta}_0}\left[\beta^1_t f_t(\xi^1_t) + \tilde{f}_t(\xi^1_t) \mid \tilde{\mathcal{F}}_{t-1}\right] \\
&= \mathbb{E}^{\boldsymbol{P},z}_{\boldsymbol{\eta}_0}\left[\mathbb{E}^{\boldsymbol{P},z}_{\boldsymbol{\eta}_0}\left[\beta^1_t \mid \mathcal{F}_t\right] f_t(\xi^1_t) + \tilde{f}_t(\xi^1_t) \mid \tilde{\mathcal{F}}_{t-1}\right] \\
&= \mathbb{E}^{\boldsymbol{P},z}_{\boldsymbol{\eta}_0}\left[f_t(\xi^1_t)\sum_{\ell=1}^N \frac{q_{t-1}(\xi^\ell_{t-1}, \xi^1_t)}{\sum_{\ell'=1}^N q_{t-1}(\xi^{\ell'}_{t-1}, \xi^1_t)}\left(\beta^\ell_{t-1} + \tilde{h}_{t-1}(\xi^\ell_{t-1}, \xi^1_t)\right) + \tilde{f}_t(\xi^1_t) \mid \tilde{\mathcal{F}}_{t-1}\right].
\end{aligned}$$

Using (3.10), we get

$$\begin{aligned}
&\mathbb{E}^{\boldsymbol{P},z}_{\boldsymbol{\eta}_0}\left[\beta^1_t f_t(\xi^1_t) + \tilde{f}_t(\xi^1_t) \mid \tilde{\mathcal{F}}_{t-1}\right] \\
&\qquad = \left(1 - \frac{1}{N}\right)\frac{\sum_{\ell=1}^N\{\beta^\ell_{t-1}Q_{t-1}f_t(\xi^\ell_{t-1}) + Q_{t-1}(\tilde{h}_{t-1}f_t + \tilde{f}_t)(\xi^\ell_{t-1})\}}{\sum_{\ell'=1}^N g_{t-1}(\xi^{\ell'}_{t-1})} \\
&\qquad\quad + \frac{1}{N}\left(f_t(z_t)\sum_{\ell=1}^N \frac{q_{t-1}(\xi^\ell_{t-1}, z_t)}{\sum_{\ell'=1}^N q_{t-1}(\xi^{\ell'}_{t-1}, z_t)}\left(\beta^\ell_{t-1} + \tilde{h}_t(\xi^\ell_{t-1}, z_t)\right) + \tilde{f}_t(z_t)\right). \qquad \text{(B.2)}
\end{aligned}$$

In order to apply the induction hypothesis to each term on the right-hand side of the previous identity, note that

$$B_t\langle z_{0:t-1}\rangle h_t(z_t) = \frac{\eta_{t-1}\langle z_{0:t-1}\rangle[q_{t-1}(\cdot, z_t)\{B_{t-1}\langle z_{0:t-2}\rangle h_{t-1}(\cdot) + \tilde{h}_{t-1}(\cdot, z_t)\}]}{\eta_{t-1}\langle z_{0:t-1}\rangle[q_{t-1}(\cdot, z_t)]}.$$

Therefore, using 41 and noting that $\gamma_t\langle z_{0:t}\rangle \mathbb{1}_{\mathsf{X}_t} / \gamma_{t-1}\langle z_{0:t}\rangle \mathbb{1}_{\mathsf{X}_{t-1}} = \eta_{t-1}\langle z_{0:t-1}\rangle g_{t-1}$ yields

$$\eta_t\langle z_{0:t}\rangle(f_t B_t\langle z_{0:t-1}\rangle h_t + \tilde{f}_t) = \frac{1}{N}\left(f_t(z_t)B_t\langle z_{0:t-1}\rangle h_t(z_t) + \tilde{f}_t(z_t)\right) + \left(1-\frac{1}{N}\right)\frac{\eta_{t-1}\langle z_{0:t-1}\rangle\{Q_{t-1}f_t B_{t-1}\langle z_{0:t-2}\rangle h_t + Q_{t-1}(\tilde{h}_{t-1}f_t + \tilde{f}_t)\}}{\eta_{t-1}\langle z_{0:t-1}\rangle g_{t-1}}. \tag{B.3}$$

By combining (B.2) with (B.3), we decompose the error according to

$$\begin{aligned}
&\frac{1}{N}\sum_{i=1}^{N}\{\beta_t^i f_t(\xi_{t|t}^i) + \tilde{f}_t(\xi_{t|t}^i)\} - \eta_t\langle z_{0:t}\rangle(f_t B_t\langle z_{0:t-1}\rangle h_t + \tilde{f}_t) \\
&\quad= \frac{1}{N}\sum_{i=1}^{N}\{\beta_t^i f_t(\xi_{t|t}^i) + \tilde{f}_t(\xi_{t|t}^i)\} - \mathbb{E}_{\boldsymbol{\eta}_0}^{\boldsymbol{P},z}\left[\beta_t^1 f_t(\xi_t^1) + \tilde{f}_t(\xi_t^1) \mid \tilde{\mathcal{F}}_{t-1}\right] \\
&\qquad+ \mathbb{E}_{\boldsymbol{\eta}_0}^{\boldsymbol{P},z}\left[\beta_t^1 f_t(\xi_t^1) + \tilde{f}_t(\xi_t^1) \mid \tilde{\mathcal{F}}_{t-1}\right] - \eta_t\langle z_{0:t}\rangle(f_t B_t\langle z_{0:t-1}\rangle h_t + \tilde{f}_t) \\
&\quad= \mathrm{I}_N^{(1)} + \left(1-\frac{1}{N}\right)\mathrm{I}_N^{(2)} + \frac{1}{N}\mathrm{I}_N^{(3)},
\end{aligned} \tag{B.4}$$

where

$$\begin{aligned}
\mathrm{I}_N^{(1)} &:= \frac{1}{N}\sum_{i=1}^{N}\{\beta_t^i f_t(\xi_t^i) + \tilde{f}_t(\xi_t^i)\} - \mathbb{E}_{\boldsymbol{\eta}_0}^{\boldsymbol{P},z}\left[\beta_t^1 f_t(\xi_t^1) + \tilde{f}_t(\xi_t^1) \mid \tilde{\mathcal{F}}_{t-1}\right], \\
\mathrm{I}_N^{(2)} &:= \frac{\sum_{\ell=1}^{N}\{\beta_{t-1}^\ell Q_{t-1}f_t(\xi_{t-1}^\ell) + Q_{t-1}(\tilde{h}_{t-1}f_t + \tilde{f}_t)(\xi_{t-1}^\ell)\}}{\sum_{\ell'=1}^{N} g_{t-1}(\xi_{t-1}^{\ell'})} \\
&\qquad- \frac{\eta_{t-1}\langle z_{0:t-1}\rangle\{Q_{t-1}f_t B_t\langle z_{0:t-1}\rangle h_t + Q_{t-1}(\tilde{h}_{t-1}f_t + \tilde{f}_t)\}}{\eta_{t-1}\langle z_{0:t-1}\rangle g_{t-1}},
\end{aligned} \tag{B.5}$$

and

$$\begin{aligned}
\mathrm{I}_N^{(3)} &:= f_t(z_t)\sum_{\ell=1}^{N}\frac{q_{t-1}(\xi_{t-1}^\ell, z_t)}{\sum_{\ell'=1}^{N} q_{t-1}(\xi_{t-1}^{\ell'}, z_t)}\left(\beta_{t-1}^\ell + \tilde{h}_{t-1}(\xi_{t-1}^\ell, z_t)\right) \\
&\qquad- f_t(z_t)\frac{\eta_{t-1}\langle z_{0:t-1}\rangle[q_{t-1}(\cdot, z_t)\{B_{t-1}\langle z_{0:t-2}\rangle h_{t-1}(\cdot) + \tilde{h}_{t-1}(\cdot, z_t)\}]}{\eta_{t-1}\langle z_{0:t-1}\rangle[q_{t-1}(\cdot, z_t)]}.
\end{aligned} \tag{B.6}$$

The proof is now completed by treating the terms $\mathrm{I}_N^{(1)}$, $\mathrm{I}_N^{(2)}$, and $\mathrm{I}_N^{(3)}$ separately, using Hoeffding's inequality and its generalisation in (Douc et al., 2011, Lemma 4). Choose $\varepsilon > 0$; then, by Hoeffding's inequality,

$$\mathbb{P}_{\boldsymbol{\eta}_0}^{\boldsymbol{P},z}\left(|\,\mathrm{I}_N^{(1)}\,| \geq \varepsilon\right) \leq 2\exp\left(-\frac{1}{2}\frac{\varepsilon^2}{\kappa_t^2}N\right). \tag{B.7}$$

To treat $\mathrm{I}_N^{(2)}$, we apply the induction hypothesis to the numerator and denominator, each normalized by $1/N$, yielding, since $\|Q_{t-1}h\|_\infty \leq \bar{\tau}_{t-1}\|h\|_\infty$ for all $h \in \mathsf{F}(\mathcal{X}_{t-1}\otimes\mathcal{X}_t)$,

$$\begin{aligned}
&\mathbb{P}_{\boldsymbol{\eta}_0}^{\boldsymbol{P},z}\left(\left|\frac{1}{N}\sum_{\ell=1}^{N}\{\beta_{t-1}^\ell Q_{t-1}f_t(\xi_{t-1}^\ell) + Q_{t-1}(\tilde{h}_{t-1}f_t + \tilde{f}_t)(\xi_{t-1}^\ell)\}\right.\right. \\
&\qquad\left.\left.-\eta_{t-1}\langle z_{0:t-1}\rangle\{Q_{t-1}f_t B_t\langle z_{0:t-1}\rangle h_t + Q_{t-1}(\tilde{h}_{t-1}f_t + \tilde{f}_t)\}\right| \geq \varepsilon\right) \\
&\qquad\qquad\leq \mathsf{c}_{t-1}\exp\left(-\mathsf{d}_{t-1}\frac{\varepsilon^2}{\bar{\tau}_{t-1}^2\kappa_t^2}N\right)
\end{aligned}$$

and

$$\mathbb{P}_{\boldsymbol{\eta}_0}^{\boldsymbol{P},z}\left(\left|\frac{1}{N}\sum_{\ell=1}^{N} g_{t-1}(\xi_{t-1}^{\ell}) - \eta_{t-1}\langle z_{0:t-1}\rangle g_{t-1}\right| \geq \varepsilon\right) \leq \mathsf{c}_{t-1}\exp\left(-\mathsf{d}_{t-1}\frac{\varepsilon^2}{\bar{\tau}_{t-1}^2}N\right).$$

Combining the previous two bounds with the generalised Hoeffding inequality in (Douc et al., 2011, Lemma 4) yields, using also the bounds

$$\frac{\sum_{\ell=1}^{N}\{\beta_{t-1}^{\ell}Q_{t-1}f_t(\xi_{t-1}^{\ell}) + Q_{t-1}(\tilde{h}_{t-1}f_t + \tilde{f}_t)(\xi_{t-1}^{\ell})\}}{\sum_{\ell'=1}^{N} g_{t-1}(\xi_{t-1}^{\ell'})} \leq \kappa_t$$

and $\eta_{t-1}\langle z_{0:t-1}\rangle g_{t-1} \geq \underline{\tau}_{t-1}$, the inequality

$$\mathbb{P}_{\boldsymbol{\eta}_0}^{\boldsymbol{P},z}\left(|\,\mathrm{I}_N^{(2)}\,| \geq \varepsilon\right) \leq \mathsf{c}_{t-1}\exp\left(-\mathsf{d}_{t-1}\frac{\underline{\tau}_{t-1}^2\varepsilon^2}{\bar{\tau}_{t-1}^2\kappa_t^2}N\right). \tag{B.8}$$

The last term $\mathrm{I}_N^{(3)}$ is treated along similar lines; indeed, by the induction hypothesis, since $\|q_{t-1}\|_\infty \leq \bar{\tau}_{t-1}\bar{\sigma}_{t-1}$,

$$\begin{aligned}\mathbb{P}_{\boldsymbol{\eta}_0}^{\boldsymbol{P},z}\Bigg(\Bigg|\frac{1}{N}\sum_{\ell=1}^{N} q_{t-1}(\xi_{t-1}^{\ell}, z_t)\left(\beta_{t-1}^{\ell} + \tilde{h}_{t-1}(\xi_{t-1}^{\ell}, z_t)\right)\\ - \eta_{t-1}\langle z_{0:t-1}\rangle[q_{t-1}(\cdot, z_t)\{B_{t-1}\langle z_{0:t-1}\rangle h_{t-1}(\cdot) + \tilde{h}_{t-1}(\cdot, z_t)\}]\Bigg| \geq \varepsilon\Bigg)\\ \leq \mathsf{c}_{t-1}\exp\left(-\mathsf{d}_{t-1}\left(\frac{\varepsilon}{\bar{\tau}_{t-1}\bar{\sigma}_{t-1}\sum_{m=0}^{t-1}\|\tilde{h}_m\|_\infty}\right)^2 N\right)\end{aligned}$$

and

$$\begin{aligned}\mathbb{P}_{\boldsymbol{\eta}_0}^{\boldsymbol{P},z}\left(\left|\frac{1}{N}\sum_{\ell=1}^{N} q_{t-1}(\xi_{t-1}^{\ell}, z_t) - \eta_{t-1}\langle z_{0:t-1}\rangle[q_{t-1}(\cdot, z_t)]\right| \geq \varepsilon\right)\\ \leq \mathsf{c}_{t-1}\exp\left(-\mathsf{d}_{t-1}\left(\frac{\varepsilon}{\bar{\tau}_{t-1}\bar{\sigma}_{t-1}}\right)^2 N\right).\end{aligned}$$

Thus, since

$$\sum_{\ell=1}^{N}\frac{q_{t-1}(\xi_{t-1}^{\ell}, z_t)}{\sum_{\ell'=1}^{N} q_{t-1}(\xi_{t-1}^{\ell'}, z_t)}\left(\beta_{t-1}^{\ell} + \tilde{h}_{t-1}(\xi_{t-1}^{\ell}, z_t)\right) \leq \sum_{m=0}^{t-1}\|\tilde{h}_m\|_\infty$$

and $\eta_{t-1}\langle z_{0:t-1}\rangle[q_{t-1}(\cdot, z_t)] \geq \underline{\tau}_{t-1}$, the generalised Hoeffding inequality provides

$$\mathbb{P}_{\boldsymbol{\eta}_0}^{\boldsymbol{P},z}\left(|\,\mathrm{I}_N^{(3)}\,| \geq \varepsilon\right) \leq \mathsf{c}_{t-1}\exp\left(-\mathsf{d}_{t-1}\left(\frac{\underline{\tau}_{t-1}\varepsilon}{2\bar{\tau}_{t-1}\bar{\sigma}_{t-1}\|f_t\|_\infty\sum_{m=0}^{t-1}\|\tilde{h}_m\|_\infty}\right)^2 N\right). \tag{B.9}$$

Finally, combining the bounds (B.7–B.9) completes the proof. □

B.3.3 Proof of 16

The statement of 16 is implied by the following more general result, which we will prove below.

Proposition 42. *For every $t \in \mathbb{N}$, $M \in \mathbb{N}^*$, $N \in \mathbb{N}^*$, $z_{0:t} \in \mathsf{X}_{0:t}$, $(f_t, \tilde{f}_t) \in \mathsf{F}(\mathcal{X}_t)^2$, and $p \geq 2$, it holds that*

$$\int \mathbb{C}_t \mathbb{S}_t(z_{0:t}, \mathrm{d}\boldsymbol{b}_t) \left| \frac{1}{N} \sum_{i=1}^{N} \{b_t^i f_t(x_{t|t}^i) + \tilde{f}_t(x_{t|t}^i)\} - \eta_t\langle z_{0:t}\rangle (f_t B_t \langle z_{0:t-1}\rangle h_t + \tilde{f}_t) \right|^p \leq \mathsf{c}_t (p/\mathsf{d}_t)^{p/2} N^{-p/2} \kappa_t^p,$$

where $\mathsf{c}_t > 0$, $\mathsf{d}_t > 0$ and κ_t are defined in 40 and (B.1), respectively.

Before proving 42, we establish the following result.

Lemma 43. *Let X be an $\mathbb{R}^d$-valued random variable, defined on some probability space $(\Omega, \mathcal{F}, \mathbb{P})$, satisfying $\mathbb{P}(|X| \geq t) \leq c \exp(-t^2/(2\sigma^2))$ for every $t \geq 0$ and some $c > 0$ and $\sigma > 0$. Then for every $p \geq 2$ it holds that $\mathbb{E}[|X|^p] \leq c p^{p/2} \sigma^p$.*

Proof. Using Fubini's theorem and the change of variable formula,

$$\mathbb{E}\left[|X|^p\right] = \int_0^\infty p t^{p-1} \mathbb{P}(|X| \geq t) \, \mathrm{d}t = c p 2^{p/2-1} \sigma^p \Gamma(p/2),$$

where Γ is the Gamma function. It remains to apply the bound $\Gamma(p/2) \leq (p/2)^{p/2-1}$ (see Anderson and Qiu (1997)), which holds for $p \geq 2$ by [2, Theorem 1.5]. □

Proof of 42. By combining 40 and 43 we obtain

$$N \int \mathbb{C}_t \mathbb{S}_t(z_{0:t}, \mathrm{d}\boldsymbol{b}_t) \left| \frac{1}{N} \sum_{i=1}^{N} \{b_t^i f_t(x_{t|t}^i) + \tilde{f}_t(x_{t|t}^i)\} - \eta_t\langle z_{0:t}\rangle (f_t B_t \langle z_{0:t-1}\rangle h_t + \tilde{f}_t) \right|^2 \leq \mathsf{c}_t (p/\mathsf{d}_t)^{p/2} N^{-p/2} \left(\|f_t\|_\infty \sum_{m=0}^{t-1} \|\tilde{h}_m\|_\infty + \|\tilde{f}_t\|_\infty \right)^p,$$

which was to be established. □

B.3.4 Proof of 17

Like previously, we establish 17 via a more general result, namely the following.

Proposition 44. *For every $t \in \mathbb{N}$, the exists $\bar{\mathsf{c}}_t^{bias} < \infty$ such that for every $M \in \mathbb{N}^*$, $N \in \mathbb{N}^*$, $z_{0:t} \in \mathsf{X}_{0:t}$, and $(f_t, \tilde{f}_t) \in \mathsf{F}(\mathcal{X}_t)^2$,*

$$\left| \int \mathbb{C}_t \mathbb{S}_t(z_{0:t}, \mathrm{d}\boldsymbol{b}_t) \frac{1}{N} \sum_{i=1}^{N} \{b_t^i f_t(x_{t|t}^i) + \tilde{f}_t(x_{t|t}^i)\} - \eta_t\langle z_{0:t}\rangle (f_t B_t \langle z_{0:t-1}\rangle h_t + \tilde{f}_t) \right| \leq \bar{\mathsf{c}}_t^{bias} \kappa_t N^{-1},$$

where κ_t is defined in (B.1).

We preface the proof of 44 by a technical lemma providing a bound on the bias of ratios of random variables.

Lemma 45. *Let α and β be (possibly dependent) random variables defined on some probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and such that $\mathbb{E}[\alpha^2] < \infty$ and $\mathbb{E}[\beta^2] < \infty$. Moreover, assume that there exist $c > 0$ and $d > 0$ such that $|\alpha/\beta| \leq c$, $\mathbb{P}$-a.s., $|a/b| \leq c$, $\mathbb{E}[(\alpha - a)^2] \leq c^2 d^2$, and $\mathbb{E}[(\beta - b)^2] \leq d^2$. Then*

$$|\mathbb{E}[\alpha/\beta] - a/b| \leq 2c(d/b)^2 + c|\mathbb{E}[\beta - b]|/|b| + |\mathbb{E}[\alpha - a]|/|b|. \tag{B.10}$$

Proof. Using the identity

$$\mathbb{E}[\alpha/\beta] - a/b = \mathbb{E}[(\alpha/\beta)(b-\beta)^2]/b^2 + \mathbb{E}[(\alpha - a)(b-\beta)]/b^2 + a\mathbb{E}[b-\beta]/b^2 + \mathbb{E}[\alpha - a]/b,$$

the claim is established by applying the Cauchy–Schwarz inequality and the assumptions of the lemma according to

$$\begin{aligned}
&|\mathbb{E}[\alpha/\beta] - a/b| \\
&\quad \le c\mathbb{E}[(\beta-b)^2]/b^2 + \{\mathbb{E}[(\alpha-a)^2]\mathbb{E}[(\beta-b)^2]\}^{1/2}/b^2 + |a||\mathbb{E}[\beta-b]|/b^2 + |\mathbb{E}[\alpha-a]|/b^2 \\
&\quad \le 2c(d/b)^2 + c|\mathbb{E}[\beta-b]|/|b| + |\mathbb{E}[\alpha-a]|/|b|.
\end{aligned}$$

□

Proof of 17. We proceed by induction and assume that the claim holds true for $t-1$. Reusing the error decomposition (B.4), it is enough to bound the expectations of the terms $\mathrm{I}_N^{(2)}$ and $\mathrm{I}_N^{(3)}$ given in (B.5) and (B.6), respectively (since $\mathbb{E}_{\boldsymbol{\eta}_0}^{\boldsymbol{P},z}[\mathrm{I}_N^{(1)}] = 0$). This will be done using the induction hypothesis, 45, and 42. More precisely, to bound the expectation of $\mathrm{I}_N^{(2)}$, we use 45 with $\alpha \leftarrow \alpha_t$, $\beta \leftarrow \beta_t$, $a \leftarrow a_t$, and $b \leftarrow b_t$, where

$$\begin{aligned}
\alpha_t &\coloneqq \frac{1}{N}\sum_{\ell=1}^{N}\{\beta_{t-1}^{\ell} Q_{t-1} f_t(\xi_{t-1}^{\ell}) + Q_{t-1}(\tilde{h}_{t-1} f_t + \tilde{f}_t)(\xi_{t-1}^{\ell})\}, \quad \beta_t \coloneqq \frac{1}{N}\sum_{\ell=1}^{N} g_{t-1}(\xi_{t-1}^{\ell}), \\
a_t &\coloneqq \eta_{t-1}\langle z_{0:t-1}\rangle\{Q_{t-1} f_t B_t\langle z_{0:t-1}\rangle h_t + Q_{t-1}(\tilde{h}_{t-1} f_t + \tilde{f}_t)\}, \quad b_t \coloneqq \eta_{t-1}\langle z_{0:t-1}\rangle g_{t-1}.
\end{aligned}$$

For this purpose, note that $|\alpha_t/\beta_t| \le \kappa_t$ and $|a_t/b_t| \le \kappa_t$, where κ_t is defined in (B.1). On the other hand, using 42 (applied with $p=2$), we obtain

$$\mathbb{E}_{\boldsymbol{\eta}_0}^{\boldsymbol{P},z}[(\alpha_t - a_t)^2] \le d_t^2 \kappa_t^2 \quad \text{and} \quad \mathbb{E}_{\boldsymbol{\eta}_0}^{\boldsymbol{P},z}[(\beta_t - b_t)^2] \le d_t^2,$$

where $d_t^2 \coloneqq \mathsf{c}_t \bar{\tau}_{t-1}^2/(\mathsf{d}_t N)$. Using the induction assumption, we get

$$|\mathbb{E}_{\boldsymbol{\eta}_0}^{\boldsymbol{P},z}[\alpha_t] - a_t| \le \bar{\mathsf{c}}_{t-1}^{\mathit{bias}} N^{-1} \bar{\tau}_{t-1} \kappa_t \quad \text{and} \quad |\mathbb{E}_{\boldsymbol{\eta}_0}^{\boldsymbol{P},z}[\beta_t] - b_t| \le \bar{\mathsf{c}}_{t-1}^{\mathit{bias}} N^{-1} \bar{\tau}_{t-1}.$$

Hence, the conditions of 45 are satisfied and we deduce that

$$|\mathbb{E}_{\boldsymbol{\eta}_0}^{\boldsymbol{P},z}[\mathrm{I}_N^{(2)}]| = |\mathbb{E}_{\boldsymbol{\eta}_0}^{\boldsymbol{P},z}[\alpha_t/\beta_t] - a_t/b_t| \le 2\kappa_t \frac{\mathsf{c}_t}{\mathsf{d}_t N}\frac{\bar{\tau}_{t-1}^2}{\underline{\tau}_{t-1}^2} + 2\bar{\mathsf{c}}_{t-1}^{\mathit{bias}} \kappa_t \frac{\bar{\tau}_{t-1}}{\underline{\tau}_{t-1} N}.$$

The bound on $|\mathbb{E}_{\boldsymbol{\eta}_0}^{\boldsymbol{P},z}[\mathrm{I}_N^{(2)}]|$ is obtained along the same lines. □

B.3.5 Proof of 19

We first consider the bias, which can be bounded according to

$$\begin{aligned}
\left|\mathbb{E}_{\xi}[\Pi_{(k_0,k),N}(f)] - \eta_{0:t} h_t\right| &\le (k-k_0)^{-1} \sum_{\ell=k_0+1}^{k} |\mathbb{E}_{\xi}\mu(\boldsymbol{\beta}_t[\ell])(\mathrm{id}) - \eta_{0:t} h_t| \\
&\le (k-k_0)^{-1} N^{-1} \mathsf{c}_t^{\mathit{bias}} \left(\sum_{m=0}^{t-1} \|\tilde{h}_m\|_\infty\right) \sum_{\ell=k_0+1}^{k} \kappa_{N,t}^{\ell},
\end{aligned}$$

from which the bound (3.29) follows immediately.

We turn to the MSE. Using the decomposition

$$\mathbb{E}_\xi[(\Pi_{(k_0,k),N}(f) - \eta_{0:t}h_t)^2] \leq (k-k_0)^{-2}\left\{\sum_{\ell=k_0+1}^{k}\mathbb{E}_\xi[(\mu(\boldsymbol{\beta}_t[\ell])(\mathrm{id}) - \eta_{0:t}h_t)^2] + 2\sum_{\ell=k_0+1}^{k}\sum_{j=\ell+1}^{k}\mathbb{E}_\xi[(\mu(\boldsymbol{\beta}_t[\ell])(\mathrm{id}) - \eta_{0:t}h_t)(\mu(\boldsymbol{\beta}_t[j])(\mathrm{id}) - \eta_{0:t}h_t)]\right\},$$

the MSE bound in 12 implies that

$$\sum_{\ell=k_0+1}^{k}\mathbb{E}_\xi[(\mu(\boldsymbol{\beta}_t[\ell])(\mathrm{id}) - \eta_{0:t}h_t)^2] \leq \mathsf{c}_t^{mse}\left(\sum_{m=0}^{t-1}\|\tilde{h}_m\|_\infty\right)^2 N^{-1}(k-k_0).$$

Moreover, using the covariance bound in 12, we deduce that

$$\sum_{\ell=k_0+1}^{k}\sum_{j=\ell+1}^{k}\mathbb{E}_\xi[(\mu(\boldsymbol{\beta}_t[\ell])(\mathrm{id}) - \eta_{0:t}h_t)(\mu(\boldsymbol{\beta}_t[j])(\mathrm{id}) - \eta_{0:t}h_t)] \leq \mathsf{c}_t^{cov}\left(\sum_{m=0}^{t-1}\|\tilde{h}_m\|_\infty\right)^2 N^{-3/2}\left(\sum_{\ell=k_0+1}^{k}\sum_{j=\ell+1}^{k}\kappa_{N,t}^{(j-\ell)}\right).$$

Thus, the proof is concluded by noting that $\sum_{\ell=k_0+1}^{k}\sum_{j=\ell+1}^{k}\kappa_{N,t}^{(j-\ell)} \leq (k-k_0)/(1-\kappa_{N,t})$.

Appendix C

Appendix of Chapter 4

C.1 Conditions on the model to verify A3

In our specific application to score ascent, we work with the following assumptions.

A12 (Lipschitz). *(i) For all $t \in \mathbb{N}$, there exists $L_t^s \in \mathsf{M}(\mathsf{X}_{t:t+1})$ such that for all $(x_t, x_{t+1}) \in \mathsf{X}_{t:t+1}$, the function $\theta \mapsto s_{t,\theta}(x_t, x_{t+1})$ is $L_t^s(x_t, x_{t+1})$-Lipschitz and $\mathsf{X}_{t:t+1} \ni (x_t, x_{t+1}) \mapsto s_{t,\theta}(x_t, x_{t+1})$ is bounded by $\|s_t(\theta)\|_\infty$ for all $\theta \in \Theta$. Furthermore, $\|L_k^s\|_\infty < \infty$.*
(ii) For all $t \in \mathbb{N}$, there exists $L_t^q \in \mathsf{X}_{\mathrm{t:t+1}}$ such that $\|L_t^q\|_\infty < \infty$ and that for all $(x_t, x_{t+1}) \in \mathsf{X}_{t:t+1}$, $\theta \mapsto q_{t,\theta}(x_t, x_{t+1})$ is $L_t^q(x_t, x_{t+1})$-Lipschitz.

Lemma 46 (A5(i) holds). *Assume A 11 and A 3. There exists a constant L^V such that the Lyapunov function V satisfies, for all $(\theta_1, \theta_2) \in \Theta^2$,*

$$\|\nabla V(\theta_1) - \nabla V(\theta_2)\| \leq L^V \|\theta_1 - \theta_2\|.$$

Proof. For all θ_1, θ_2,

$$\begin{aligned}\|\nabla V(\theta_1) - \nabla V(\theta_2)\| &= \|\eta_{0:t,\theta_1}(s_{0:t,\theta_1}) - \eta_{0:t,\theta_2}(s_{0:t,\theta_2})\| \\ &\leq \|\eta_{0:t,\theta_1}(s_{0:t,\theta_1}) - \eta_{0:t,\theta_1}(s_{0:t,\theta_2})\| + \|\eta_{0:t,\theta_1}(s_{0:t,\theta_2}) - \eta_{0:t,\theta_2}(s_{0:t,\theta_2})\|\,.\end{aligned}$$

By (2) and by (Gloaguen et al., 2022, Theorem 4.10) there exists a constant c such that

$$\|\eta_{0:t,\theta_1}(s_{0:t,\theta_2}) - \eta_{0:t,\theta_2}(s_{0:t,\theta_2})\| \leq ct\|\theta_1 - \theta_2\| \sup_\theta \sup_k \|s_k(\theta)\|_\infty\,,$$

Using **A** 2 and **A** 3[i], we can write:

$$\begin{aligned}\|\eta_{0:t,\theta_1}(s_{0:t,\theta_1}) - \eta_{0:t,\theta_1}(s_{0:t,\theta_2})\| &\leq \sum_{u=0}^{t-1} \eta_{0:t,\theta_1}\left[\|s_{u,\theta_1}(x_{u:u+1}) - s_{u,\theta_2}(x_{u:u+1})\|\right], \\ &\leq \sum_{u=0}^{t-1} \eta_{0:t,\theta_1}\left[L_u^s(x_{u:u+1})\right] \|\theta_1 - \theta_2\|, \\ &\leq \frac{\sigma_+}{\sigma_-} \sup_{u\in[\![0,t-1]\!]}\left[L_u^s\right] \|\theta_1 - \theta_2\| t.\end{aligned}$$

□

Theorem 47 (Lipschitz continuity of Particle Gibbs with Backward Sampling). *Assume A 12. For every $t \in \mathbb{N}$, $\theta \in \Theta$ and $N \in \mathbb{N}^*$*

$$\sup_{x_{0:t}\in\mathsf{X}_{0:t}} \|K_{\theta_1,t}(x_{0:t}, .) - K_{\theta_2,t}(x_{0:t}, .)\|_{\mathrm{TV}} \leq L_{t,N}^K \|\theta_1 - \theta_2\|\,,$$

where

$$L_{t,N}^K \coloneqq \sum_{\ell=0}^{t-1} \bar{\tau}_\ell^{-1} \left[\bar{\sigma}_\ell^{-1} + (N-1)\right] \|L_\ell^q\|_\infty \,. \tag{C.1}$$

Proof. We know that $K_{\theta,t} = \mathbb{C}_{m,\theta}\mathbb{B}_{t,\theta}$. Therefore, by Lemmas 57, 59 and 63, we have that $K_{\theta,t}$ is Lipschitz with constant equals $L_t^{\mathbb{C}} + \sup_\theta \mathbb{C}_{t,\theta} L_t^{\mathbb{B}}$. □

Corollary 48 (A3(iii) holds.)**.** *Assume A 12. For every $t \in \mathbb{N}$, $\theta \in \Theta$, $r \in \mathbb{N}^*$ and $N \in \mathbb{N}^*$ such that $N > 1 + 5\rho_t^2 t/2$*

$$\sup_{x_{0:t} \in \mathsf{X}_{0:t}} \left\| K_{\theta_1,t}^r(x_{0:t}, .) - K_{\theta_2,t}^r(x_{0:t}, .) \right\|_{\mathrm{TV}} \leq L_{t,N}^P \|\theta_1 - \theta_2\|$$

where

$$L_{t,N}^P \coloneqq (1 - \kappa_{t,N})^{-1} \|L_{t,N}^K\|_\infty \tag{C.2}$$

where $L_{t,N}^K$ is defined in (C.1).

Proof. Under 11, the Particle Gibbs with backward sampling is geometrically ergodic with contraction rate $\kappa_{t,N}$ and thus $L_{t,N}^K$ is bounded and the result follows from Lemma 62 □

Corollary 49 (A3(i))**.** *Assume A 11 and A 12. For all $t \in \mathbb{N}^*$, $(\theta_0, \theta_1) \in \Theta^2$,*

$$\|\eta_{0:t,\theta_0} - \eta_{0:t,\theta_1}\|_{\mathrm{TV}} \leq L^\eta \|\theta_0 - \theta_1\|,$$

where

$$L^\eta \coloneqq L_{t,N^*}^P \,, \tag{C.3}$$

and $L_{t,N}^P$ is defined in (C.2) and $N^* = \lceil 1 + 5\rho_t^2/2 \rceil$.

Proof. Consider the following decomposition, valid for all $k \in \mathbb{N}^*$ and $N \geq 1 + 5\rho_t^2/2$, and all $x_{0:t} \in \mathsf{X}_{0:t}$,

$$\begin{aligned}
&\|\eta_{0:t,\theta_1} - \eta_{0:t,\theta_2}\|_{\mathrm{TV}} \\
&\quad \leq \left\|\eta_{0:t,\theta_1} - K_{\theta_1,t}^k(x_{0:t}, \cdot)\right\|_{\mathrm{TV}} + \left\|\eta_{0:t,\theta_2} - K_{\theta_2,t}^k(x_{0:t}, \cdot)\right\|_{\mathrm{TV}} + \left\|K_{\theta_1,t}^k(x_{0:t}, \cdot) - K_{\theta_2,t}^k(x_{0:t}, \cdot)\right\|_{\mathrm{TV}} \\
&\quad \leq \left\|\eta_{0:t,\theta_1} - K_{\theta_1,t}^k(x_{0:t}, \cdot)\right\|_{\mathrm{TV}} + \left\|\eta_{0:t,\theta_2} - K_{\theta_2,t}^k(x_{0:t}, \cdot)\right\|_{\mathrm{TV}} + L_{t,N}^P \|\theta_1 - \theta_2\| \,,
\end{aligned}$$

where we applied Corollary 48. Since the Lipschitz constant of $K_{\theta,t}$ is independent of k, and $K_{\theta,t}$ is geometrically ergodic for all θ, we obtain by taking the limit when k goes to infinity with N fixed,

$$\|\eta_{0:t,\theta_1} - \eta_{0:t,\theta_2}\|_{\mathrm{TV}} \leq \frac{\|L_{t,N}^K\|_\infty}{1 - \kappa_{t,N}} \|\theta_1 - \theta_2\| \,,$$

for all $N \geq 1 + 5\rho_t^2/2$, where the dependence in N is hidden in $L_{t,N}^P$. The result follows by choosing $N = \lceil 1 + 5\rho_t^2/2 \rceil$. □

Remark 50. *As noted by Lindholm and Lindsten (2018), the Lipschitz constant appearing in Corollary 48 possesses an unexpected dependence on $N - 1$. One would expect it not to be true, in that we know that $\mathbb{K}_{\theta,t}$ converges geometrically fast and uniformly to $\eta_{0:t}$ and this is faster as N gets bigger. Therefore, for large N the Lipschitz constant is expected to converge to that of $\eta_{0:t}$ whose Lipschitz constant is independent of N.*

Proposition 51 (Lipschitz continuity of $\theta \mapsto \mathbb{K}_{\theta,t}\mu(\boldsymbol{\beta}_t)(\mathrm{id})$)**.** *Assume A 12. For every $t \in \mathbb{N}$, $\theta \in \Theta$ and $N \in \mathbb{N}^*$,*

$$\|\mathbb{K}_{\theta_1,t}\mu(\boldsymbol{\beta}_t)(\mathrm{id}) - \mathbb{K}_{\theta_2,t}\mu(\boldsymbol{\beta}_t)(\mathrm{id})\|_\infty \leq L_t^{\mathbb{K}}\|\theta_1 - \theta_2\| \ ,$$

where

$$L_t^{\mathbb{K}} \coloneqq (N-1)\sum_{\ell=0}^{t-1}\bar{\tau}_\ell\|L_\ell^q\|_\infty + \sum_{j=1}^{m}\|L_j^{\overleftarrow{Q}}\|_\infty\left[\sum_{\ell=0}^{m-1}s_\ell^\infty\right] + \sum_{j=1}^{m}\|L_j^s\|_\infty \ . \tag{C.4}$$

Proof. Consider $e = (x_{0:t}, \boldsymbol{y}_{0:t}) \in \mathsf{E}_t$ and $f_\theta(e) \coloneqq \int \mathbb{S}_{m,\theta}(x_{0:t}, \mathrm{d}\tilde{\boldsymbol{y}}_t)\mu(\boldsymbol{b}_t)(id)$. Then $\mathbb{K}_{\theta,t}\mu(\boldsymbol{b}_t)(id) = \mathbb{C}_{m,\theta}f_\theta(x_{0:t})$ is a composition of a Markov kernel and a Lipschitz function, therefore Lipschitz. □

Corollary 52 (A3(iv) holds.)**.** *Assume A 12. For every $t \in \mathbb{N}$, $\theta \in \Theta$ and $N \in \mathbb{N}^*$*

$$\sup_{x_{0:t}\in\mathsf{X}_{0:t}}\|\mathbb{P}_{\theta_1,t}H - \mathbb{P}_{\theta_2,t}H\| \leq L_2^P\|\theta_1 - \theta_2\| \ ,$$

where

$$L_2^P = L_{t,N}^P + L_t^{\mathbb{K}} \ , \tag{C.5}$$

with L^P and $L_t^{\mathbb{K}}$ are defined in (C.4) and (C.2).

Proof. Let $\tilde{f} : \mathsf{E}^{k-k_0} \ni (x_{0:t}[k_0:k], \boldsymbol{x}_{0:t|t}[k_0:k], \boldsymbol{b}_t[k_0:k]) \mapsto (k-k_0)^{-1}\sum_{\ell=k_0+1}^{k}\mu(\boldsymbol{b}_t[\ell])(\mathrm{id})$. As $\mathbb{K}_{\theta,t}$ depends only on the path, with a slight abuse of notation, we can define $f_\theta(x_{0:t}) \coloneqq \mathbb{K}_{\theta,t}^{\otimes k-k_0}(\tilde{f})(x_{0:t})$. By proposition 51, we have that f_θ is Lipschitz with $L^f = L_t^{\mathbb{K}}$. Note that $\mathbb{P}_{\theta,t}H(x_{0:t}, \boldsymbol{y}_t) = K_{\theta,t}^{k_0}f_\theta(x_{0:t})$, therefore, by lemma 63 Lipschitz with constant $L^P + L_t^{\mathbb{K}}$. □

C.2 Lipschitz properties

C.2.1 Lipschitz continuity of $\mathbb{P}_{\theta,}$

In this section we prove the following items:

- $\mathbb{C}_{m,\theta}(z_{0:m}, \cdot)$ is Lipschitz, see Section C.2.1
- $\mathbb{B}_{m,\theta}(\boldsymbol{x}_{0:m}, \cdot)$ is Lipschitz, see Line 45
- $\int \mathbb{S}_{m,\theta}(\boldsymbol{x}_{0:m}, \mathrm{d}\boldsymbol{b}_m)\mu(\boldsymbol{b}_m)(\mathrm{Id})$ is Lipschitz, see Line 45

The following technical lemma will be useful.

Lemma 53. *Let $\alpha \in]0,1]$, $x \in \mathbb{R}_{\geq 0}$ and $\ell \in \mathbb{N}$. Then for all $\lambda_i \in \mathbb{R}_{\geq 0}$, $i \in [\![0,\ell]\!]$, such that $\alpha \geq \prod_{i=0}^{\ell}(1-\lambda_i x)$ it holds that $\alpha \geq 1 - x\sum_{i=0}^{\ell}\lambda_i$.*

Proof. Consider first the case where $x\lambda_i \leq 1$ for all $i \in [\![0,\ell]\!]$. We prove the result by induction. The case $\ell = 0$ is straightforward. Assume now that the result holds for some $r \in [\![0, \ell-1]\!]$. Then,

$$\begin{aligned}\prod_{i=0}^{r+1}(1-\lambda_i x) = (1-\lambda_{r+1}x)\prod_{i=0}^{r}(1-\lambda_i x) &\geq (1-\lambda_{r+1}x)(1-x\sum_{i=0}^{r}\lambda_i) \\ &= 1 - x\sum_{i=0}^{r+1}\lambda_i + x^2\sum_{i=0}^{r}\lambda_i\lambda_{r+1} \geq 1 - x\sum_{i=0}^{r+1}\lambda_i \ .\end{aligned}$$

Consider now the case where there is a index $j \in [\![0,\ell]\!]$ such that $x\lambda_j \geq 1$. Then $\alpha \geq 0 \geq 1 - (\sum_{i=0}^{\ell}\lambda_i)x$. □

We begin with some important definitions. Let P and Q be probability distributions on some common measurable space $(\mathsf{X}, \mathcal{X})$, and assume that these distributions admit densities p and q w.r.t some common reference measure λ. Let $\mathbb{M}[P, Q]$ denote a maximal coupling between P and Q. As in (Lindholm and Lindsten, 2018, Theorem 2), it is possible to explicitly construct one such maximal coupling by

$$\mathbb{M}[P,Q](\mathrm{d}(x,y)) \coloneqq \min\{p(x), g(x)\}\lambda(\mathrm{d}x)\delta_x(\mathrm{d}y) + \frac{[P(\mathrm{d}x) - \min\{p(x), g(x)\}\lambda(\mathrm{d}x)]\,[Q(\mathrm{d}y) - \min\{p(y), g(y)\}\lambda(\mathrm{d}y)]}{1 - \lambda(\min\{p, q\})}. \quad \text{(C.6)}$$

From this definition it follows that for continuous and discrete dominating measures λ,

$$\int \mathbb{1}_{\{x=y\}}\mathbb{M}[P,Q]\,\mathrm{d}(x,y) = \int \min\{p(x), g(x)\}\lambda(\mathrm{d}x)\,.$$

Moreover, for two Markov transition kernels K_1 and K_2 on $(\mathsf{X}, \mathcal{X})$, which are assumed to admit transition densities with respect to some common dominating measure, we let, for $(x_1, x_2) \in \mathsf{X}^2$, $\mathbb{M}[K_1, K_2]((x_1, x_2), \cdot)$ denote the maximal coupling between the measures $K_1(x_1, \cdot)$ and $K_2(x_2, \cdot)$. Defined in this way, $\mathbb{M}[K_1, K_2]$ defines a Markov transition kernel on the product space $(\mathsf{X}^2, \mathcal{X}^{\otimes 2})$

The following Lemma will be crucial in what follows.

Lemma 54. *(i) Let (μ_1, μ_2) be two probability measures admitting a density with respect to a common dominating measure and let (K_1, K_2) two Markov transition kernels also admitting transition densities with respect to some dominating measure. Then the probability measure*

$$\mathbb{M}[\mu_1,\mu_2]\,\mathbb{M}[K_1,K_2](\mathrm{d}(x_1,x_2)) = \int \mathbb{M}[\mu_1,\mu_2](\mathrm{d}(z_1,z_2))\mathbb{M}[K_1,K_2]((z_1,z_2),\mathrm{d}(x_1,x_2)),$$

is a coupling of $(\mu_1 K_1, \mu_2 K_2)$, and it holds that

$$\int \mathbb{1}_{x_1=x_2}\mathbb{M}[\mu_1 K_1, \mu_2 K_2](\mathrm{d}(x_1,x_2)) \geq \int\int \mathbb{1}_{z_1=z_2}\mathbb{1}_{x_1=x_2}\mathbb{M}[\mu_1,\mu_2](\mathrm{d}(z_1,z_2))\mathbb{M}[K_1,K_2]((z_1,z_2),\mathrm{d}(x_1,x_2)).$$

(ii) Let $(\mu_1, \cdots, \mu_n)$ and $(\nu_1, \cdots, \nu_n)$ be probability measures such that for all $i \in [\![1, n]\!]$, μ_i and ν_i admit densities with respect to the same dominating measure. Then $\bigotimes_{i=1}^n \mathbb{M}[\mu_i, \nu_i]$ is a coupling of $\bigotimes_{i=1}^n \mu_i$ and $\bigotimes_{i=1}^n \nu_i$, and thus

$$\int \prod_{i=1}^n \mathbb{1}_{x_i=y_i}\mathbb{M}\left[\bigotimes_{i=1}^n \mu_i, \bigotimes_{i=1}^n \nu_i\right](\mathrm{d}(x_1,\ldots,x_n,y_1,\ldots,y_n)) \geq \int \prod_{i=1}^n \mathbb{1}_{x_i=y_i}\bigotimes_{i=1}^n \mathbb{M}[\mu_i,\nu_i](\mathrm{d}(x_1,\ldots,x_n,y_1,\ldots,y_n)).$$

Proof. It is enough to show that $\mathbb{M}[\mu_1, \mu_2]\,\mathbb{M}[K_1, K_2]$ admits $\mu_1 K_1$ and $\mu_2 K_2$ as marginal distributions. This follows immediately from the fact that $\mathbb{M}[\mu_1, \mu_1]$ and $\mathbb{M}[K_1, K_2]$ admit the right marginal distributions; indeed,

$$\begin{aligned}
&\mathbb{M}[\mu_1,\mu_2]\mathbb{M}[K_1,K_2](\mathsf{X}\times A)\\
&\quad= \int \mathbb{M}[\mu_1,\mu_2](\mathrm{d}z_1,\mathrm{d}_2)\,\mathbb{M}[K_1,K_2](z_1,z_2,\mathrm{d}(x_1,x_2))\,\mathbb{1}_{\mathsf{X}\times A}(x_1,x_2)\mathbb{1}_{\mathsf{X}^2}(z_1,z_2)\\
&\quad= \int \mathbb{M}[\mu_1,\mu_2](\mathrm{d}z_1,\mathrm{d}_2)K_2(z_2,A)\\
&\quad= \int \mu_2(\mathrm{d}z_2)K_2(z_2,A)\\
&\quad= \mu_2 K_2(A).
\end{aligned}$$

The derivation for the first marginal distribution follows similarly. For the second point, $\mathbb{M}\left[\mu_1, \mu_2\right] \mathbb{M}\left[K_1, K_2\right]$ is a coupling of $(\mu_1 K_1, \mu_2 K_2)$ and $\mathbb{M}\left[\mu_1 K_1, \mu_2 K_2\right]$ is the maximal coupling, we have that

$$\begin{aligned}
&\int \mathbb{1}_{x_1=x_2} \mathbb{M}\left[\mu_1 K_1, \mu_2 K_2\right]\left(\mathrm{d}(x_1, x_2)\right) \\
&\qquad \geq \iint \mathbb{1}_{x_1=x_2} \mathbb{M}\left[\mu_1, \mu_2\right]\left(\mathrm{d}(z_1, z_2)\right) \mathbb{M}\left[K_1, K_2\right]\left(z_1, z_2 ; \mathrm{d}(x_1, x_2)\right) \\
&\qquad \geq \iint \mathbb{1}_{x_1=x_2} \mathbb{1}_{z_1=z_2} \mathbb{M}\left[\mu_1, \mu_2\right]\left(\mathrm{d}(z_1, z_2)\right) \mathbb{M}\left[K_1, K_2\right]\left(z_1, z_2 ; \mathrm{d}(x_1, x_2)\right).
\end{aligned}$$

The proof of the second item follows similarly. □

$\theta \mapsto \mathbb{C}_{m,\theta}$ is Lipschitz. We proceed by a coupling method that is inspired by (Lindholm and Lindsten, 2018, Theorem 2). The coupling we consider is that where the *selection* and *mutation* steps of the particle filter are respectively coupled maximally.

Algorithm 11 Coupling $\mathbb{C}_{m,\theta}$

Data: θ_1, θ_2, $\zeta_{0:m}$
Result: $\boldsymbol{x}_{0:m,1}$, $\boldsymbol{x}_{0:m,1}$
draw $\boldsymbol{x}_{0,1}, \boldsymbol{x}_{0,2} \sim \mathbb{M}\left[\boldsymbol{\eta}_0\langle\zeta_0\rangle, \boldsymbol{\eta}_0\langle\zeta_0\rangle\right]$
for $s \leftarrow 1$ **to** t **do**
  draw $(\boldsymbol{x}_{s,1}, \boldsymbol{x}_{s,2}) \sim \mathbb{M}\left[\boldsymbol{M}_{s-1,\theta_1}\langle\zeta_s\rangle(\boldsymbol{x}_{s-1,1}, \cdot), \boldsymbol{M}_{s-1,\theta_2}\langle\zeta_s\rangle(\boldsymbol{x}_{s-1,2}, \cdot)\right]$

First, let us prove that the one step *selection–mutation* kernel is Lipschitz.

Lemma 55. *For all $t \in \mathbb{N}$, $\boldsymbol{x}_{t-1} \in \mathbf{X}_{t-1}$ and $(\theta_1, \theta_2) \in \Theta^2$,*

$$\int \mathbb{1}_{\{x_1=x_2\}} \mathbb{M}\left[\Phi_{t-1,\theta_1}(\mu(\boldsymbol{x}_{t-1})), \Phi_{t-1,\theta_2}(\mu(\boldsymbol{x}_{t-1}))\right]\left(\mathrm{d}(x_1, x_2)\right) \geq 1 - \frac{\sum_{i=1}^N \lambda_t(L_{t-1}^q(x_{t-1}^i, \cdot))}{N\bar{\tau}_n}\|\theta_1 - \theta_2\|. \tag{C.7}$$

Proof. By **A**2(i) and **A**3(iii),

$$\begin{aligned}
&\int \mathbb{1}_{\{x_1=x_2\}} \mathbb{M}\left[\Phi_{t-1,\theta_1}(\mu(\boldsymbol{x}_{t-1})), \Phi_{t-1,\theta_2}(\mu(\boldsymbol{x}_{t-1}))\right]\left(\mathrm{d}(x_1, x_2)\right) \\
&\qquad = \int \min\left(\sum_{i=1}^N \frac{q_{t-1,\theta_1}(x_{t-1}^i, x)}{\sum_{j=1}^N g_{t-1,\theta_1}(x_{t-1}^j)}, \sum_{i=1}^N \frac{q_{t-1,\theta_2}(x_{t-1}^i, x)}{\sum_{j=1}^N g_{t-1,\theta_2}(x_{t-1}^j)}\right) \lambda_t(\mathrm{d}x) \\
&\qquad \geq \sum_{j=1}^N \int \min\left(\frac{q_{t-1,\theta_1}(x_{t-1}^i, x)}{\sum_{j=1}^N g_{t-1,\theta_1}(x_{t-1}^j)}, \frac{q_{t-1,\theta_2}(x_{t-1}^i, x)}{\sum_{j=1}^N g_{t-1,\theta_2}(x_{t-1}^j)}\right) \lambda_t(\mathrm{d}x) \\
&\qquad \geq \frac{1}{\sum_{j=1}^N \max\left(g_{t-1,\theta_1}(x_{t-1}^j), g_{t-1,\theta_2}(x_{t-1}^j)\right)} \sum_{j=1}^N \int \min\left(q_{t-1,\theta_1}(x_{t-1}^j, x), q_{t-1,\theta_2}(x_{t-1}^j, x)\right) \lambda_t(\mathrm{d}x) \\
&\qquad \geq \frac{\sum_{j=1}^N \max\left(g_{t-1,\theta_1}(x_{t-1}^j), g_{t-1,\theta_2}(x_{t-1}^j)\right) - \sum_{i=1}^N \lambda_t\left(L_{t-1}^q(x_{t-1}^i, \cdot)\right)\|\theta_1 - \theta_2\|}{\sum_{j=1}^N \max\left(g_{t-1,\theta_1}(x_{t-1}^j), g_{t-1,\theta_2}(x_{t-1}^j)\right)} \\
&\qquad \geq 1 - \frac{\sum_{i=1}^N \lambda_t(L_{t-1}^q(x_{t-1}^i, \cdot))}{N\bar{\tau}_n}\|\theta_1 - \theta_2\|,
\end{aligned}$$

where we have used that

$$\int \max(q_{t-1,\theta_1}(x^i_{t-1},x), q_{t-1,\theta_2}(x^i_{t-1},x))\lambda_t(\mathrm{d}x) \geq \max\left(\int q_{t-1,\theta_1}(x^i_{t-1},x)\lambda_t(\mathrm{d}x), \int q_{t-1,\theta_2}(x^i_{t-1},x)\lambda_t(\mathrm{d}x)\right)$$
$$\geq \max(g_{t-1,\theta_1}(x^i_{t-1}), g_{t-1,\theta_2}(x^i_{t-1})).$$

□

Lemma 56. *For all $t \in \mathbb{N}$, $\boldsymbol{x}_{t-1} \in \mathbf{X}_{t-1}$, $z \in \mathsf{X}_t$ and $(\theta_1,\theta_2) \in \Theta^2$,*

$$\|\boldsymbol{M}_{t-1,\theta_1}\langle z\rangle(\boldsymbol{x}_{t-1},\cdot) - \boldsymbol{M}_{t-1,\theta_2}\langle z\rangle(\boldsymbol{x}_{t-1},\cdot)\|_{\mathrm{TV}} \leq L^{\boldsymbol{M}}_{t-1}(\boldsymbol{x}_{t-1})\|\theta_1-\theta_2\|$$

where $L^{\boldsymbol{M}}_{t-1}(\boldsymbol{x}_{t-1}) = (1-N^{-1})\bar{\tau}^{-1}_{t-1}\sum_{i=1}^N \lambda_t\left(L^q_{t-1}(x^i_{t-1},\cdot)\right)$.

Proof. Let us denote by $\mathrm{U}[\![1,n]\!]$ the uniform distribution on $[\![1,n]\!]$. By definition of the kernel $\boldsymbol{M}_{t-1,\theta}\langle z\rangle$, we have that

$$\boldsymbol{M}_{t-1,\theta}\langle z\rangle(\boldsymbol{x}_{t-1},\mathrm{d}\boldsymbol{x}_t) = \int \mathrm{U}[\![1,n]\!](\mathrm{d}j)\{\Phi_{t-1}(\mu(\boldsymbol{x}_{t-1}))^{\otimes j} \otimes \delta_z \otimes \Phi_{t-1}(\mu(\boldsymbol{x}_{t-1}))^{\otimes(N-j-1)}\}(\mathrm{d}\boldsymbol{x}_t)$$

and thus, applying the two items of Lemma 54 combined with the fact that $\mathbb{M}\left[\mu,\mu\right]\left(\mathrm{d}(x_1,x_2)\right) = \mu(\mathrm{d}x_1)\delta_{x_1}(\mathrm{d}x_2)$ for any probability measure μ, we get that

$$\begin{aligned}
&\int \mathbb{1}_{\{\boldsymbol{x}_{t,1}=\boldsymbol{x}_{t,2}\}}\mathbb{M}\left[\boldsymbol{M}_{t-1,\theta_1}\langle z\rangle(\boldsymbol{x}_{t-1},\cdot), \boldsymbol{M}_{t-1,\theta_2}\langle z\rangle(\boldsymbol{x}_{t-1},\cdot)\right]\mathrm{d}(\boldsymbol{x}_{t,1},\boldsymbol{x}_{t,2}) \\
&\quad\geq \int \mathbb{1}_{\boldsymbol{x}_{t,1}=\boldsymbol{x}_{t,2},i_1=i_2}\mathbb{M}\left[\mathrm{U}[\![1,n]\!],\mathrm{U}[\![1,n]\!]\right]\left(\mathrm{d}(i_1,i_2)\right) \\
&\qquad\qquad \times \mathbb{M}\left[\Phi_{t-1,\theta_1}(\mu(\boldsymbol{x}_{t-1})),\Phi_{t-1,\theta_2}(\mu(\boldsymbol{x}_{t-1}))\right]^{\otimes i_1} \otimes \mathbb{M}\left[\delta_z,\delta_z\right] \\
&\qquad\qquad \otimes \mathbb{M}\left[\Phi_{t-1,\theta_1}(\mu(\boldsymbol{x}_{t-1})),\Phi_{t-1,\theta_2}(\mu(\boldsymbol{x}_{t-1}))\right]^{\otimes N-i_1-1}\mathrm{d}(\boldsymbol{x}_{t,1},\boldsymbol{x}_{t,2}) \\
&\quad= \frac{1}{N}\sum_{i=1}^N \int \prod_{k=1,k\neq i}^n \mathbb{1}_{x^i_{t,1}=x^i_{t,2}}\mathbb{M}\left[\Phi_{t-1,\theta_1}(\mu(\boldsymbol{x}_{t-1})),\Phi_{t-1,\theta_2}(\mu(\boldsymbol{x}_{t-1}))\right]\left(\mathrm{d}(x^i_{t,1},x^i_{t,2})\right) \\
&\quad\geq \left(1 - \frac{\sum_{i=1}^N \lambda_t(L^q_{t-1}(x^i_{t-1},\cdot))}{N\bar{\tau}_{t-1}}\|\theta_1-\theta_2\|\right)^{N-1} \\
&\quad\geq 1 - \frac{N-1}{\bar{\tau}_{t-1}N}\sum_{i=1}^N \lambda_t(L^q_{t-1}(x^i_{t-1},\cdot))\|\theta_1-\theta_2\| \,.
\end{aligned}$$

where we have applied Lemma 55 in the penultimate line and Lemma 53 in the last one. □

Lemma 57. *For every $t \in \mathbb{N}^*$, there exists $L^{\mathbb{C}}_t \in \mathsf{M}(\mathcal{X}_{0:t})$ such that*

$$\|\mathbb{C}_{t,\theta_1}(z_{0:t}) - \mathbb{C}_{t,\theta_2}(z_{0:t})\|_{\mathrm{TV}} \leq L^{\mathbb{C}}_t(z_{0:t})\|\theta_1-\theta_2\| \,, \tag{C.8}$$

*where $L^{\mathbb{C}}_t(z_{0:t}) = \sup_\theta \mathbb{C}_{t,\theta}\left[\sum_{i=0}^{t-1} L^{\boldsymbol{M}}_i\right](z_{0:t})$. Under **A** 12(i), we obtain that $\|L^{\mathbb{C}}_t\|_\infty \leq (N-1)\sum_{\ell=0}^{t-1}\bar{\tau}_\ell\|L^q_\ell\|_\infty$.*

Proof. This is a direct application of lemma 65. □

$\theta \mapsto \mathbb{B}_{t,\theta}(\boldsymbol{x}_{0:t}, \cdot)$ **is Lipschitz** We start by recalling the definition of $\mathbb{B}_m$

$$\mathbb{B}_{t,\theta} : \mathbf{X}_{0:t} \times \mathcal{X}_{0:t} \ni (\boldsymbol{x}_{0:t}, A) \mapsto \int \cdots \int \mathbb{1}_A(x_{0:t}) \left(\prod_{s=0}^{t-1} \overleftarrow{Q}_{s,\mu(\boldsymbol{x}_s)}(x_{s+1}, \mathrm{d}x_s) \right) \mu(\boldsymbol{x}_t)(\mathrm{d}x_t) \ . \tag{C.9}$$

Lemma 58. *For all $s \in [\![0, t]\!]$, $x_{t+1} \in \mathsf{X}_{t+1}$, $\boldsymbol{x}_t \in \mathbf{X}_t$ and $(\theta_1, \theta_2) \in \Theta^2$*

$$\left\| \overleftarrow{Q}_{s,\mu(\boldsymbol{x}_s),\theta_1}(x_{s+1}, \cdot) - \overleftarrow{Q}_{s,\mu(\boldsymbol{x}_s),\theta_2}(x_{s+1}, \cdot) \right\|_{\mathrm{TV}} \leq L_s^{\overleftarrow{Q}}(x_{s+1}, \boldsymbol{x}_s) \|\theta_1 - \theta_2\| \ . \tag{C.10}$$

with $L_s^{\overleftarrow{Q}}(x_{s+1}, \boldsymbol{x}_s) = (N\bar{\tau}_t\bar{\sigma}_s)^{-1} \sum_{i=1}^N L_s^q(x_s^i, x_{s+1})$. Under **A** *12(i), we have $\|L_m^{\overleftarrow{Q}}\|_\infty = (\bar{\tau}_m\bar{\sigma}_m)^{-1}\|L_m^q\|_\infty$.*

Proof. Note that $\overleftarrow{Q}_{t,\mu(\boldsymbol{x}_t)}(x_{t+1}, \cdot) = \sum_{\ell=1}^N \frac{q_t(x_t^\ell, x_{t+1})}{\sum_{\ell'=1}^N q_t(x_t^{\ell'}, x_{t+1})} \delta_{x_t^\ell}$. Therefore, similarly to the proof of Lemma 55,

$$\begin{aligned}
&\int \mathbb{1}_{\{x_{t,1}=x_{t,2}\}} \mathbb{M}\left[\overleftarrow{Q}_{t,\mu(\boldsymbol{x}_t),\theta_1}(x_{t+1}, \cdot), \overleftarrow{Q}_{t,\mu(\boldsymbol{x}_t),\theta_2}(x_{t+1}, \cdot) \right] \mathrm{d}(x_{t,1}, x_{t,2}) \\
&\qquad \geq \frac{\sum_{\ell=1}^N \max(q_{t,\theta_1}(x_t^\ell, x_{t+1}), q_{t,\theta_2}(x_t^\ell, x_{t+1})) - L_t^q(x_t^\ell, x_{t+1})\|\theta_1 - \theta_2\|}{\sum_{\ell=1}^N \max(q_{t,\theta_1}(x_t^\ell, x_{t+1}), q_{t,\theta_2}(x_t^\ell, x_{t+1}))} \\
&\qquad \geq 1 - \frac{\sum_{\ell=1}^N L_t^q(x_t^\ell, x_{t+1})}{N\bar{\tau}_t\bar{\sigma}_t} \|\theta_1 - \theta_2\| \ .
\end{aligned}$$

□

Lemma 59. *For all $t \in \mathbb{N}$, $\boldsymbol{x}_{0:t} \in \mathbf{X}_{0:t}$ and $(\theta_1, \theta_2) \in \Theta^2$*

$$\|\mathbb{B}_{t,\theta_1}(\boldsymbol{x}_{0:t}, \cdot) - \mathbb{B}_{t,\theta_2}(\boldsymbol{x}_{0:t}, \cdot)\|_{\mathrm{TV}} \leq L_t^{\mathbb{B}}(\boldsymbol{x}_{0:t})\|\theta_1 - \theta_2\| \tag{C.11}$$

where $L_t^{\mathbb{B}}(\boldsymbol{x}_{0:t}) = \sup_\theta \mathbb{B}_t \left[\sum_{i=0}^{t-1} L_i^{\overleftarrow{Q}} \right] (\boldsymbol{x}_{0:t})$. Under **A** *12(i), we have that $\|L_t^{\mathbb{B}}\|_\infty = \sum_{i=0}^{t-1} (\bar{\tau}_i\bar{\sigma}_i)^{-1}\|L_i^q\|_\infty$.*

Proof. Apply lemma 63 and lemma 58. □

$\theta \mapsto \int \mathbb{S}_{t,\theta}(\boldsymbol{x}_{0:t}, \mathrm{d}\boldsymbol{b}_t)\mu(\boldsymbol{b}_t)(\mathrm{id})$ **is Lipschitz** Define the backward ancestors kernel

$$\mathcal{B}_{\theta,t} : \mathsf{X}_{t+1} \times \mathbf{X}_t \times \sigma([\![1, N]\!]) \mapsto \int \mathbb{1}_A(\tilde{j}) \left(\sum_{\ell=1}^N \frac{q_t(x_t^\ell, x_{t+1})}{\sum_{\ell'=1}^N q_t(x_t^{\ell'}, x_{t+1})} \delta_\ell(\mathrm{d}\tilde{j}) \right) \ .$$

Lemma 60. *($\mathcal{B}_{\theta,t}$ is Lipschitz) For every $m \in [\![0, t]\!]$, there exists $L_m^{BK} \in \mathsf{M}(\boldsymbol{\mathcal{X}}_{m:m+1})$ such that*

$$\|\mathcal{B}_{\theta_1,m}(x_{m+1}, \boldsymbol{x}_m) - \mathcal{B}_{\theta_2,m}(x_{m+1}, \boldsymbol{x}_m)\|_{\mathrm{TV}} \leq L_m^{\overleftarrow{Q}}(x_{m+1}, \boldsymbol{x}_m)\|\theta_1 - \theta_2\| \ , \tag{C.12}$$

where $L_s^{\overleftarrow{Q}}$ is defined in Lemma 58

Proof. $\mathcal{B}_{\theta,s}$ is the index version of the kernel (C.9) and thus it is Lipschitz with the same constant. □

Proposition 61. *For every $m \in [\![0, t]\!]$, we have that*

$$\left| \int \mathbb{C}_m \mathbb{S}_{m,\theta}(z_{0:m}, \mathrm{d}\boldsymbol{b}_m)\mu(\boldsymbol{b}_m)(\mathrm{Id}) \right| \leq \sum_{\ell=0}^{m-1} s_\ell^\infty \tag{C.13}$$

and

$$\left|\int \mathbb{S}_{m,\theta_1}(\boldsymbol{x}_{0:m}, \mathrm{d}\boldsymbol{b}_m)\mu(\boldsymbol{b}_m)(\mathrm{Id}) - \int \mathbb{S}_{m,\theta_2}(\boldsymbol{x}_{0:m}, \mathrm{d}\boldsymbol{b}_m)\mu(\boldsymbol{b}_m)(\mathrm{Id})\right| \leq L_m^{\mathbb{S}\mu}(\boldsymbol{x}_{0:m})\|\theta_1 - \theta_2\| \ . \quad \text{(C.14)}$$

where $L_m^{\mathbb{S}\mu}(\boldsymbol{x}_{0:m}) = N^{-1}\sum_{i=1}^N L_m^B(x_m^k, \boldsymbol{x}_{0:m})$ *and* L_m^B *is defined recursively as*

$$L_{m+1}^B(x_{m+1}^k, \boldsymbol{x}_{0:m}) = L_m^{\overleftarrow{Q}}(x_{m+1}^k, \boldsymbol{x}_m)\sum_{\ell=0}^m s_\ell^\infty + \int \mathcal{B}_{\theta,m}(x_{m+1}^k, \boldsymbol{x}_m, \mathrm{dJ})\left\{L_m^s(x_m^{\mathrm{J}}, x_{m+1}^k) + L_m^B(x_m^{\mathrm{J}}, \boldsymbol{x}_{0:m-1})\right\}. \quad \text{(C.15)}$$

In particular, under ***A****12, we have that* $L_m^B \leq \sum_{j=1}^m \|L_j^{\overleftarrow{Q}}\|_\infty \left[\sum_{\ell=0}^{m-1} s_\ell^\infty\right] + \sum_{j=1}^m \|L_j^s\|_\infty$.

Proof. Consider the following kernels,

$$\widetilde{\mathbb{S}}_{m,\theta}(\boldsymbol{x}_{0:m+1}, \mathrm{d}(\mathrm{J}_0^{i,j}, \ldots, \mathrm{J}_m^{i,j})_{i=1,j=1}^{N,M}) := \prod_{\ell=0}^m \prod_{k=1}^N \widetilde{\boldsymbol{S}}_{\ell,\theta}(x_{\ell+1}^k, \boldsymbol{x}_\ell, \mathrm{d}(\mathrm{J}_\ell^{k,j})_{j=1}^M) \ , \quad \text{(C.16)}$$

$$\widetilde{\boldsymbol{S}}_{\ell,\theta}(x_{\ell+1}^k, \boldsymbol{x}_\ell, \mathrm{d}(\mathrm{J}_\ell^{k,j})_{j=1}^M) := \prod_{j=1}^M \mathcal{B}_{\theta,\ell}(x_{\ell+1}^k, \boldsymbol{x}_\ell, \mathrm{dJ}_\ell^{k,j}) \ . \quad \text{(C.17)}$$

Define for all $k \in [1:N]$, $m \in \mathbb{N}_{>0}$,

$$B_{m+1,k} : \theta \mapsto \int \widetilde{\mathbb{S}}_{m,\theta}(\boldsymbol{x}_{0:m+1}, \mathrm{d}(\mathrm{J}_0^{i,j}, \ldots, \mathrm{J}_m^{i,j})_{i=1,j=1}^{N,M}) b_{m+1}^k(\boldsymbol{x}_{0:m+1}, (\mathrm{J}_0^{i,j}, \ldots, \mathrm{J}_m^{i,j})_{i=1,j=1}^{N,M}) \ ,$$

where $b_{m+1}^k(\boldsymbol{x}_{0:m+1}, (\mathrm{J}_0^{i,j}, \ldots, \mathrm{J}_m^{i,j})_{i=1,j=1}^{N,M})$ is defined recursively as

$$b_{m+1}^k(\boldsymbol{x}_{0:m+1}, (\mathrm{J}_0^{i,j}, \ldots, \mathrm{J}_m^{i,j})_{i=1,j=1}^{N,M}) = M^{-1}\sum_{\ell=1}^M b_m^{\mathrm{J}_m^{k,\ell}}(\boldsymbol{x}_{0:m}, (\mathrm{J}_0^{i,j}, \ldots, \mathrm{J}_{m-1}^{i,j})_{i=1,j=1}^{N,M}) + s_{m,\theta}(x_m^{\mathrm{J}_m^{k,\ell}}, x_{m+1}^k).$$

For notational convenience, we henceforth drop the arguments and simply write b_{m+1}^k.

We herebelow show that $B_{m+1,k}$ is Lipschitz with constant $L_m^B(x_{m+1}^k, \boldsymbol{x}_m)$ and bounded by $\sum_{\ell=0}^{m-1} s_\ell^\infty$.
For $m > 2$ and $k \in [1:N]$,

$$\begin{aligned}
B_{m+1,k}(\theta) &= \int \widetilde{\mathbb{S}}_{m,\theta}(\boldsymbol{x}_{0:m+1}, \mathrm{d}(\mathrm{J}_0^{i,j}, \ldots, \mathrm{J}_m^{i,j})_{i=1,j=1}^{N,M}) b_{m+1}^k \\
&= \int \cdots \int \widetilde{\mathbb{S}}_{m-1,\theta}(\boldsymbol{x}_{0:m}, \mathrm{d}(\mathrm{J}_0^{i,j}, \ldots, \mathrm{J}_{m-1}^{i,j})_{i=1,j=1}^{N,M}) \widetilde{\boldsymbol{S}}_{m,\theta}(x_{m+1}^k, \boldsymbol{x}_m, \mathrm{d}(\mathrm{J}_m^{k,j})_{j=1}^M) \\
&\qquad \times \left\{M^{-1}\sum_{\ell=1}^M b_m^{\mathrm{J}_m^{k,\ell}} + s_{m,\theta}(x_m^{\mathrm{J}_m^{k,\ell}}, x_{m+1}^k)\right\} \\
&= \int \cdots \int \widetilde{\boldsymbol{S}}_{m,\theta}(x_{m+1}^k, \boldsymbol{x}_m, \mathrm{d}\{\mathrm{J}_m^{k,j}\}_{j=1}^M)\left[M^{-1}\sum_{\ell=1}^M \left\{s_{m,\theta}(x_m^{\mathrm{J}_m^{k,\ell}}, x_{m+1}^k)\right.\right. \\
&\qquad \left.\left. + \int \widetilde{\mathbb{S}}_{m-1,\theta}(\boldsymbol{x}_{0:m}, \mathrm{d}(\mathrm{J}_0^{i,j}, \ldots, \mathrm{J}_{m-1}^{i,j})_{i=1,j=1}^{N,M}) b_m^{\mathrm{J}_m^{k,\ell}}\right\}\right] \\
&= \int \cdots \int \widetilde{\boldsymbol{S}}_{m,\theta}(x_{m+1}^k, \boldsymbol{x}_m, \mathrm{d}(\mathrm{J}_m^{k,j})_{j=1}^M)\left[M^{-1}\sum_{\ell=1}^M \left\{s_{m,\theta}(x_m^{\mathrm{J}_m^{k,\ell}}, x_{m+1}^k) + B_{m,\mathrm{J}_m^{k,\ell}}(\theta)\right\}\right] \\
&= \int \mathcal{B}_{\theta,m}(x_{m+1}^k, \boldsymbol{x}_m, \mathrm{dJ})\left\{s_{m,\theta}(x_m^{\mathrm{J}}, x_{m+1}^k) + B_{m,\mathrm{J}}(\theta)\right\}
\end{aligned}$$

Applying the induction hypothesis conditionally on $\mathrm{J}_m^{k,\ell}$, $B_{m,\mathrm{J}_m^{k,\ell}}$ is Lipschitz with constant $L_m^B(x_m^{\mathrm{J}_m^{k,\ell}}, \boldsymbol{x}_{0:m-1})$ and thus the Lipschitz constant of $B_{m+1,k}$ is

$$L_{m+1}^B(x_{m+1}^k, \boldsymbol{x}_{0:m}) = L_m^{\overleftarrow{Q}}(x_{m+1}^k, \boldsymbol{x}_m) \sum_{\ell=0}^{m} s_\ell^\infty + \int \mathcal{B}_{\theta,m}(x_{m+1}^k, \boldsymbol{x}_m, \mathrm{dJ}) \left\{ L_m^s(x_m^{\mathrm{J}}, x_{m+1}^k) + L_m^B(x_m^{\mathrm{J}}, \boldsymbol{x}_{0:m-1}) \right\} . \tag{C.18}$$

where we have used the fact that $\mathcal{B}_{\theta,m}$ and $s_{m,\theta}$ are also Lipschitz. Again by induction $B_{m+1,k}$ is bounded uniformly by $\sum_{\ell=0}^m s_\ell^\infty$. The induction is concluded by noting that for the base case $m = 0$, $\beta_m^k = 0$ for all $k \in \mathbb{N}$ and thus the result holds.

It now remains to check that for all $\theta \in \Theta$, $m \in [\![0, t]\!]$ and $k \in [1 : N]$,

$$B_{m,k}(\theta) = \int \mathbb{S}_m(\boldsymbol{x}_{0:m}, \mathrm{d}\boldsymbol{b}_m) b_m^k .$$

Again, we proceed by induction.

$$\begin{aligned}
&\int \mathbb{S}_m(\boldsymbol{x}_{0:m}, \mathrm{d}\boldsymbol{b}_m) b_m^k \\
&= \int \cdots \int \mathbb{S}_{m-1}(\boldsymbol{x}_{0:m-1}, \mathrm{d}\boldsymbol{b}_{m-1}) \boldsymbol{S}_m(\boldsymbol{b}_{m-1}, \boldsymbol{x}_{m-1:m}, \mathrm{d}\boldsymbol{b}_m) b_m^k \\
&= \int \cdots \int \mathbb{S}_{m-1}(\boldsymbol{x}_{0:m-1}, \mathrm{d}\boldsymbol{b}_{m-1}) \\
&\qquad \times \prod_{j=1}^M \left(\sum_{p=1}^N \frac{q_{m-1}(x_{m-1}^p, x_m^k)}{\sum_{\ell=1}^N q_{m-1}(x_{m-1}^\ell, x_m^k)} \delta_{x_{m-1}^p, b_{m-1}^p}(\mathrm{d}(\tilde{x}_{m-1}^{k,j}, \tilde{b}_{m-1}^{k,j})) \right) \\
&\qquad \times \left[M^{-1} \sum_{n=1}^M \left\{ \tilde{b}_{m-1}^{k,n} + s_{m,\theta}(\tilde{x}_{m-1}^{k,n}, x_m^k) \right\} \right] \\
&= \int \cdots \int \mathbb{S}_{m-1}(\boldsymbol{x}_{0:m-1}, \mathrm{d}\boldsymbol{b}_{m-1}) \\
&\qquad \times \prod_{j=1}^M \left(\sum_{p=1}^N \frac{q_{m-1}(x_{m-1}^p, x_m^k)}{\sum_{\ell=1}^N q_{m-1}(x_{m-1}^\ell, x_m^k)} \delta_p(\mathrm{dJ}_{m-1}^{k,j}) \right) \left[M^{-1} \sum_{n=1}^M \left\{ b_{m-1}^{\mathrm{J}_{m-1}^{k,n}} + s_{m,\theta}(x_{m-1}^{\mathrm{J}_{m-1}^{k,n}}, x_m^k) \right\} \right] \\
&= \int \cdots \int \widetilde{\boldsymbol{S}}_{m,\theta}(x_{m-1}^k, \boldsymbol{x}_{\ell-1}, \mathrm{d}(\mathrm{J}_{\ell-1}^{k,j})_{j=1}^M) \\
&\quad \times \left[M^{-1} \sum_{\ell=1}^M \left\{ s_{m,\theta}(x_{m-1}^{\mathrm{J}_{m-1}^{k,\ell}}, x_m^k) + \mathbb{S}_{m-1}(\boldsymbol{x}_{0:m-1}, \mathrm{d}\boldsymbol{b}_{m-1}) b_{m-1}^{\mathrm{J}_{m-1}^{k,\ell}} \right\} \right] \\
&= \int \cdots \int \widetilde{\boldsymbol{S}}_{m,\theta}(x_{m-1}^k, \boldsymbol{x}_{\ell-1}, \mathrm{d}(\mathrm{J}_{\ell-1}^{k,j})_{j=1}^M) \\
&\quad \times \left[M^{-1} \sum_{\ell=1}^M \left\{ s_{m,\theta}(x_{m-1}^{\mathrm{J}_{m-1}^{k,\ell}}, x_m^k) + \int \mathbb{S}_{m-1}(\boldsymbol{x}_{0:m-1}, \mathrm{d}\boldsymbol{b}_{m-1}) b_{m-1}^{\mathrm{J}_{m-1}^{k,\ell}} \right\} \right] \\
&= \int \cdots \int \widetilde{\boldsymbol{S}}_{m,\theta}(x_{m-1}^k, \boldsymbol{x}_{\ell-1}, \mathrm{d}(\mathrm{J}_{\ell-1}^{k,j})_{j=1}^M) \left[M^{-1} \sum_{\ell=1}^M \left\{ s_{m,\theta}(x_{m-1}^{\mathrm{J}_{m-1}^{k,\ell}}, x_m^k) + B_{m-1,\mathrm{J}_{m-1}^{k,\ell}}(\theta) \right\} \right] \\
&= B_{m,k}(\theta)
\end{aligned}$$

The proof is finalized by noting that

$$\int \mathbb{S}_m(\boldsymbol{x}_{0:m}, \mathrm{d}\boldsymbol{b}_m) \mu(\boldsymbol{b}_m)(\mathrm{Id}) = N^{-1} \sum_{k=1}^N B_{m,k}(\theta)$$

and thus it is Lipschitz with constant $L_m^{\mathbb{S}\mu}(\boldsymbol{x}_{0:m}) = N^{-1} \sum_{i=1}^N L_m^B(x_m^k, \boldsymbol{x}_{m-1})$. □

C.2.2 Lipschitz properties of Markov Kernels

Lemma 62 (Composition of ergodic Lipschitz kernels is lipschitz)**.** *Let P_θ be a Markov kernel over $X \times \mathcal{Y}$ that is uniformly π-geometrically ergodic for any θ with contraction constant ρ independent of θ and such that there exists $L_p > 0$ such that for every $x \in \mathsf{X}$*

$$\|P_{\theta_0}(x,\cdot) - P_{\theta_1}(x,\cdot)\|_{\mathrm{TV}} \leq L_P \|\theta_0 - \theta_1\|.$$

Then, for all $k > 0$

$$\left\|P^k_{\theta_0}(x,\cdot) - P^k_{\theta_1}(x,\cdot)\right\|_{\mathrm{TV}} \leq \frac{L_P}{1-\rho}\|\theta_0 - \theta_1\|.$$

Proof. We use the following decomposition borrowed from Fort et al. (2011). For any $k \geq 1$,

$$P^k_{\theta_0} f - P^k_{\theta_1} f = \sum_{j=0}^{k-1} P^j_{\theta_0}(P_{\theta_0} - P_{\theta_1})(P^{k-j-1}_{\theta_1} f - \pi f).$$

Then, for any f s.t. $\|f\|_\infty \leq 1$ and $x \in \mathsf{X}$,

$$\begin{aligned}
|P^k_{\theta_0} f(x) - P^k_{\theta_1} f(x)| &\leq \sum_{j=0}^{k-1} \left| \int P^j_{\theta_0}(x, \mathrm{d}y) \sup_{z \in \mathsf{X}} |P^{k-j-1}_{\theta_1} f(z) - \pi f| \right| L_P \|\theta_0 - \theta_1\| \\
&\leq L_P \Big(\sum_{j=0}^{k-1} \rho^{k-j-1} \Big) \|\theta_0 - \theta_1\| \\
&\leq \frac{L_P}{1-\rho} \|\theta_0 - \theta_1\|.
\end{aligned}$$

□

Lemma 63 (Composition of Lipschitz kernels is lipschitz)**.** *Let P_θ, Q_θ be two kernels defined over $X \times \mathcal{Y}$ and $Y \times \mathcal{Z}$ such that for ever $x \in X$, $y \in Y$ there are $L_p \in \mathsf{M}(X)$, $L_q \in \mathsf{M}(Y)$ that satisfy*

$$\|P_{\theta_0}(x,\cdot) - P_{\theta_1}(x,\cdot)\|_{\mathrm{TV}} \leq L_p(x)\|\theta_0 - \theta_1\|$$

and

$$\|Q_{\theta_0}(y,\cdot) - Q_{\theta_1}(y,\cdot)\|_{\mathrm{TV}} \leq L_q(y)\|\theta_0 - \theta_1\| \ .$$

Then

$$\|P_{\theta_0} Q_{\theta_0}(x,\cdot) - P_{\theta_1} Q_{\theta_1}(x,\cdot)\|_{\mathrm{TV}} \leq L_{pq}(x)\|\theta_0 - \theta_1\| \ ,$$

where $L_{pq}(x) = (\sup_\theta P_\theta L_q(x) + L_p(x) \sup_y \sup_\theta Q_\theta(y, Z))$.

Proof. Let $f \in \mathsf{M}$ such that $\|f\|_\infty \leq 1$.

$$\begin{aligned}
\|P_{\theta_1} Q_{\theta_1} f - P_{\theta_2} Q_{\theta_2} f\| &\leq \|P_{\theta_1}\left[Q_{\theta_1} f - Q_{\theta_2} f\right]\| + \|(P_{\theta_1} - P_{\theta_2}) Q_{\theta_2} f\| \\
&\leq (P_{\theta_1} L_q(x) + L_p(x)\|Q_{\theta_2} f\|_\infty)\|\theta_1 - \theta_2\| \ .
\end{aligned}$$

□

Corollary 64. *Let P_θ, Q_θ be two Markov kernels defined over $X \times \mathcal{Y}$ and $Y \times \mathcal{Z}$ such that for ever $x \in X$, $y \in Y$ there are $L_p \in \mathsf{M}(X)$, $L_q \in \mathsf{M}(Y)$ that satisfy*

$$\|P_{\theta_0}(x,\cdot) - P_{\theta_1}(x,\cdot)\|_{\mathrm{TV}} \leq L_p(x)\|\theta_0 - \theta_1\|$$

Algorithm	N	k_0	k	D_{mle}
PPG	64	0	8	0.205 ± 0.013
PPG	64	1	8	0.213 ± 0.016
PPG	64	2	8	0.201 ± 0.010
PPG	64	3	8	0.201 ± 0.010
PPG	64	4	8	0.207 ± 0.012
PPG	64	5	8	0.212 ± 0.015
PPG	64	6	8	0.210 ± 0.017
PPG	64	7	8	0.211 ± 0.018

Table C.1: Distance to θ_{MLE} for each configuration in the LGSSM case.

and

$$\|Q_{\theta_0}(y,\cdot) - Q_{\theta_1}(y,\cdot)\|_{\mathrm{TV}} \leq L_q(y)\|\theta_0 - \theta_1\| \ .$$

Then

$$\|P_{\theta_0}Q_{\theta_0}(x,\cdot) - P_{\theta_1}Q_{\theta_1}(x,\cdot)\|_{\mathrm{TV}} \leq L_{pq}(x)\|\theta_0 - \theta_1\| \ ,$$

where $L_{pq}(x) = (\sup_\theta P_\theta L_q(x) + L_p(x))$.

Lemma 65 (Product of Lipschitz kernels is lipschitz)**.** *Let P_θ, Q_θ be two Markov kernels that are uniformly Lipschitz with constants L_P, L_Q. Then $P_\theta \otimes Q_\theta$ is uniformly Lipschitz with constant $L_P + L_Q$.*

Proof. Let $h_\theta : y \mapsto \int Q_\theta(y, \mathrm{d}z) f(y,z)$. Then $(P_{\theta_i} \otimes Q_{\theta_i})(f) = P_{\theta_i}(h_{\theta_i})$ and the proof is similar to that of the previous Lemma since h_θ is Lipschitz with constant L_Q and $\|h_\theta\|_\infty \leq 1$. □

C.3 Additional numerical results

For both experiments, all the parameters were initialized by sampling from a centered multivariate gaussian distribution with covariance matrix of $0.01I$. We have used the ADAM optimizer Kingma and Ba (2015a) with a learning rate decay of $1/\sqrt{\ell}$ where ℓ is the iteration index, with a starting learning rate of 0.2. We rescale the gradients by T.

LGSSM For LGSSM we evaluated for fixed number of particles ($N = 64$) and number of gibbs iterations ($k = 8$) the influence of the burn-in phase (k_0) over the final distance obtained to the MLE estimator. Table C.1 indicates that configurations with smaller k_0 perform better. A possible interpretation of this phenomenon is that, since between two gradient ascent iterates the conditioning path is being passed on, this conditioning path from a moment on makes the estimates less biased, so the importance of having k_0 high to have less bias vanishes, but the effect of augmenting the variance with k_0 is still shown, since the fact of having a conditioning particle from the right marginal does not affect the variance of the estimator, only it's bias.

Appendix D

Appendix of Chapter 5

D.1 `SMCdiff` extension

The identity (5.15) allows us to extend `SMCdiff` Trippe et al. (2023) to handle noisy inverse problems as we now show. We have that

$$\begin{aligned}\phi_\tau^{\tilde{y}_\tau}(\underline{x}_\tau) &= \frac{\int p_\tau(\tilde{y}_\tau \frown \underline{x}_\tau | x_{\tau+1}) \left\{\prod_{s=\tau+1}^{n-1} p_s(\mathrm{d}x_s | x_{s+1})\right\} \mathsf{p}_n(\mathrm{d}x_n)}{\int \mathsf{p}_\tau(\tilde{y}_\tau \frown \underline{z}_\tau)\mathrm{d}\underline{z}_\tau} \\ &= \int b_{\tau:n}^{\tilde{y}_\tau}(\underline{x}_{\tau:n} | \overline{x}_{\tau+1:n}) f_{\tau+1:n}^{\tilde{y}_\tau}(\mathrm{d}\overline{x}_{\tau+1:n})\mathrm{d}\underline{x}_{\tau+1:n}\,,\end{aligned}$$

where

$$\begin{aligned}b_{\tau:n}(\underline{x}_{\tau:n} | \overline{x}_{\tau+1:n}) &= \frac{p_\tau(\tilde{y}_\tau \frown \underline{x}_\tau | x_{\tau+1}) \left\{\prod_{s=\tau+1}^{n-1} \underline{p}_s(\underline{x}_s | x_{s+1}) \overline{p}_s(\overline{x}_s | x_{s+1})\right\} \underline{\mathsf{p}}_n(\underline{x}_n)}{\mathsf{L}_{\tau:n}^{\tilde{y}_\tau}(\overline{x}_{\tau+1:n})}\,, \\ f_{\tau+1:n}^{\tilde{y}_\tau}(\overline{x}_{\tau+1:n}) &= \frac{\mathsf{L}_{\tau:n}^{\tilde{y}_\tau}(\overline{x}_{\tau+1:n})}{\int \mathsf{p}_\tau(\tilde{y}_\tau \frown \underline{z}_\tau)\mathrm{d}\underline{z}_\tau}\,,\end{aligned}$$

and

$$\mathsf{L}_{\tau:n}^{\tilde{y}_\tau}(\overline{x}_{\tau+1:n}) = \int p_\tau(\tilde{y}_\tau \frown \underline{z}_\tau | \overline{x}_{\tau+1} \frown \underline{z}_{\tau+1}) \left\{\prod_{s=\tau+1}^{n-1} \underline{p}_s(\mathrm{d}\underline{z}_s | \overline{x}_{s+1} \frown \underline{z}_{s+1}) \overline{p}_s(\overline{x}_s | \overline{x}_{s+1} \frown \underline{z}_{s+1})\right\} \underline{\mathsf{p}}_n(\mathrm{d}\underline{z}_n)\,.$$

Next, (5.14) implies that

$$\begin{aligned}\int \mathsf{p}_{s+1}(\overline{x}_{s+1} \frown \underline{z}_{s+1}) \underline{p}_s(\mathrm{d}\underline{z}_s | \overline{x}_{s+1} \frown \underline{z}_{s+1}) \overline{p}_s(\overline{x}_s | \overline{x}_{s+1} \frown \underline{z}_{s+1}) \mathrm{d}\underline{z}_{s:s+1} = \\ \int \mathsf{p}_s(\overline{x}_s \frown \underline{z}_s) \overline{q}_{s+1}(\overline{x}_{s+1} | \overline{x}_s) \underline{q}_{s+1}(\underline{z}_{s+1} | \underline{z}_s) \mathrm{d}\underline{z}_{s:s+1}\,,\end{aligned}$$

and applied repeatedly, we find that

$$\mathsf{L}^{\tilde{y}_\tau}(\overline{x}_{\tau+1:n}) = \int \mathsf{p}_\tau(\tilde{y}_\tau \frown \underline{x}_\tau)\mathrm{d}\underline{x}_\tau \cdot \int \delta_{\tilde{y}_\tau}(\mathrm{d}\overline{x}_\tau) \prod_{s=\tau+1}^{n} \overline{q}_s(\overline{x}_s | \overline{x}_{s-1})\,.$$

and thus, $f_{\tau:n}^{\tilde{y}_\tau}(\overline{x}_{\tau+1:n}) = \int \delta_{\tilde{y}_\tau}(\mathrm{d}\overline{x}_\tau) \prod_{s=\tau+1}^{n} \overline{q}_s(\overline{x}_s | \overline{x}_{s-1})$. In order to approximate $\phi_\tau^{\tilde{y}_\tau}$ we first diffuse the noised observation up to time n, resulting in $\overline{x}_{\tau+1:n}$, and then estimate $b_{\tau+1:n}^{\tilde{y}_\tau}(\cdot | \overline{x}_{\tau+1:n})$ using a particle filter with $\underline{p}_s(\underline{x}_s | x_{s+1})$ as transition kernel at step $s \in [\tau+1 : n]$ and $g_s : \underline{z}_s \mapsto \overline{p}_{s-1}(\overline{x}_{s-1} | \overline{x}_s \frown \underline{z}_s)$ as potential, similarly to `SMCdiff`.

D.2 Proofs

D.2.1 Proof of Proposition 33

Preliminary definitions.

We preface the proof with notations and definitions of a few quantities that will be used throughout.

For a probability measure μ and f a bounded measurable function, we write $\mu(f) := \int f(x)\mu(\mathrm{d}x)$ the expectation of f under μ and if $K(\mathrm{d}x|z)$ is a transition kernel we write $K(f)(z) := \int f(x)K(\mathrm{d}x|z)$.

Define the *smoothing* distribution

$$\phi^y_{0:n}(\mathrm{d}x_{0:n}) \propto \delta_y(\mathrm{d}\overline{x}_0)\mathsf{p}_{0:n}(x_{0:n})\mathrm{d}\underline{x}_0\mathrm{d}x_{1:n}\,, \tag{D.1}$$

which admits the posterior ϕ^y_0 as time 0 marginal. Its particle estimate known as the *poor man smoother* is given by

$$\phi^N_{0:n}(\mathrm{d}x_{0:n}) = N^{-1}\sum_{k_{0:n}\in[1:N]^{n+1}} \delta_{y^\frown \xi^{k_0}_0}(\mathrm{d}x_0)\prod_{s=1}^n \mathbb{1}\{k_s = I^{k_{s-1}}_s\}\delta_{\xi^{k_s}_s}(\mathrm{d}x_s)\,. \tag{D.2}$$

We also let $\Phi^N_{0:n}$ be the probability measure defined for any $B \in \mathcal{B}(\mathbb{R}^{\mathsf{d}_x})^{\otimes n+1}$ by

$$\Phi^N_{0:n}(B) = \mathbb{E}[\phi^N_{0:n}(B)]\,,$$

where the expectation is with respect to the probability measure

$$\begin{aligned} P^N_{0:n}(\mathrm{d}(x^{1:N}_{0:n}, a^{1:N}_{1:n})) = \prod_{i=1}^N p^y_n(\mathrm{d}x^i_n)\prod_{\ell=2}^n\left\{\prod_{j=1}^N\sum_{k=1}^N \omega^k_{\ell-1}\delta_k(\mathrm{d}a^j_\ell)p^y_{\ell-1}(\mathrm{d}x^j_{\ell-1}|x^{a^j_\ell}_\ell)\right\} \\ \times \prod_{j=1}^N\sum_{k=1}^N \omega^k_0\delta_k(\mathrm{d}a^j_1)\underline{p}^y_0(\mathrm{d}\underline{x}^j_0|x^{a^j_1}_1)\delta_y(\mathrm{d}\overline{x}^j_0)\,, \end{aligned} \tag{D.3}$$

where $\omega^i_t := \widetilde{\omega}_t(\xi^i_{t+1})/\sum_{j=1}^N \widetilde{\omega}_t(\xi^j_{t+1})$ and which corresponds to the joint law of all the random variables generated by Algorithm 1. It then follows by definition that for any $C \in \mathcal{B}(\mathbb{R}^{\mathsf{d}_x})$,

$$\int \Phi^N_{0:n}(\mathrm{d}z_{0:n})\mathbb{1}_C(z_0) = \mathbb{E}\left[\int \phi^N_{0:n}(\mathrm{d}z_{0:n})\mathbb{1}_C(z_0)\right] = \mathbb{E}[\phi^N_0(C)] = \Phi^N_0(C)\,.$$

Define also the law of the *conditional* particle cloud

$$\begin{aligned} \mathbf{P}^N(\mathrm{d}(x^{1:N}_{0:n}, a^{1:N}_{1:n})|z_{0:n}) = \delta_{z_n}(\mathrm{d}x^N_n)\prod_{i=1}^{N-1} p^y_n(\mathrm{d}x^i_n) \\ \times \prod_{\ell=2}^n \delta_{z_{\ell-1}}(\mathrm{d}x^N_{\ell-1})\delta_N(\mathrm{d}a^N_{\ell-1})\prod_{j=1}^{N-1}\sum_{k=1}^N \omega^k_{\ell-1}\delta_k(\mathrm{d}a^j_\ell)p^y_{\ell-1}(\mathrm{d}x^j_{\ell-1}|x^{a^j_\ell}_\ell) \\ \times \delta_{z_0}(\mathrm{d}x^N_0)\delta_N(\mathrm{d}a^N_1)\prod_{j=1}^{N-1}\sum_{k=1}^N \omega^k_0\delta_k(\mathrm{d}a^j_1)\underline{p}^y_0(\mathrm{d}\underline{x}^j_0|x^{a^j_1}_1)\delta_y(\mathrm{d}\overline{x}^j_0)\,. \end{aligned} \tag{D.4}$$

In what follows $\mathbb{E}_{z_{0:n}}$ refers to expectation with respect to $\mathbf{P}^N(\cdot|z_{0:n})$. Finally, for $s \in [0:n-1]$ we let Ω^N_s denote the sum of the filtering weights at step s, i.e. $\Omega^N_s = \sum_{i=1}^N \widetilde{\omega}_s(\xi^i_{s+1})$. We also write $\mathcal{Z}_0 = \int \mathsf{p}_0(x_0)\delta_y(\mathrm{d}\overline{x}_0)\mathrm{d}\underline{x}_0$ and for all $\ell \in [1:n]$, $\mathcal{Z}_\ell = \int \overline{q}_{\ell|0}(\overline{x}_\ell|y)\mathsf{p}_\ell(\mathrm{d}x_\ell)$.

The proof of Proposition 33 relies on two Lemmata stated below and proved in Section D.2.1; in Lemma 66 we provide an expression for the Radon-Nikodym derivative $\mathrm{d}\phi^y_{0:n}/\mathrm{d}\Phi^y_{0:n}$ and in Lemma 67 we explicit its leading term.

Lemma 66. *$\phi^y_{0:n}$ and $\Phi^N_{0:n}$ are equivalent and we have that*

$$\Phi^N_{0:n}(\mathrm{d}z_{0:n}) = \mathbb{E}_{z_{0:n}}\left[\frac{N^n \mathcal{Z}_0/\mathcal{Z}_n}{\prod_{s=0}^{n-1}\Omega^N_s}\right]\phi^y_{0:n}(\mathrm{d}z_{0:n})\,. \tag{D.5}$$

Lemma 67. *It holds that*

$$\begin{aligned}\frac{\mathcal{Z}_n}{\mathcal{Z}_0}\mathbb{E}_{z_{0:n}}\left[\prod_{s=0}^{n-1} N^{-1}\Omega^N_s\right] &= \left(\frac{N-1}{N}\right)^n \\ &+ \frac{(N-1)^{n-1}}{N^n}\sum_{s=1}^{n}\frac{\mathcal{Z}_s/\mathcal{Z}_0}{\overline{q}_{s|0}(\overline{z}_s|y)}\int p_{0|s}(x_0|z_s)\delta_y(\mathrm{d}\overline{x}_0)\mathrm{d}\underline{x}_0 + \frac{\mathsf{D}^y_{0:n}}{N^2}\,.\end{aligned} \tag{D.6}$$

where $\mathsf{D}^y_{0:n}$ is a positive constant.

Before proceeding with the proof of Proposition 33, let us note that having $z \mapsto \widetilde{\omega}_\ell(z)$ bounded on $\mathbb{R}^{\mathsf{d}_x}$ for all $\ell \in [0:n-1]$ is sufficient to guarantee that $\mathsf{C}^y_{0:n}$ and $\mathsf{D}^y_{0:n}$ are finite since in this case it follows immediately that $\mathbb{E}_{z_{0:n}}\left[\prod_{s=0}^{n-1} N^{-1}\Omega^N_s\right]$ is bounded and so is the right hand side of (D.6). This can be achieved with a slight modification of (5.9) and (5.10). Indeed, consider instead the following recursion for $s \in [0:n]$ where $\delta > 0$,

$$\begin{aligned}\phi^y_n(x_n) &\propto (\overline{q}_{n|0}(\overline{x}_n|y) + \delta)\mathsf{p}_n(x_n)\,,\\ \phi^y_s(x_s) &\propto \int \phi^y_{s+1}(x_{s+1})p_s(\mathrm{d}x_s|x_{s+1})\frac{\overline{q}_s(\overline{x}_s|y)+\delta}{\overline{q}_{s+1}(\overline{x}_{s+1}|y)+\delta}\mathrm{d}x_{s+1}\,.\end{aligned}$$

Then we have that

$$\phi^y_0(\underline{x}_0) \propto \int \phi^y_1(x_1)\underline{p}_0(\underline{x}_0|x_1)\frac{\overline{p}_0(y|x_1)}{\overline{q}_{1|0}(\overline{x}_1|y)+\delta}\mathrm{d}x_1\,.$$

We can then use Algorithm 1 to produce a particle approximation of ϕ^y_0 using the following transition and weight function,

$$\begin{aligned}p^{y,\delta}_s(x_s|x_{s+1}) &= \frac{\gamma_s(y|x_{s+1})}{\gamma_s(y|x_{s+1})+\delta}p^y_s(x_s|x_{s+1}) + \frac{\delta}{\gamma_s(y|x_{s+1})+\delta}p_s(x_s|x_{s+1})\,,\\ \widetilde{\omega}_s(x_{s+1}) &= (\gamma_s(y|x_{s+1})+\delta)/(\overline{q}_{s+1|0}(\overline{x}_{s+1}|y)+\delta)\,,\end{aligned}$$

where $\gamma_s(y|x_{s+1}) = \int \overline{q}_{s|0}(\overline{x}_s|y)p_s(x_s|x_{s+1})\mathrm{d}x_s$ is available in closed form and p^y_s is defined in (5.7). $\widetilde{\omega}_s$ is thus clearly bounded for all $s \in [0:n-1]$ and it is still possible to sample from $p^{y,\delta}_s$ since it is simply a mixture between the transition (5.7) and the "prior" transition.

Proof of Proposition 33. Consider the *forward* Markov kernel

$$\overrightarrow{\mathbf{B}}_{1:n}(z_0, \mathrm{d}z_{1:n}) = \frac{\mathsf{p}_{1:n}(\mathrm{d}z_{1:n})p_0(z_0|z_1)}{\int \mathsf{p}_{1:n}(\mathrm{d}\tilde{z}_{1:n})p_0(\tilde{z}_0|\tilde{z}_1)}\,, \tag{D.7}$$

which satisfies

$$\phi^y_{0:n}(\mathrm{d}z_{0:n}) = \phi^y_0(\mathrm{d}z_0)\overrightarrow{\mathbf{B}}_{1:n}(z_0, \mathrm{d}z_{1:n})\,.$$

By Lemma 66 we have for any $C \in \mathcal{B}(\mathbb{R}^{\mathsf{d}_x})$ that

$$\begin{aligned}\Phi^N_0(C) &= \int \Phi^N_{0:n}(\mathrm{d}z_{0:n})\mathbb{1}_C(z_0)\\ &= \int \mathbb{1}_C(z_0)\mathbb{E}_{z_{0:n}}\left[\frac{N^n\mathcal{Z}_0/\mathcal{Z}_n}{\prod_{s=0}^{n-1}\Omega^N_s}\right]\phi^y_{0:n}(\mathrm{d}z_{0:n})\\ &= \int \mathbb{1}_C(z_0)\int \overrightarrow{\mathbf{B}}_{1:n}(z_0,\mathrm{d}z_{1:n})\mathbb{E}_{z_{0:n}}\left[\frac{N^n\mathcal{Z}_0/\mathcal{Z}_n}{\prod_{s=0}^{n-1}\Omega^N_s}\right]\phi^y_0(\mathrm{d}z_0)\,,\end{aligned}$$

which shows that the Radon-Nikodym derivative $\mathrm{d}\Phi_0^N/\mathrm{d}\phi_0^y$ is,

$$\frac{\mathrm{d}\Phi_0^N}{\mathrm{d}\phi_0^y}(z_0) = \int \overrightarrow{\mathbf{B}}_{1:n}(z_0, \mathrm{d}z_{1:n}) \mathbb{E}_{z_{0:n}} \left[\frac{N^n \mathcal{Z}_0/\mathcal{Z}_n}{\prod_{s=0}^{n-1} \Omega_s^N} \right] .$$

Applying Jensen's inequality twice yields

$$\frac{\mathrm{d}\Phi_0^N}{\mathrm{d}\phi_0^y}(z_0) \geq \frac{N^n \mathcal{Z}_0/\mathcal{Z}_n}{\int \overrightarrow{\mathbf{B}}_{1:n}(z_0, \mathrm{d}z_{1:n}) \mathbb{E}_{z_{0:n}} \left[\prod_{s=0}^{n-1} \Omega_s^N \right]} ,$$

and it then follows that

$$\mathsf{KL}(\phi_0^y \parallel \Phi_0^N) \leq \int \log \left(\frac{\mathcal{Z}_n}{\mathcal{Z}_0} \int \overrightarrow{\mathbf{B}}_{1:n}(z_0, \mathrm{d}z_{1:n}) \mathbb{E}_{z_{0:n}} \left[\prod_{s=0}^{n-1} N^{-1} \Omega_s^N \right] \right) \phi_0^y(\mathrm{d}z_0) .$$

Finally, using Lemma 67 and the fact that $\log(1+x) < x$ for $x > 0$ we get

$$\mathsf{KL}(\phi_0^y \parallel \Phi_0^N) \leq \frac{\mathsf{C}_{0:n}^y}{N-1} + \frac{\mathsf{D}_{0:n}^y}{N^2}$$

where

$$\mathsf{C}_{0:n}^y := \sum_{s=1}^{n} \int \frac{\mathcal{Z}_s/\mathcal{Z}_0}{\overline{q}_{s|0}(\overline{z}_s|y)} \left(p_{0|s}(x_0|z_s) \delta_y(\mathrm{d}\overline{x}_0) \mathrm{d}\underline{x}_0 \right) \phi_s^y(\mathrm{d}z_s) ,$$

and $\phi_s^y(z_s) \propto \mathsf{p}_s(z_s) \int p_{0|s}(z_0|z_s) \delta_y(\mathrm{d}\underline{z}_0) \mathrm{d}\overline{z}_0$. □

Proof of Lemma 66 and Lemma 67

Proof of Lemma 66. We have that

$$\begin{aligned}
&\Phi_{0:n}^N(\mathrm{d}z_{0:n}) \\
&= N^{-1} \int P_{0:n}^N(\mathrm{d}x_{0:n}^{1:N}, \mathrm{d}a_{1:n}^{1:N}) \sum_{k_{0:n} \in [1:N]^{n+1}} \delta_{y^\frown \underline{x}_0^{k_0}}(\mathrm{d}z_0) \prod_{s=1}^{n} \mathbb{1}\{k_s = a_s^{k_{s-1}}\} \delta_{x_s^{k_s}}(\mathrm{d}z_s) \\
&= N^{-1} \int \sum_{k_{0:n}} \sum_{a_{1:n}^{1:N}} \delta_{y^\frown \underline{x}_0^{k_0}}(\mathrm{d}z_0) \prod_{s=1}^{n} \mathbb{1}\{k_s = a_s^{k_{s-1}}\} \delta_{x_s^{k_s}}(\mathrm{d}z_s) \\
&\quad \times \prod_{j=1}^{N} p_n^y(\mathrm{d}x_n^j) \left\{ \prod_{\ell=2}^{n} \prod_{i=1}^{N} \omega_{\ell-1}^{a_\ell^i} p_{\ell-1}^y(\mathrm{d}x_{\ell-1}^i | x_\ell^{a_\ell^i}) \right\} \prod_{r=1}^{N} \omega_0^{a_1^r} \underline{p}_{\ell-1}^y(\mathrm{d}\underline{x}_0^r | x_1^{a_1^r}) \delta_y(\overline{x}_0^r) \\
&= N^{-1} \int \sum_{k_{0:n}} \sum_{a_{1:n}^{1:N}} p_n^y(\mathrm{d}x_n^{k_n}) \delta_{x_n^{k_n}}(\mathrm{d}z_n) \prod_{j \neq k_n} p_n^y(\mathrm{d}x_n^j) \prod_{\ell=2}^{n} \Bigg\{ \prod_{i \neq k_{\ell-1}} \omega_{\ell-1}^{a_\ell^i} p_{\ell-1}^y(\mathrm{d}x_{\ell-1}^i | x_\ell^{a_\ell^i}) \\
&\quad \times \mathbb{1}\{a_\ell^{k_{\ell-1}} = k_\ell\} \frac{\widetilde{\omega}_{\ell-1}(x_\ell^{a_\ell^{k_{\ell-1}}})}{\Omega_{\ell-1}^N} p_{\ell-1}^y(\mathrm{d}x_\ell^{k_{\ell-1}} | x_\ell^{a_\ell^{k_{\ell-1}}}) \delta_{x_{\ell-1}^{k_{\ell-1}}}(\mathrm{d}z_{\ell-1}) \Bigg\} \\
&\quad \times \left\{ \prod_{r \neq k_0} \omega_0^{a_1^r} \underline{p}_0^y(\mathrm{d}\underline{x}_0^r | x_1^{a_1^r}) \delta_y(\mathrm{d}\overline{x}_0^r) \right\} \mathbb{1}\{a_1^{k_0} = k_1\} \frac{\widetilde{\omega}_0(x_1^{a_1^{k_0}})}{\Omega_0^N} p_0^y(\mathrm{d}\underline{x}_0^{k_0} | x_0^{a_1^{k_0}}) \delta_{y^\frown \underline{x}_0^{k_0}}(\mathrm{d}z_0) .
\end{aligned}$$

Then, using that for all $s \in [2:n]$

$$\widetilde{\omega}_{s-1}(x_s^{k_s}) p_{s-1}^y(\mathrm{d}x_{s-1}^{k_{s-1}} | x_s^{k_s}) = \frac{\overline{q}_{s-1|0}(\overline{x}_{s-1}^{k_{s-1}} | y)}{\overline{q}_{s|0}(\overline{x}_s^{k_s} | y)} p_s(\mathrm{d}x_{s-1}^{k_{s-1}} | x_s^{k_s}) ,$$

we recursively get that

$$
\begin{aligned}
p_n^y(\mathrm{d}x_n^{k_n})\delta_{x_n^{k_n}}(\mathrm{d}z_n)\prod_{s=2}^{n}\mathbb{1}\{a_s^{k_{s-1}}=k_s\}\frac{\tilde{\omega}_{s-1}(x_s^{a_s^{k_{s-1}}})}{\Omega_{s-1}^N}p_{s-1}^y(\mathrm{d}x_{s-1}^{k_{s-1}}|x_s^{a_s^{k_{s-1}}})\delta_{x_{s-1}^{k_{s-1}}}(\mathrm{d}z_{s-1}) \\
\times\mathbb{1}\{a_1^{k_0}=k_1\}\frac{\tilde{\omega}_0(x_1^{a_1^{k_0}})}{\Omega_0^N}p_0^y(\mathrm{d}\underline{x}_0^{k_0}|x_1^{a_1^{k_0}})\delta_{y^\frown\underline{x}_0^{k_0}}(\mathrm{d}z_0) \\
=\frac{\overline{q}_{n|0}(z_n|y)\mathsf{p}_n(\mathrm{d}z_n)}{\mathcal{Z}_n}\delta_{z_n}(\mathrm{d}x_n^{k_n})\prod_{s=2}^{n}\mathbb{1}\{a_s^{k_{s-1}}=k_s\}\frac{\overline{q}_{s-1|0}(\overline{z}_{s-1}|y)}{\Omega_{s-1}^N\overline{q}_{s|0}(\overline{z}_s|y)}p_{s-1}(\mathrm{d}z_{s-1}|z_s)\delta_{z_{s-1}}(\mathrm{d}x_{s-1}^{k_{s-1}}) \\
\times\mathbb{1}\{a_1^{k_0}=k_1\}\frac{\overline{p}_0(y|z_1)}{\Omega_0^N\overline{q}_{1|0}(\overline{z}_1|y)}p_0(\mathrm{d}\underline{z}_0|z_1)\delta_y(\mathrm{d}\overline{z}_0)\delta_{z_0}(\mathrm{d}x_0^{k_0}) \\
=\frac{\mathcal{Z}_0}{\mathcal{Z}_n}\phi_{0:n}^y(\mathrm{d}z_{0:n})\delta_{z_n}(\mathrm{d}x_n^{k_n})\prod_{s=1}^{n}\mathbb{1}\{a_s^{k_{s-1}}=k_s\}\frac{1}{\Omega_{s-1}^N}\delta_{z_{s-1}}(\mathrm{d}x_{s-1}^{k_{s-1}})\,.
\end{aligned}
$$

Thus, we obtain

$$
\begin{aligned}
\Phi_{0:n}^N(\mathrm{d}z_{0:n}) &= N^{-1}\int\sum_{k_{0:n}}\sum_{a_{1:n}^{1:N}}\phi_{0:n}^y(\mathrm{d}z_{0:n})\frac{\mathcal{Z}_0/\mathcal{Z}_n}{\prod_{s=0}^{n-1}\Omega_s^N}\delta_{z_n}(\mathrm{d}x_n^{k_n})\prod_{j\neq k_n}p_n^y(\mathrm{d}x_n^j) \\
&\quad\times\prod_{\ell=2}^{n}\mathbb{1}\{a_\ell^{k_{\ell-1}}=k_\ell\}\delta_{z_{\ell-1}}(\mathrm{d}x_{\ell-1}^{k_{\ell-1}})\prod_{i\neq k_{\ell-1}}\omega_{\ell-1}^{a_\ell^i}p_{\ell-1}^y(\mathrm{d}x_{\ell-1}^i|x_\ell^{a_\ell^i}) \\
&\quad\times\mathbb{1}\{a_1^{k_0}=k_1\}\delta_{z_0}(\mathrm{d}x_0^{k_0})\prod_{i\neq k_0}\omega_0^{a_1^i}\underline{p}_0(\underline{x}_0^i|x_1^{a_1^i})\delta_y(\mathrm{d}\overline{x}_0^i) \\
&= N^{-1}\sum_{k_{0:n}}\phi_{0:n}^y(\mathrm{d}z_{0:n})\mathbb{E}_{z_{0:n}}^{k_{0:n}}\left[\frac{\mathcal{Z}_0/\mathcal{Z}_n}{\prod_{s=0}^{n-1}\Omega_s^N}\right],
\end{aligned}
$$

where for all $k_{0:n}\in[1:N]^{n+1}$ $\mathbb{E}_{z_{0:n}}^{k_{0:n}}$ denotes the expectation under the Markov kernel

$$
\begin{aligned}
\mathbf{P}_{k_{0:n}}^N(\mathrm{d}(x_{0:n}^{1:N},a_{1:n}^{1:N})|z_{0:n}) &= \delta_{z_n}(\mathrm{d}x_n^{k_n})\prod_{i\neq k_n}p_n^y(\mathrm{d}x_n^i) \\
&\quad\times\prod_{\ell=2}^{n}\delta_{z_{\ell-1}}(\mathrm{d}x_{\ell-1}^{k_{\ell-1}})\delta_{k_\ell}(\mathrm{d}a_\ell^{k_{\ell-1}})\prod_{j\neq k_{\ell-1}}\sum_{k=1}^{N}\omega_{\ell-1}^k\delta_k(\mathrm{d}a_\ell^j)p_{\ell-1}^y(\mathrm{d}x_{\ell-1}^j|x_\ell^{a_\ell^j}) \\
&\quad\times\delta_{z_0}(\mathrm{d}x_0^{k_0})\delta_{k_1}(\mathrm{d}a_1^{k_0})\prod_{j\neq k_0}\sum_{k=1}^{N}\omega_0^k\delta_k(\mathrm{d}a_1^j)\underline{p}_0^y(\mathrm{d}\underline{x}_0^j|x_1^{a_1^j})\delta_y(\mathrm{d}\overline{x}_0)\,.
\end{aligned}
$$

Note however that for all $(k_{0:n},\ell_{0:n})\in([1:N]^{n+1})^2$,

$$
\mathbb{E}_{z_{0:n}}^{k_{0:n}}\left[\frac{1}{\prod_{s=0}^{n-1}\Omega_s^N}\right]=\mathbb{E}_{z_{0:n}}^{\ell_{0:n}}\left[\frac{1}{\prod_{s=0}^{n-1}\Omega_s^N}\right]
$$

and thus it follows that

$$
\Phi_{0:n}^N(\mathrm{d}z_{0:n})=\mathbb{E}_{z_{0:n}}\left[\frac{N^n\mathcal{Z}_0/\mathcal{Z}_n}{\prod_{s=0}^{n-1}\Omega_s^N}\right]\phi_{0:n}^y(\mathrm{d}z_{0:n})\,. \tag{D.8}
$$

□

Denote by $\{\mathcal{F}_s\}_{s=0}^n$ the filtration generated by a conditional particle cloud sampled from the kernel $\mathbf{P}^N$ (D.4), i.e. for all $\ell\in[0:n-1]$

$$
\mathcal{F}_s=\sigma(\xi_{s:n}^{1:N},I_{s+1:n}^{1:N})\,.
$$

and $\mathcal{F}_n = \sigma(\xi_n^{1:N})$. Define for all bounded f and $\ell \in [0:n-1]$

$$\gamma_{\ell:n}^N(f) = \left\{ \prod_{s=\ell+1}^{n-1} N^{-1}\Omega_s^N \right\} N^{-1} \sum_{k=1}^N \widetilde{\omega}_\ell(\xi_{\ell+1}^k) f(\xi_{\ell+1}^k)\,, \tag{D.9}$$

with the convention $\gamma_{\ell:n}^N(f) = 1$ if $\ell \geq n$. Define also the transition Kernel

$$Q_{\ell-1|\ell+1}^y : \mathbb{R}^{\mathsf{d}_x} \times \mathcal{B}(\mathbb{R}^{\mathsf{d}_x}) \ni (x_{\ell+1}, A) \mapsto \int \mathbb{1}_A(x_\ell)\widetilde{\omega}_{\ell-1}(x_\ell) p_\ell^y(\mathrm{d}x_\ell|x_{\ell+1})\,. \tag{D.10}$$

Using eqs. (5.7) and (5.8), it is easily seen that for all $\ell \in [0:n-1]$,

$$\widetilde{\omega}_\ell(x_{\ell+1}) Q_{\ell-1|\ell+1}^y(f)(x_{\ell+1}) = \frac{1}{\overline{q}_{\ell+1|0}(\overline{x}_{\ell+1}|y)} \int \overline{q}_{\ell|0}(\overline{x}_s|y)\widetilde{\omega}_{\ell-1}(x_\ell) f(x_\ell) p_\ell(\mathrm{d}x_\ell|x_{\ell+1})\,. \tag{D.11}$$

Define $\mathbf{1} : x \in \mathbb{R}^{\mathsf{d}_x} \mapsto 1$. We may thus write that $\gamma_{\ell:n}^N(f) = N^{-1}\gamma_{\ell+1:n}^N(\mathbf{1}) \sum_{k=1}^N \widetilde{\omega}_\ell(\xi_{\ell+1}^k) f(\xi_{\ell+1}^k)$.

Lemma 68. *For all $\ell \in [0:n-1]$ it holds that*

$$\mathbb{E}_{z_{0:n}}\left[\gamma_{\ell-1:n}^N(f)\right] = \frac{N-1}{N}\mathbb{E}_{z_{0:n}}\left[\gamma_{\ell:n}^N\left(Q_{\ell-1|\ell+1}^y(f)\right)\right] + \frac{1}{N}\mathbb{E}_{z_{0:n}}\left[\gamma_{\ell:n}^N(\mathbf{1})\right]\widetilde{\omega}_{\ell-1}(z_\ell) f(z_\ell)\,.$$

Proof. By the tower property and the fact that $\gamma_{\ell:n}^N(f)$ is $\mathcal{F}_{\ell+1}$-measurable, we have that

$$\mathbb{E}_{z_{0:n}}\left[\gamma_{\ell-1:n}^N(f)\right] = \mathbb{E}_{z_{0:n}}\left[N^{-1}\gamma_{\ell+1:n}^N(\mathbf{1})\Omega_\ell^N \mathbb{E}_{z_{0:n}}\left[N^{-1}\sum_{k=1}^N \widetilde{\omega}_{\ell-1}(\xi_\ell^k) f(\xi_\ell^k) \middle| \mathcal{F}_{\ell+1}\right]\right]\,.$$

Note that for all $\ell \in [0:n-1]$, $(\xi_\ell^1, \ldots, \xi_\ell^{N-1})$ are identically distributed conditionally on $\mathcal{F}_{\ell+1}$ and

$$\mathbb{E}_{z_{0:n}}\left[\widetilde{\omega}_{\ell-1}(\xi_\ell^j) f(\xi_\ell^j) \middle| \mathcal{F}_{\ell+1}\right] = \frac{1}{\Omega_\ell^N}\sum_{k=1}^N \widetilde{\omega}_\ell(\xi_{\ell+1}^k) \int \widetilde{\omega}_{\ell-1}(x_\ell) f(x_\ell) p_\ell^y(\mathrm{d}x_\ell|\xi_{\ell+1}^k)\,,$$

leading to

$$\begin{aligned}
&\mathbb{E}_{z_{0:n}}\left[N^{-1}\sum_{k=1}^N \widetilde{\omega}_{\ell-1}(\xi_\ell^k) f(\xi_\ell^k) \middle| \mathcal{F}_{\ell+1}\right] \\
&\qquad = \frac{N-1}{N\Omega_\ell^N}\sum_{k=1}^N \widetilde{\omega}_\ell(\xi_{\ell+1}^k) \int \widetilde{\omega}_{\ell-1}(x_\ell) f(x_\ell) p_\ell^y(\mathrm{d}x_\ell|\xi_{\ell+1}^k) + \frac{1}{N}\widetilde{\omega}_{\ell-1}(z_\ell) f(z_\ell)\,,
\end{aligned}$$

and the desired recursion follows. □

Proof of Lemma 67. We proceed by induction and show for all $\ell \in [0:n-2]$

$$\begin{aligned}
&\mathbb{E}_{z_{0:n}}\left[\gamma_{\ell:n}^N(f)\right] \\
&= \left(\frac{N-1}{N}\right)^{n-\ell} \frac{\int \mathsf{p}_{\ell+1}(\mathrm{d}x_{\ell+1})\overline{q}_{\ell+1|0}(\overline{x}_{\ell+1}|y)\widetilde{\omega}_\ell(x_{\ell+1}) f(x_{\ell+1})}{\mathcal{Z}_n} \\
&+ \frac{(N-1)^{n-\ell-1}}{N^{n-\ell}}\left[(\mathcal{Z}_{\ell+1}/\mathcal{Z}_n) f(z_{\ell+1})\widetilde{\omega}_\ell(z_{\ell+1}) \right. \\
&\left. + \sum_{s=\ell+2}^n \frac{\mathcal{Z}_s/\mathcal{Z}_n}{\overline{q}_{s|0}(\overline{z}_s|y)} \int \widetilde{\omega}_\ell(x_{\ell+1})\overline{q}_{\ell+1|0}(\overline{x}_{\ell+1}|y) f(x_{\ell+1}) p_{\ell+1|s}(\mathrm{d}x_{\ell+1}|z_s)\right] + \frac{\mathsf{D}_{\ell:n}^y}{N^2}\,.
\end{aligned} \tag{D.12}$$

where f is a bounded function and $\mathsf{D}^y_{\ell:n}$ is a a positive constant. The desired result in Lemma 67 then follows by taking $\ell = 0$ and $f = \mathbf{1}$.

Assume that (D.12) holds at step ℓ. To show that it holds at step $\ell - 1$ we use Lemma 68 and we compute $\mathbb{E}_{z_{0:n}}\left[\gamma^N_{\ell:n}\left(Q^y_{\ell-1|\ell+1}(f)\right)\right]$ and $\mathbb{E}_{z_{0:n}}\left[\gamma^N_{\ell:n}(\mathbf{1})\right]\widetilde{\omega}_{\ell-1}(z_\ell)f(z_\ell)$.

Using the following identities which follow from (D.11)

$$\int \overline{q}_{\ell+1|0}(\overline{x}_{\ell+1}|y)\widetilde{\omega}_\ell(x_{\ell+1})Q^y_{\ell-1|\ell+1}(f)(x_{\ell+1})\mathsf{p}_{\ell+1}(\mathrm{d}x_{\ell+1}) \\ = \int \overline{q}_{\ell|0}(\overline{x}_\ell|y)\widetilde{\omega}_{\ell-1}(x_\ell)f(x_\ell)\mathsf{p}_\ell(\mathrm{d}x_\ell)\,,$$

and

$$\int \widetilde{\omega}_\ell(x_{\ell+1})\overline{q}_{\ell+1|0}(\overline{x}_{\ell+1}|y)Q^y_{\ell-1|\ell+1}(f)(x_{\ell+1})p_{\ell+1|s}(\mathrm{d}x_{\ell+1}|x_s) \\ = \int \widetilde{\omega}_{\ell-1}(x_\ell)\overline{q}_{\ell|0}(\overline{x}_\ell|y)f(x_\ell)p_{\ell|s}(\mathrm{d}x_\ell|x_s)\,,$$

we get by (D.12) that

$$\begin{aligned}
&\frac{N-1}{N}\mathbb{E}_{z_{0:n}}\left[\gamma^N_{\ell:n}\left(Q^y_{\ell-1|\ell+1}(f)\right)\right] \\
&\quad= \left(\frac{N-1}{N}\right)^{n-\ell+1}\frac{\int \overline{q}_{\ell|0}(\overline{x}_\ell|y)\widetilde{\omega}_{\ell-1}(x_\ell)f(x_\ell)\mathsf{p}_\ell(\mathrm{d}x_\ell)}{\mathcal{Z}_n} \\
&\qquad+ \frac{(N-1)^{n-\ell}}{N^{n-\ell+1}}\Bigg[\frac{\mathcal{Z}_{\ell+1}/\mathcal{Z}_n}{\overline{q}_{\ell+1|0}(\overline{z}_{\ell+1}|y)}\int \overline{q}_{\ell|0}(\overline{x}_\ell|y)\widetilde{\omega}_{\ell-1}(x_\ell)f(x_\ell)p_\ell(\mathrm{d}x_\ell|z_{\ell+1}) \\
&\qquad+ \sum_{s=\ell+2}^{n}\frac{\mathcal{Z}_s/\mathcal{Z}_n}{\overline{q}_{s|0}(\overline{z}_s|y)}\int \widetilde{\omega}_{\ell-1}(x_\ell)\overline{q}_{\ell|0}(\overline{x}_\ell|y)f(x_\ell)p_{\ell|s}(\mathrm{d}x_\ell|z_s)\Bigg] + \frac{\mathsf{D}^y_{\ell:n}}{N^2} \\
&\quad= \left(\frac{N-1}{N}\right)^{n-\ell+1}\frac{\int \overline{q}_{\ell|0}(\overline{x}_\ell|y)\widetilde{\omega}_{\ell-1}(x_\ell)f(x_\ell)\mathsf{p}_\ell(\mathrm{d}x_\ell)}{\mathcal{Z}_n} \\
&\qquad+ \frac{(N-1)^{n-\ell}}{N^{n-\ell+1}}\sum_{s=\ell+1}^{n}\frac{\mathcal{Z}_s/\mathcal{Z}_n}{\overline{q}_{s|0}(\overline{z}_s|y)}\int \widetilde{\omega}_{\ell-1}(x_\ell)\overline{q}_{\ell|0}(\overline{x}_s|y)f(x_\ell)p_{\ell|s}(\mathrm{d}x_\ell|z_s) + \frac{\mathsf{D}^y_{\ell:n}}{N^2}\,.
\end{aligned} \tag{D.13}$$

The induction step is finished by using again (D.12) and noting that

$$\frac{1}{N}\mathbb{E}_{z_{0:n}}\left[\gamma^N_{\ell:n}(\mathbf{1})\right]\widetilde{\omega}_{\ell-1}(z_\ell)f(z_\ell) = \frac{(N-1)^{n-\ell}}{N^{n-\ell+1}}(\mathcal{Z}_\ell/\mathcal{Z}_n)\widetilde{\omega}_{\ell-1}(z_\ell)f(z_\ell) + \frac{\widetilde{\mathsf{D}}^y_{\ell:n}}{N^2}\,.$$

and then setting $\mathsf{D}^y_{\ell-1:n} = \mathsf{D}^y_{\ell:n} + \widetilde{\mathsf{D}}^y_{\ell:n}$.

It remains to compute the initial value at $\ell = n-2$. Note that

$$\mathbb{E}_{z_{0:n}}\left[\gamma^N_{n-1:n}(f)\right] = \frac{N-1}{N}\int p^y_n(\mathrm{d}x_n)\widetilde{\omega}_{n-1}(x_n)f(x_n) + \frac{1}{N}\widetilde{\omega}_{n-1}(z_n)f(z_n) \tag{D.14}$$

and thus by Lemma 68 and similarly to the previous computations

$$
\begin{aligned}
&\mathbb{E}_{z_{0:n}}\left[\gamma^N_{n-2:n}(f)\right] \\
&= \left(\frac{N-1}{N}\right)^2 \int p^y_n(\mathrm{d}x_n)\widetilde{\omega}_{n-1}(x_n)Q^y_{n-2|n}(f)(x_n) + \frac{N-1}{N^2}\Big[\widetilde{\omega}_{n-1}(z_n)Q^y_{n-2|n}(f)(z_n) \\
&+ \widetilde{\omega}_{n-2}(z_{n-1})f(z_{n-1})\int p^y_n(\mathrm{d}x_n)\widetilde{\omega}_{n-1|n}(x_n)\Big] + \frac{\mathsf{D}^y_{n-2fa:n}}{N^2} \\
&= \left(\frac{N-1}{N}\right)^2 \frac{\int \overline{q}_{n-1|0}(x_{n-1}|y)\widetilde{\omega}_{n-2}(x_{n-1})\mathsf{p}_{n-1}(\mathrm{d}x_{n-1})}{\mathcal{Z}_n} \\
&+ \frac{N-1}{N^2}\Big[(\mathcal{Z}_{n-1}/\mathcal{Z}_n)\widetilde{\omega}_{n-2}(z_{n-1})f(z_{n-1}) \\
&+ \frac{1}{\overline{q}_{n|0}(\overline{x}_n|y)}\int \overline{q}_{n-1|0}(\overline{x}_{n-1}|y)\widetilde{\omega}_{n-2}(x_{n-1})f(x_{n-1})p_{n-1}(\mathrm{d}x_{n-1}|z_n)\Big] + \frac{\mathsf{D}^y_{n-2:n}}{N^2}\,.
\end{aligned}
$$

□

D.2.2 Proof of Proposition 34 and Lemma 69

In this section and only in this section we make the following assumption

(**A2**) For all $s \in [0:n-1]$, $\mathfrak{p}_s(x_s)q_{s+1}(x_{s+1}|x_s) = \mathfrak{p}_{s+1}(x_{s+1})\lambda_s(x_s|x_{s+1})$.

We also consider $\sigma_\delta = 0$. In what follows we let $\tau_{\mathsf{d_y}+1} = n$ and we write $\tau_{1:\mathsf{d_y}} = \{\tau_1, \ldots, \tau_{\mathsf{d_y}}\}$ and $\overline{\tau_{1:\mathsf{d_y}}} = [1:n] \setminus \tau_{1:t}$. Define the measure

$$
\Gamma^{\mathbf{y}}_{0:n}(\mathrm{d}\mathbf{x}_{0:n}) = \mathfrak{p}_n(\mathrm{d}\mathbf{x}_n)\prod_{s\in\overline{\tau_{1:\mathsf{d_y}}}}\lambda_s(\mathrm{d}\mathbf{x}_s|\mathbf{x}_{s+1})\prod_{i=1}^{\mathsf{d_y}}\lambda_{\tau_i}(\mathbf{x}_{\tau_i}|\mathbf{x}_{\tau_i+1})\mathrm{d}\mathbf{x}^{\setminus i}_{\tau_i}\delta_{\mathbf{y}[i]}(\mathrm{d}\mathbf{x}_{\tau_i}[i])\,. \tag{D.15}
$$

Under (**A2**) it has the following alternative *forward* expression,

$$
\Gamma^{\mathbf{y}}_{0:n}(\mathrm{d}\mathbf{x}_{0:n}) = \mathfrak{p}_0(\mathrm{d}\mathbf{x}_0)\prod_{s\in\overline{\tau_{1:\mathsf{d_y}}}}q_{s+1}(\mathrm{d}\mathbf{x}_{s+1}|\mathbf{x}_s)\prod_{i=1}^{\mathsf{d_y}}q_{\tau_i}(\mathbf{x}_{\tau_i}|\mathbf{x}_{\tau_i-1})\mathrm{d}\mathbf{x}^{\setminus i}_{\tau_i}\delta_{\mathbf{y}[i]}(\mathrm{d}\mathbf{x}_{\tau_i}[i])\,. \tag{D.16}
$$

Since the forward kernels decompose over the dimensions of the states, i.e.

$$
q_{s+1}(\mathbf{x}_{s+1}|\mathbf{x}_s) = \prod_{\ell=1}^{\mathsf{d}_x} q^\ell_{s+1}(\mathbf{x}_{s+1}[\ell]|\mathbf{x}_s[\ell])
$$

where $q^\ell_{s+1}(\mathbf{x}_{s+1}[\ell]|\mathbf{x}_s[\ell]) = \mathcal{N}(\mathbf{x}_{s+1}[\ell]; (\alpha_{s+1}/\alpha_s)^{1/2}\mathbf{x}_s[\ell], 1-(\alpha_{s+1}/\alpha_s))$, we can write

$$
\Gamma^{\mathbf{y}}_{0:n}(\mathbf{x}_{0:n}) = \mathfrak{p}_0(\mathbf{x}_0)\prod_{\ell=1}^{\mathsf{d}_x}\Gamma^{\mathbf{y}}_{1:n|0,\ell}(\mathbf{x}_1[\ell], \ldots, \mathbf{x}_n[\ell]|\mathbf{x}_0[\ell])\,, \tag{D.17}
$$

where for $\ell \in [1:\mathsf{d_y}]$

$$
\Gamma^{\mathbf{y}}_{1:n|0,\ell}(\mathbf{x}_1[\ell], \ldots, \mathbf{x}_n[\ell]|\mathbf{x}_0[\ell]) = q^\ell_{\tau_\ell}(\mathbf{y}[\ell]|\mathbf{x}_{\tau_\ell-1}[\ell])\prod_{s\neq\tau_\ell} q^\ell_s(\mathrm{d}\mathbf{x}_s[\ell]|\mathbf{x}_{s-1}[\ell])\,, \tag{D.18}
$$

and for $\ell \in [\mathsf{d_y}+1:\mathsf{d}_x]$,

$$
\Gamma^{\mathbf{y}}_{1:n|0,\ell}(\mathbf{x}_1[\ell], \ldots, \mathbf{x}_n[\ell]|\mathbf{x}_0[\ell]) = \prod_{s=0}^{n-1} q^\ell_{s+1}(\mathbf{x}_{s+1}[\ell]|\mathbf{x}_s[\ell])\,. \tag{D.19}
$$

With these quantities in hand we can now prove Proposition 34.

Proof of Proposition 34. Note that for $\ell \in [1 : \mathsf{d}_{\mathbf{y}}]$,

$$\mathcal{N}(\mathbf{y}[\ell]; \alpha_{\tau_\ell}\mathbf{x}_0[\ell], 1-\alpha_{\tau_\ell}) = q^\ell_{\tau_\ell|0}(\mathbf{y}[\ell]|\mathbf{x}_0[\ell]) = \int q^\ell_{\tau_\ell}(\mathbf{y}[\ell]|\mathbf{x}_{\tau_\ell-1}[\ell]) \prod_{s\neq\tau_\ell} q^\ell_s(\mathrm{d}\mathbf{x}_s[\ell]|\mathbf{x}_{s-1}[\ell])$$
$$= \int \Gamma^{\mathbf{y}}_{1:n|0,\ell}(\mathrm{d}(\mathbf{x}_1[\ell], \ldots, \mathbf{x}_n[\ell])|\mathbf{x}_0[\ell])$$

and thus

$$\mathfrak{p}_0(\mathbf{x}_0) g^y_0(\mathbf{x}_0) \propto \mathfrak{p}_0(\mathbf{x}_0) \prod_{\ell=1}^{\mathsf{d}_{\mathbf{y}}} \mathcal{N}(\mathbf{y}[\ell]; \alpha_{\tau_\ell}\mathbf{x}_0[\ell], 1-\alpha_{\tau_\ell})$$
$$= \mathfrak{p}_0(\mathbf{x}_0) \prod_{\ell=1}^{\mathsf{d}_{\mathbf{y}}} \int \Gamma^{\mathbf{y}}_{1:n|0,\ell}(\mathrm{d}(\mathbf{x}_1[\ell], \ldots, \mathbf{x}_n[\ell])|\mathbf{x}_0[\ell])$$
$$= \mathfrak{p}_0(\mathbf{x}_0) \prod_{\ell=1}^{\mathsf{d}_x} \int \Gamma^{\mathbf{y}}_{1:n|0,\ell}(\mathrm{d}(\mathbf{x}_1[\ell], \ldots, \mathbf{x}_n[\ell])|\mathbf{x}_0[\ell]) \,.$$

By (D.16) it follows that

$$\phi^{\mathbf{y}}_0(\mathbf{x}_0) = \frac{1}{\int \Gamma^{\mathbf{y}}_{0:n}(\tilde{\mathbf{x}}_{0:n})\mathrm{d}\tilde{\mathbf{x}}_{0:n}} \int \Gamma^{\mathbf{y}}_{0:n}(\mathbf{x}_{0:n})\mathrm{d}\mathbf{x}_{1:n} \,,$$

and hence by (D.16) and (D.15) we get

$$\phi^{\mathbf{y}}_0(\mathbf{x}_0) \propto \int \mathfrak{p}_{\tau_{\mathsf{d}_{\mathbf{y}}}}(\mathbf{x}_{\tau_{\mathsf{d}_{\mathbf{y}}}}) \delta_{\mathbf{y}[\mathsf{d}_{\mathbf{y}}]}(\mathrm{d}\mathbf{x}_{\tau_{\mathsf{d}_{\mathbf{y}}}}[\mathsf{d}_{\mathbf{y}}]) \mathrm{d}\mathbf{x}^{\setminus \mathsf{d}_{\mathbf{y}}}_{\tau_{\mathsf{d}_{\mathbf{y}}}} \left\{ \prod_{i=1}^{\mathsf{d}_{\mathbf{y}}-1} \lambda_{\tau_i|\tau_{i+1}}(\mathbf{x}_{\tau_i}|\mathbf{x}_{\tau_{i+1}}) \delta_{\mathbf{y}[i]}(\mathrm{d}\mathbf{x}_{\tau_i}[i]) \mathrm{d}\mathbf{x}^{\setminus i}_{\tau_i} \right\} \lambda_{0|\tau_1}(\mathbf{x}_0|\mathbf{x}_{\tau_1}) \,.$$

This completes the proof. □

Let $\gamma^{\mathbf{y}}_{0,s}$ denote the joint time 0 and s marginal of the measure (D.15), i.e.

$$\gamma^{\mathbf{y}}_{0,s}(\mathbf{x}_0, \mathbf{x}_s) = \int \Gamma^{\mathbf{y}}_{0:n}(\mathbf{x}_{0:n})\mathrm{d}\mathbf{x}_{1:s-1}\mathrm{d}\mathbf{x}_{s+1:n} \tag{D.20}$$

We now prove the following result.

Lemma 69. *Assume (A2) and let $\tau_0 := 0$, $\tau_{\mathsf{d}_{\mathbf{y}}+1} := n$. For all $k \in [1 : \mathsf{d}_{\mathbf{y}}]$,*

(i) If $s \in [\tau_k + 1 : \tau_{k+1}]$,

$$\gamma^{\mathbf{y}}_{0,s}(\mathbf{x}_0, \mathbf{x}_s) =$$
$$\int \gamma^{\mathbf{y}}_{0,s+1}(\mathbf{x}_0, \mathbf{x}_{s+1}) q^\sigma_{\underline{s}|s+1,0}(\underline{x}_s|\underline{x}_{s+1}, \underline{x}_0) g^{\mathbf{y}}_s(\overline{x}_s) \prod_{\ell=k+1}^{\mathsf{d}_{\mathbf{y}}} q^{\sigma,\ell}_{s|s+1,0}(\mathbf{x}_s[\ell]|\mathbf{x}_{s+1}[\ell], \mathbf{x}_0[\ell])\mathrm{d}\mathbf{x}_{s+1} \,.$$

(ii) If $s = \tau_k$,

$$\gamma^{\mathbf{y}}_{0,s}(\mathbf{x}_0, \mathbf{x}_s) = \int \gamma^{\mathbf{y}}_{0,s+1}(\mathbf{x}_0, \mathbf{x}_{s+1}) q^\sigma_{\underline{s}|s+1,0}(\underline{x}_s|\underline{x}_{s+1}, \underline{x}_0)$$
$$\times \prod_{i=1}^{k-1} g^{\mathbf{y}}_{s,i}(\overline{x}_s[i]) \prod_{\ell=k+1}^{\mathsf{d}_{\mathbf{y}}} q^{\sigma,\ell}_{s|s+1,0}(\mathbf{x}_s[\ell]|\mathbf{x}_{s+1}[\ell], \mathbf{x}_0[\ell])\mathrm{d}\mathbf{x}_{s+1} \,.$$

Proof of Lemma 69. Let $k \in [1 : \mathsf{d}_{\mathbf{y}}]$ and assume that $s \in [\tau_k + 1 : \tau_{k+1} - 2]$. By (**A2**), (D.16), (D.18) and (D.19) we have that

$$\gamma^{\mathbf{y}}_{0,s}(\mathbf{x}_0, \mathbf{x}_s) = \mathfrak{p}_0(\mathbf{x}_0) q_{\underline{s}|0}(\underline{x}_s|\underline{x}_0) \prod_{i=1}^{k} q^i_{\tau_i|0}(\mathbf{y}[i]|\mathbf{x}_0[i]) q^i_{s|\tau_i}(\mathbf{x}_s[i]|\mathbf{y}[i]) \\ \times \prod_{\ell=k+1}^{\mathsf{d}_{\mathbf{y}}} q^\ell_{s|0}(\mathbf{x}_s[\ell]|\mathbf{x}_0[\ell]) q^\ell_{\tau_\ell|s}(\mathbf{y}[\ell]|\mathbf{x}_s[\ell]) \,,$$

and thus, using the following identity valid for $\ell \in [k+1 : \mathsf{d}_{\mathbf{y}}]$

$$\begin{aligned} &q^\ell_{s|0}(\mathbf{x}_s[\ell]|\mathbf{x}_0[\ell]) q^\ell_{\tau_\ell|s}(\mathbf{y}[\ell]|\mathbf{x}_s[\ell]) \\ &\quad = q^\ell_{s|0}(\mathbf{x}_s[\ell]|\mathbf{x}_0[\ell]) \int q^\ell_{\tau_\ell|s+1}(\mathbf{y}[\ell]|\mathbf{x}_{s+1}[\ell]) q^\ell_{s+1}(\mathbf{x}_{s+1}[\ell]|\mathbf{x}_s[\ell]) \mathrm{d}\mathbf{x}_{s+1}[\ell] \\ &\quad = \int q^{\sigma,\ell}_{s|s+1,0}(\mathbf{x}_s[\ell]|\mathbf{x}_{s+1}[\ell], \mathbf{x}_0[\ell]) q^\ell_{\tau_\ell|s+1}(\mathbf{y}[\ell]|\mathbf{x}_{s+1}[\ell]) q^\ell_{s+1|0}(\mathbf{x}_{s+1}[\ell]|\mathbf{x}_0[\ell]) \mathrm{d}\mathbf{x}_{s+1}[\ell] \,, \end{aligned}$$

and that $q_{\underline{s}|0}(\underline{x}_s|\underline{x}_0) q_{\underline{s}+1}(\underline{x}_{s+1}|\underline{x}_s) = q^\sigma_{\underline{s}|s+1,0}(\underline{x}_s|\underline{x}_{s+1}, \underline{x}_0) q_{\underline{s}+1|0}(\underline{x}_{s+1}|\underline{x}_0)$ we get that

$$\begin{aligned} &\gamma^{\mathbf{y}}_{0,s}(\mathbf{x}_0, \mathbf{x}_s) \\ &= \int \mathfrak{p}_0(\mathbf{x}_0) q_{\underline{s}|0}(\underline{x}_s|\underline{x}_0) q_{\underline{s}+1}(\mathrm{d}\underline{x}_{s+1}|\underline{x}_s) \\ &\quad \times \prod_{i=1}^{k} q^i_{\tau_i|0}(\mathbf{y}[i]|\mathbf{x}_0[i]) q^i_{s|\tau_i}(\mathbf{x}_s[i]|\mathbf{y}[i]) q^i_{s+1|\tau_i}(\mathrm{d}\mathbf{x}_{s+1}[i]|\mathbf{y}[i]) \\ &\quad \times \prod_{\ell=k+1}^{\mathsf{d}_{\mathbf{y}}} q^{\sigma,\ell}_{s|s+1,0}(\mathbf{x}_s[\ell]|\mathbf{x}_{s+1}[\ell], \mathbf{x}_0[\ell]) q^\ell_{\tau_\ell|s+1}(\mathbf{y}[\ell]|\mathbf{x}_{s+1}[\ell]) q^\ell_{s+1|0}(\mathbf{x}_{s+1}[\ell]|\mathbf{x}_0[\ell]) \mathrm{d}\mathbf{x}_{s+1}[\ell] \\ &= \int \gamma^{\mathbf{y}}_{0,s+1}(\mathbf{x}_0, \mathbf{x}_{s+1}) q^\sigma_{\underline{s}|s+1,0}(\underline{x}_s|\underline{x}_{s+1}, \underline{x}_0) g^{\mathbf{y}}_s(\overline{x}_s) \prod_{\ell=k+1}^{\mathsf{d}_{\mathbf{y}}} q^{\sigma,\ell}_{s|s+1,0}(\mathbf{x}_s[\ell]|\mathbf{x}_{s+1}[\ell], \mathbf{x}_0[\ell]) \mathrm{d}\mathbf{x}_{s+1} \,. \end{aligned}$$

If $s = \tau_{k+1}$ then

$$\gamma^{\mathbf{y}}_{0,s}(\mathbf{x}_0, \mathbf{x}_s) = \mathfrak{p}_0(\mathbf{x}_0) q_{\underline{s}|0}(\underline{x}_s|\underline{x}_0) \prod_{i=1}^{k} q^i_{\tau_i|0}(\mathbf{y}[i]|\mathbf{x}_0[i]) q^i_{s|\tau_i}(\mathbf{x}_s[i]|\mathbf{y}[i]) \\ \times q^{k+1}_{\tau_{k+1}|0}(\mathbf{y}[k+1]|\mathbf{x}_0[k+1]) \prod_{\ell=k+2}^{\mathsf{d}_{\mathbf{y}}} q^\ell_{s|0}(\mathbf{x}_s[\ell]|\mathbf{x}_0[\ell]) q^\ell_{\tau_\ell|s}(\mathbf{y}[\ell]|\mathbf{x}_s[\ell]) \,, \tag{D.21}$$

and similarly to the previous case we get

$$\begin{aligned} &\gamma_{0,s}(\mathbf{x}_0, \mathbf{x}_s) \\ &\quad = \int \gamma^{\mathbf{y}}_{0,s+1}(\mathbf{x}_0, \mathbf{x}_{s+1}) q^\sigma_{\underline{s}|s+1,0}(\underline{x}_s|\underline{x}_{s+1}, \underline{x}_0) g^{\mathbf{y}}_s(\overline{x}_s) \prod_{\ell=k+2}^{\mathsf{d}_{\mathbf{y}}} q^{\sigma,\ell}_{s|s+1,0}(\mathbf{x}_s[\ell]|\mathbf{x}_{s+1}[\ell], \mathbf{x}_0[\ell]) \mathrm{d}\mathbf{x}_{s+1} \,. \end{aligned}$$

Finally, if $s = \tau_{k+1} - 1$, then

$$\gamma^{\mathbf{y}}_{0,s}(\mathbf{x}_0, \mathbf{x}_s) = \mathfrak{p}_0(\mathbf{x}_0) q_{\underline{s}|0}(\underline{x}_s|\underline{x}_0) \prod_{i=1}^{k} q^i_{\tau_i|0}(\mathbf{y}[i]|\mathbf{x}_0[i]) q^i_{s|\tau_i}(\mathbf{x}_s[i]|\mathbf{y}[i]) \\ \times q^{k+1}_{s|0}(\mathbf{x}_s[k+1]|\mathbf{x}_0[k+1]) q^{k+1}_{\tau_{k+1}|s}(\mathbf{y}[k+1]|\mathbf{x}_s[k+1]) \prod_{\ell=k+2}^{\mathsf{d}_{\mathbf{y}}} q^\ell_{s|0}(\mathbf{x}_s[\ell]|\mathbf{x}_0[\ell]) q^\ell_{\tau_\ell|s}(\mathbf{y}[\ell]|\mathbf{x}_s[\ell]) \,,$$

and using

$$q_{s|0}^{k+1}(\mathbf{x}_s[k+1]|\mathbf{x}_0[k+1])q_{\tau_{k+1}|s}^{k+1}(\mathbf{y}[k+1]|\mathbf{x}_s[k+1]) \\ = q_{s|\tau_{k+1},0}^{\sigma,k+1}(\mathbf{x}_s[k+1]|\mathbf{x}_{\tau_{k+1}}[k+1],\mathbf{x}_0[k+1])q_{\tau_{k+1}|0}^{k+1}(\mathbf{y}[k+1]|\mathbf{x}_0[k+1])$$

we find that

$$\gamma_{0,s}(\mathbf{x}_0,\mathbf{x}_s) \\ = \int \gamma_{0,\tau_{k+1}}^{\mathbf{y}}(\mathbf{x}_0,\mathbf{x}_{\tau_{k+1}})\underline{q}_{s|\tau_{k+1},0}^{\sigma}(\underline{x}_s|\underline{x}_{\tau_{k+1}},\underline{x}_0)g_s^{\mathbf{y}}(\overline{x}_s)\prod_{\ell=k+1}^{\mathsf{d_y}} q_{s|s+1,0}^{\sigma,\ell}(\mathbf{x}_s[\ell]|\mathbf{x}_{\tau_{k+1}}[\ell],\mathbf{x}_0[\ell])\mathrm{d}\mathbf{x}_{\tau_{k+1}}\,.$$

□

D.3 Algorithmic details and numerics

D.3.0.1 GMM

For a given dimension d_x, we consider $\mathsf{q}_{\mathrm{data}}$ a mixture of 25 Gaussian random variables. The Gaussian random variables have mean $\boldsymbol{\mu}_{i,j} := (8i, 8j, \cdots, 8i, 8j) \in \mathbb{R}^{\mathsf{d}_x}$ for $(i,j) \in \{-2,-1,0,1,2\}^2$ and unit variance. The mixture (unnormalized) weights $\omega_{i,j}$ are independently drawn according to a χ^2 distribution. The κ paramater of `MCGdiff` is $\kappa^2 = 10^{-4}$. We use 20 steps of `DDIM` for the numerical examples and for all algorithms.

Score: Note that $\mathsf{q}_s(x_s) = \int q_{s|0}(x_s|x_0)\mathsf{q}_{\mathrm{data}}(x_0)\mathrm{d}x_0$. As $\mathsf{q}_{\mathrm{data}}$ is a mixture of Gaussians, $\mathsf{q}_s(x_s)$ is also a mixture of Gaussians with means $\alpha_s^{1/2}\boldsymbol{\mu_{i,j}}$ and unitary variances. Therefore, using automatic differentiation libraries, we can calculate $\nabla \log \mathsf{q}_s(x_s)$. Setting $\mathbf{e}(x_s,s) = -(1-\alpha_s)^{1/2}\nabla \log \mathsf{q}_s(x_s)$ leads to the optimum of (5.4).

Forward process scaling: We chose the sequence of $\{\beta_s\}_{s=1}^{1000}$ as a linearly decreasing sequence between $\beta_1 = 0.2$ and $\beta_{1000} = 10^{-4}$.

Measurement model: For a pair of dimensions $(\mathsf{d}_x, \mathsf{d_y})$ the measurement model $(y, \mathrm{A}, \sigma_y)$ is drawn as follows:

- A: We first draw $\tilde{\mathrm{A}} \sim \mathcal{N}(0_{\mathsf{d_y}\times\mathsf{d}_x}, \mathrm{I}_{\mathsf{d_y}\times\mathsf{d}_x})$ and compute the SVD decomposition of $\tilde{A} = \mathrm{USV}^T$. Then, we sample for $(i,j) \in \{-2,-1,0,1,2\}^2$, $s_{i,j}$ according to a uniform in $[0,1]$. Finally, we set $\mathrm{A} = \mathrm{U}\,\mathrm{Diag}(\{s_{i,j}\}_{(i,j)\in\{-2,-1,0,1,2\}^2})\mathrm{V}^T$.
- σ_y: We draw σ_y uniformly in the interval $[0, \max(s_1, \cdots, s_{\mathsf{d_y}})]$.
- y: We then draw $x_* \sim \mathsf{q}_{\mathrm{data}}$ and set $y := \mathrm{A}x_* + \sigma_y\epsilon$ where $\epsilon \sim \mathcal{N}(0_{\mathsf{d_y}}, \mathrm{I}_{\mathsf{d_y}})$.

Posterior: Once we have drawn both $\mathsf{q}_{\mathrm{data}}$ and $(y, \mathrm{A}, \sigma_y)$, the posterior can be exactly calculated using Bayes formula and gives a mixture of Gaussians with mixture components $c_{i,j}$ and associated weights $\tilde{\omega}_{i,j}$

$$c_{i,j} := \mathcal{N}(\Sigma\left(\mathrm{A}^T y/\sigma_y^2 + \boldsymbol{\mu}_{i,j}\right), \Sigma)\,, \\ \tilde{\omega}_i := \omega_i \mathcal{N}(y; \mathrm{A}\boldsymbol{\mu}_{i,j}, \sigma^2\,\mathrm{I}_{\mathsf{d}_x} + \mathrm{A}\mathrm{A}^T)\,,$$

where $\Sigma := \left(\mathrm{I}_{\mathsf{d}_x} + \sigma_y^{-2}\mathrm{A}^T\mathrm{A}\right)^{-1}$.

Variational Inference: The RNVP entries in the numerical examination are obtained by Variational Inference using the RNVP architecture for the normalizing flow from Dinh et al. (2017). Given a normalizing flow f_ϕ with $\phi \in \mathbb{R}^j, j \in \mathbb{N}_*$, the training procedure consists of optimizing the ELBO, i.e., solving the optimization problem

$$\phi_* = \underset{\phi \in \mathbb{R}^j}{\operatorname{argmax}} \sum_{k=1}^{N_{nf}} \log |\mathrm{Jf}_\phi(\epsilon_i)| + \log \pi_*(\mathsf{f}_\phi(\epsilon_i)) \,, \tag{D.22}$$

where $N_{nf} \in \mathbb{N}_*$ is the minibatch-size, Jf_ϕ the Jacobian of f_ϕ w.r.t ϕ, and $\epsilon_{1:N_{nf}} \sim \mathcal{N}(0, \mathbf{I})^{\otimes N_{nf}}$. All the experiments were performed using a 10 layers RNVP. Equation (D.22) is solved using Adam algorithm Kingma and Ba (2015a) with a learning rate of 10^{-3} and 200 iterations with $N_{nf} = 10$. The losses for each pair $(\mathsf{d}_x, \mathsf{d}_\mathbf{y})$ is shown in figure D.1, where one can see that the majority of the losses have converged.

Choosing DDIM timesteps for a given measurement model: Given a number of DDIM samples R, we choose the timesteps $1 = t_1 < \cdots < t_R = 1000 \in [1 : 1000]$ as to try to satisfy the two following constraints:

- For all $i \in [1 : \mathsf{d}_\mathbf{y}]$ there exists a t_j such that $\sigma_y \alpha_{t_j}^{1/2} \approx (1 - \alpha_{t_j})^{1/2} s_i$,
- For all $i \in [1 : R-1]$, $\alpha_{t_i}^{1/2} - \alpha_{t_{i+1}}^{1/2} \approx \delta$ for some $\delta > 0$.

The first constraint comes naturally from the definition of τ_i. Since the potentials have mean $\alpha_{t_i}^{1/2} y$, the second condition constrains the intermediate laws remain "close". An algorithm that approximately satisfies both constraints is given below.

Algorithm 2: Timesteps choice

Input: Number of DDIM steps R, σ_y, $\{s_i\}_{i=1}^{\mathsf{d}_\mathbf{y}}$, $\{\alpha_i\}_{i=1}^{1000}$
Output: $\{t_j\}_{j=1}^{R}$

Set $S_\tau = \{\}$.
for $j \leftarrow [1 : \mathsf{d}_\mathbf{y}]$ **do**
  Set $\tilde{\tau}_j = \operatorname{argmin}_{\ell \in [1:1000]} |\sigma_y \alpha_\ell^{1/2} - (1 - \alpha_\ell)^{1/2}) s_j|$.
  Add $\tilde{\tau}_j$ to S_τ if $\tilde{\tau}_j \notin S_\tau$.
Set $n_m = R - \#S_\tau - 1$ and $\delta = (\alpha_1^{1/2} - \alpha_{1000}^{1/2})/n_m$.
Set $t_1 = 1$, $e = 1$ and $i_e = 1$. **for** $\ell \leftarrow [2 : 1000]$ **do**
  if $\alpha_e^{1/2} - \alpha_\ell^{1/2} > \delta$ *or* $\ell \in S_\tau$ **then**
    Set $e = \ell$, $i_e = i_e + 1$ and $\tau_{i_e} = \ell$.
Set $\tau_R = 1000$.

Additional numerics: We now proceed to illustrate in Figures D.2 to D.4 the first 2 components for one of the measurement models for all the different combinations of $(\mathsf{d}_x, \mathsf{d}_\mathbf{y})$ combinations used in table 5.1.

We also show in figure D.5 the evolution of each observed coordinate in the noise case with $\mathsf{d}_\mathbf{y} = 4$. We can see that it follows closely the forward path of the diffused observations indicated by the blue line.

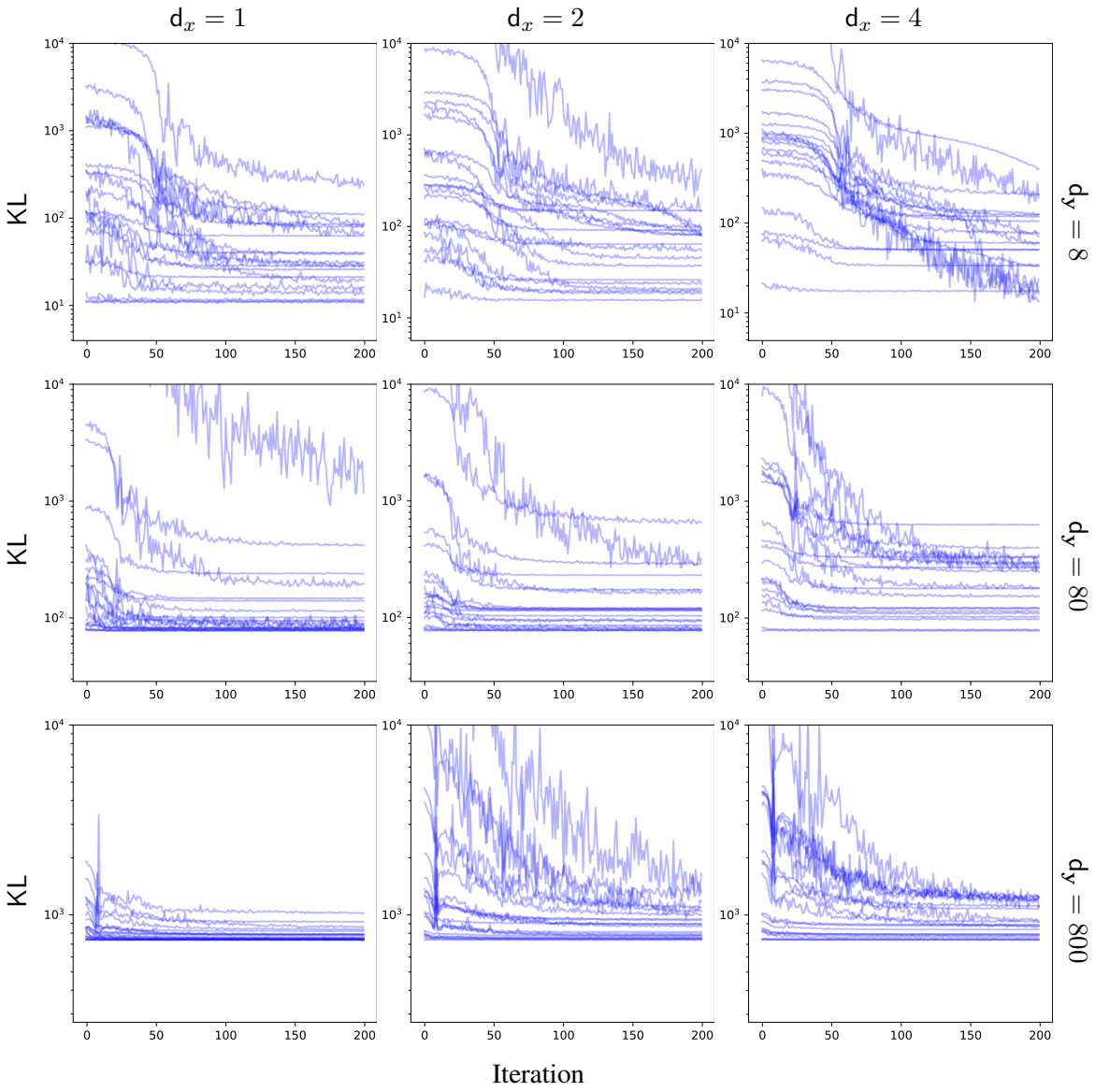


Figure D.1: Evolution of KL *with the number of iterations for all pairs of* $(\mathsf{d}_x, \mathsf{d}_\mathbf{y})$ *tested in the GMM case.*

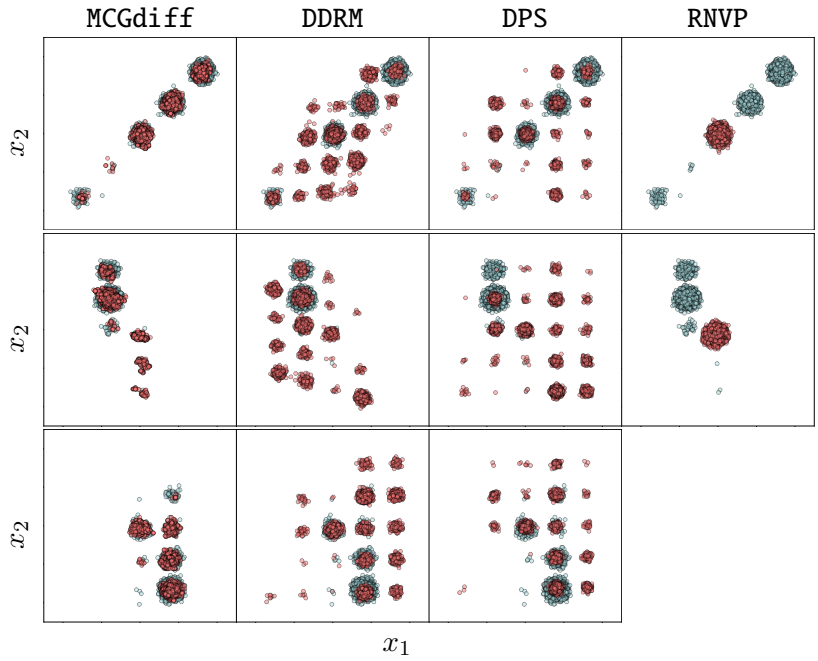


Figure D.2: First two dimensions for the GMM case with $\mathsf{d}_x = 8$. *The rows represent* $\mathsf{d}_{\mathbf{y}} = 1, 2, 4$ *respectively. The blue dots represent samples from the exact posterior, while the red dots correspond to samples generated by each of the algorithms used (the names of the algorithms are given at the top of each column).*

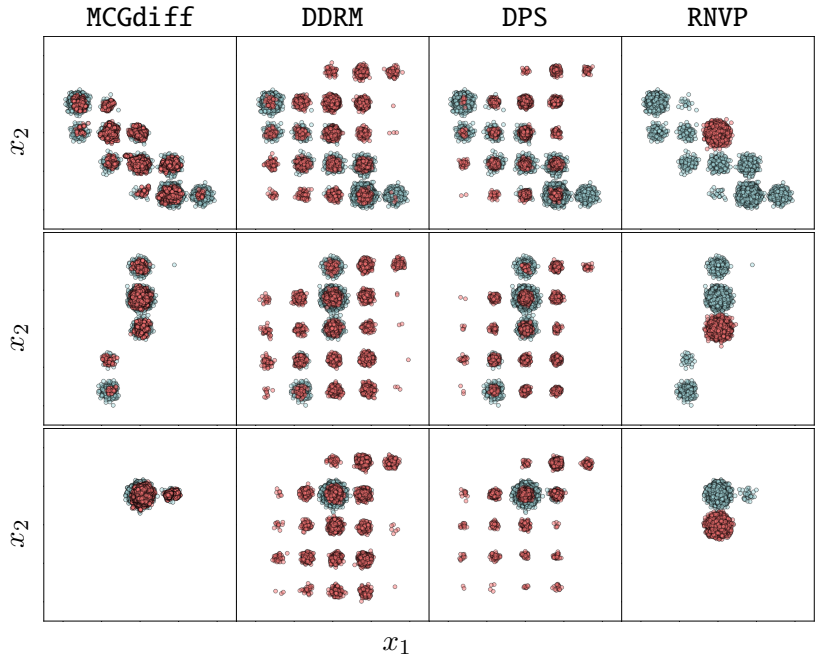


Figure D.3: First two dimensions for the GMM case with $\mathsf{d}_x = 80$. *The rows represent* $\mathsf{d}_{\mathbf{y}} = 1, 2, 4$ *respectively. The blue dots represent samples from the exact posterior, while the red dots correspond to samples generated by each of the algorithms used (the names of the algorithms are given at the top of each column).*

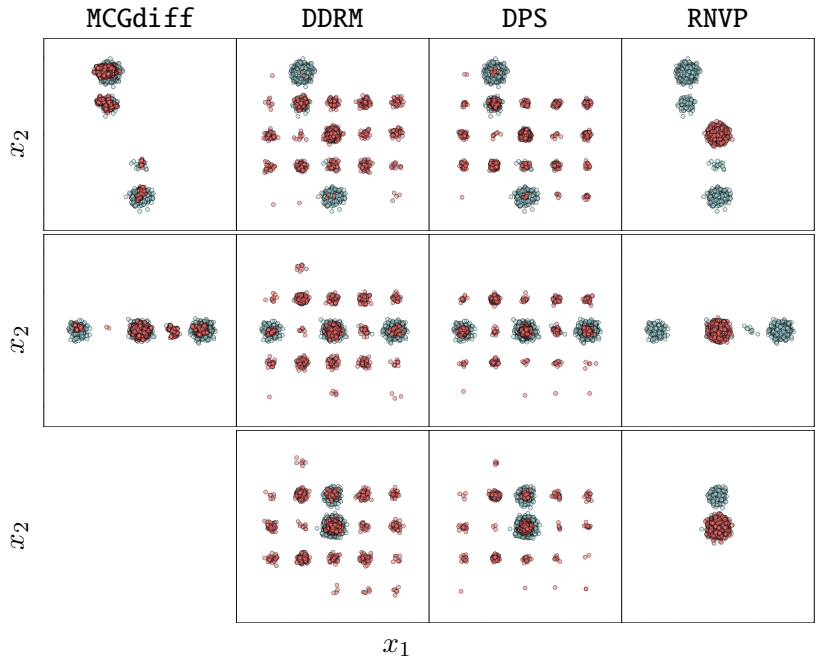


Figure D.4: First two dimensions for the GMM case with $\mathsf{d}_x = 800$. *The rows represent* $\mathsf{d}_{\mathbf{y}} = 1, 2, 4$ *respectively. The blue dots represent samples from the exact posterior, while the red dots correspond to samples generated by each of the algorithms used (the names of the algorithms are given at the top of each column).*

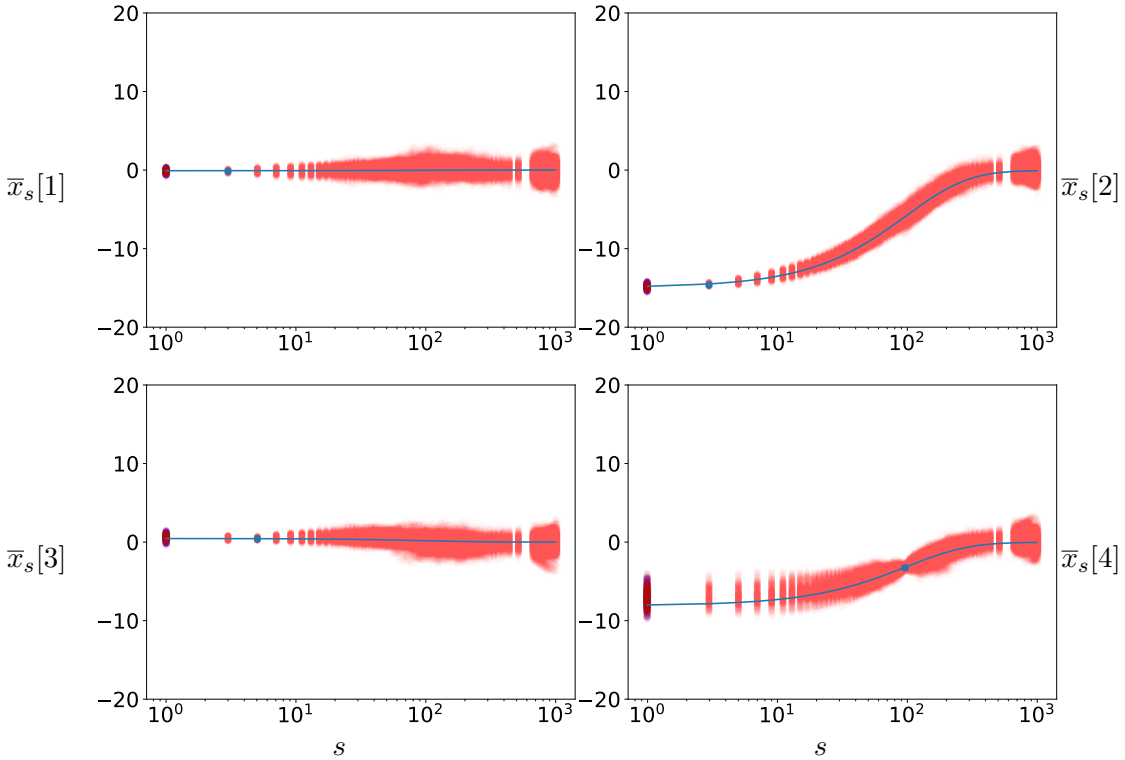


Figure D.5: Illustration of the particle cloud of the 4 *first observed coordinate in the case* $(\mathsf{d}_{\mathbf{y}}, \mathsf{d}_x) = (4, 800)$ *with* 100 `DDIM` *steps. The red points represent the particle cloud, while the purple points at the origin represent the posterior distribution. The blue curve corresponds to the curve* $s \to \alpha_s^{1/2}\mathbf{y}[\ell]$ *and the blue dot on the curve to* $\alpha_{\tau_\ell}^{1/2}\mathbf{y}[\ell]$.

d	d_y	MCGdiff	DDRM	DPS	RNVP
8	1	**1.43 ± 0.55**	5.88 ± 1.16	4.86 ± 1.01	9.43 ± 0.99
8	2	**0.49 ± 0.24**	5.20 ± 1.32	5.79 ± 1.96	8.93 ± 1.29
8	4	**0.38 ± 0.25**	2.51 ± 1.29	3.48 ± 1.52	6.71 ± 1.54
80	1	**1.39 ± 0.45**	5.64 ± 1.10	4.98 ± 1.14	6.86 ± 0.88
80	2	**0.67 ± 0.24**	7.07 ± 1.35	5.10 ± 1.23	7.79 ± 1.50
80	4	**0.28 ± 0.14**	7.81 ± 1.48	4.28 ± 1.26	7.95 ± 1.61
800	1	**2.40 ± 1.00**	7.44 ± 1.15	6.49 ± 1.16	7.74 ± 1.34
800	2	**1.31 ± 0.60**	8.95 ± 1.12	6.88 ± 1.01	8.75 ± 1.02
800	4	**0.47 ± 0.19**	8.39 ± 1.48	5.51 ± 1.18	7.81 ± 1.63

Table D.1: Extended GMM sliced wasserstein table.

d	SW
2	0.79 ± 0.15
6	0.87 ± 0.07
10	0.96 ± 0.06

Table D.2: Sliced Wasserstein between learned diffusion and target prior.

Table D.1 is an extended version of table 5.1.

D.3.0.2 FMM

A funnel distribution is defined by the following density

$$\mathcal{N}(x_1; 0, 1) \prod_{i=1}^{d} \mathcal{N}(x_i; 0, \exp(x_1/2)) \,.$$

To generate a Funnel mixture model of 20 components in dimension d, we start by firstly sampling $(\mu_i, R_i)_{i=1}^{20}$ uniformly in $([-20, 20]^d \times \mathsf{SO}(R^d))^{\times 20}$. The mixture will consist of 20 Funnel random variables translated by μ_i and rotated by R_i, with unnormalized weights $\omega_{i,j}$ that are independently drawn uniformly in $[0, 1]$.

Score The denoising diffusion network $\mathbf{e}(\theta)$ in dimension d is defined as a 5 layers Resnet network where each Resnet block consists of the chaining of three blocks where each block has the following layers:

- Linear $(512, 1024)$,
- 1d Batch Norm,
- ReLU activation.

The Resnet is preceeded by an input embedding from dimension d to 512 and in the end an output embedding layer projects the output of the resnet from 512 to d. The time t is embedded using positional embedding into dimension 512 and is added to the input at each Resnet block. The network is trained using the same loss as in Ho et al. (2020) for 10^4 iterations using a batch size of 512 samples. A learning rate of 10^{-3} is used for the Adam optimizer Kingma and Ba (2015b). Figure D.6 illustrate the outcome of the learned diffusion generative model and the target prior. In table D.2 we show the CLT 95% intervals for the SW between the learned diffusion generative model and the target prior.

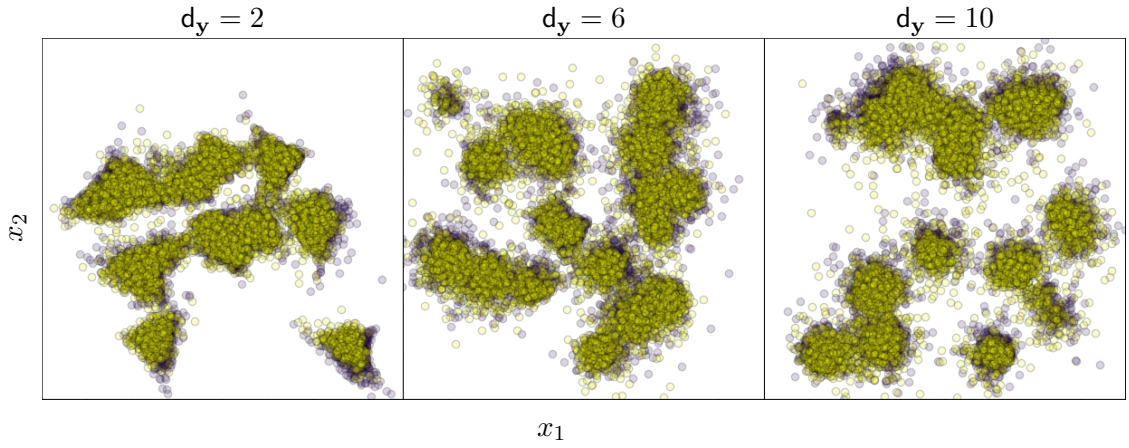


Figure D.6: Purple points are samples from the prior and yellow samples from the diffusion with 25 `DDIM` *steps.*

Forward process scaling We chose the sequence of $\{\beta_s\}_{s=1}^{1000}$ as a linearly decreasing sequence between $\beta_1 = 0.2$ and $\beta_{1000} = 10^{-4}$.

Measurement model The measurement model was generated in the same way as for the GMM case.

Posterior The posterior samples were generated by running the No U-turn sampler (Hoffman and Gelman (2011)) with a chain of length 10^4 and taking the last sample of the chain. This was done in parallel to generate 10^4 samples. The mass matrix and learning rate were set by first running Stan's warmup and taking the last values of the warmup phase.

Variational inference: Variational inference in FMM shares the same details as the GMM case. The analogous of figure D.1 is displayed at figure D.7.

Additional plots: We now proceed to illustrate in Figures D.8 to D.10 the first 2 components for one of the measurement models for all the different combinations of $(\mathsf{d}_x, \mathsf{d_y})$ combinations used in table 5.1.

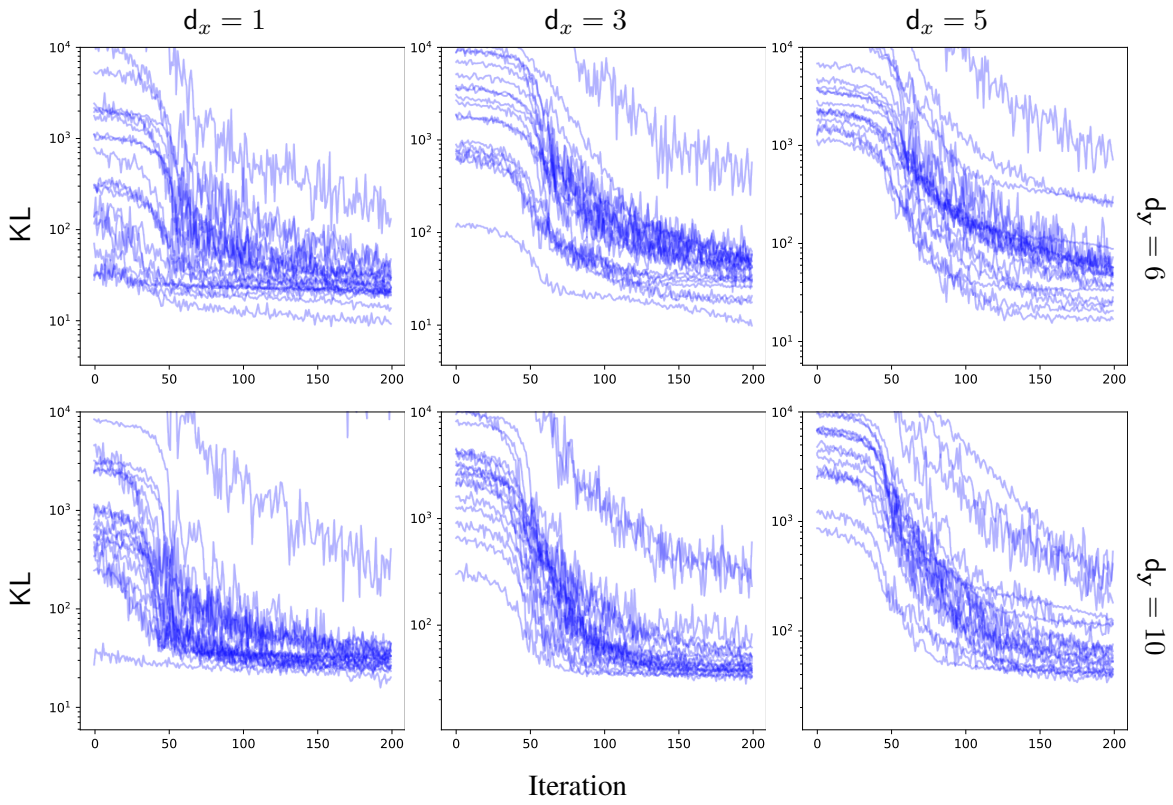


Figure D.7: Evolution of KL *with the number of iterations for all pairs of* $(\mathsf{d}_x, \mathsf{d}_\mathbf{y})$ *tested in the FMM case.*

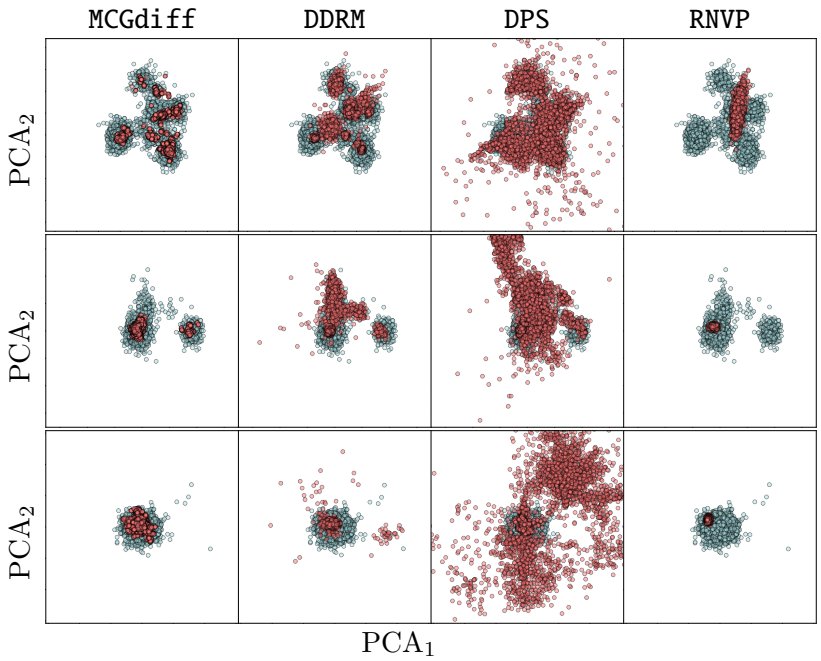


Figure D.8: First two dimensions for the FMM case with $\mathsf{d}_x = 10$*. The rows represent* $\mathsf{d}_\mathbf{y} = 1, 3, 5$ *respectively. The blue dots represent samples from the exact posterior, while the red dots correspond to samples generated by each of the algorithms used (the names of the algorithms are given at the top of each column).*

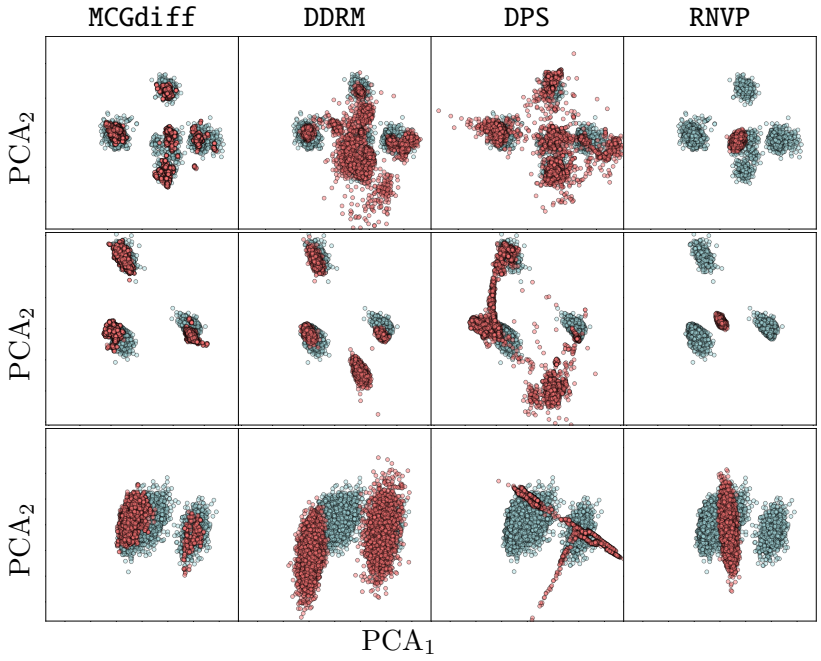


Figure D.9: First two dimensions for the FMM case with $\mathsf{d}_x = 6$. *The rows represent* $\mathsf{d}_\mathbf{y} = 1, 3, 5$ *respectively. The blue dots represent samples from the exact posterior, while the red dots correspond to samples generated by each of the algorithms used (the names of the algorithms are given at the top of each column).*

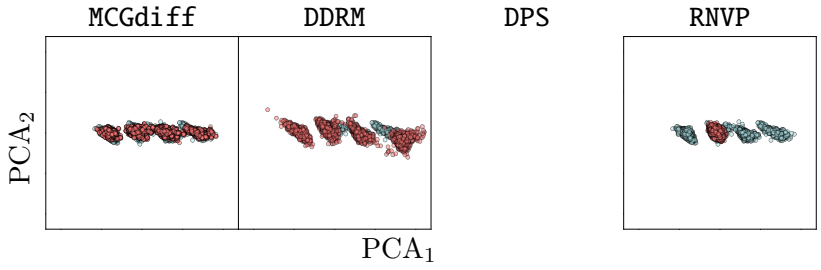


Figure D.10: First two dimensions for the FMM case with $\mathsf{d}_x = 2$ *and* $\mathsf{d}_\mathbf{y} = 1$. *The blue dots represent samples from the exact posterior, while the red dots correspond to samples generated by each of the algorithms used (the names of the algorithms are given at the top of each column).*

D.3.0.3 Image datasets

We now present samples from `MCGdiff` in different image dataset and different kinds of inverse problems.

Super Resolution We start by super resolution. We set $\sigma_y = 0.05$ for all the datasets and $\zeta_{\text{coeff}} = 0.1$ for `DPS` . We use 100 steps of `DDIM` with $\eta = 1$. The results are shown in Figure D.11. We use a downsampling ratio of 4 for the CIFAR-10 dataset, 8 for both Flowers and Cats datasets and 16 for the others. The dimension of the datasets are recalled in table D.3. We display in figure D.11 samples from `MCGdiff`, `DPS`and `DDRM`over several different image datasets (table D.3). For each algorithm, we generate 1000 samples and we show the pair of samples that are the furthest apart in L^2 norm from each other in the pool of samples. For `MCGdiff` we ran several parallel particle filters with $N = 64$ to generate 1000 samples.

	CIFAR-10	Flowers	Cats	Bedroom	Church	CelebaHQ
(W, H, C)	$(32, 32, 3)$	$(64, 64, 3)$	$(128, 128, 3)$	$(256, 256, 3)$	$(256, 256, 3)$	$(256, 256, 3)$

Table D.3: The datasets used for the inverse problems over image datasets.

Gaussian 2D debluring We consider a Gaussian 2D square kernel with sizes $(w/6, h/6)$ and standard deviation $w/30$ where (w, h) are the width and height of the image. We set $\sigma_y = 0.1$ for all the datasets and $\zeta_{\text{coeff}} = 0.1$ for `DPS` . We use 100 steps of `DDIM` with $\eta = 1$. We display in figure D.12 samples from `MCGdiff`, `DPS`and `DDRM`over several different image datasets (table D.3). For each algorithm, we generate 1000 samples and we show the pair of samples that are the furthest appart in L^2 norm from each other in the pool of samples. For `MCGdiff` we ran several parallel particle filters with $N = 64$ to generate 1000 samples.

Inpainting on CelebA We consider the inpainting problem on the CelebA dataset with several different masks in figure D.13. We show in figure D.14 the evolution of the particle cloud with s.

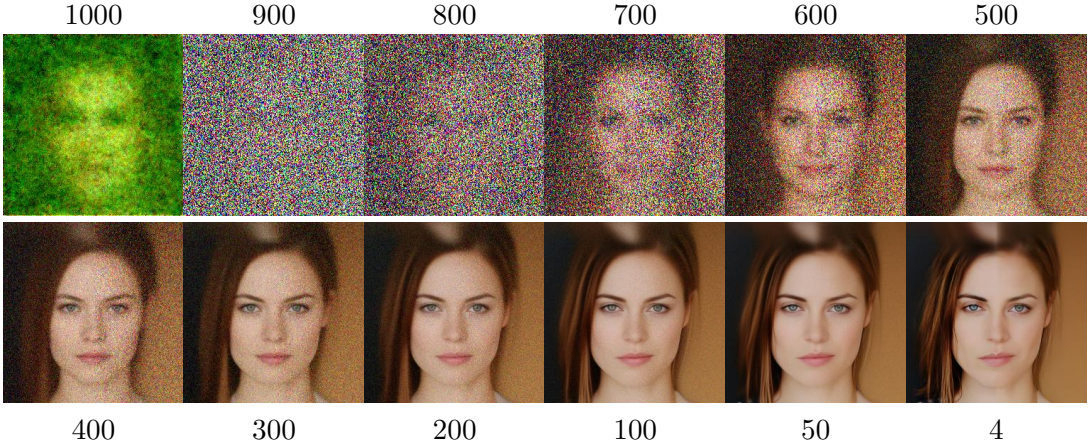


Figure D.14: Evolution of the particle cloud for one of the masks. The numbers on top and bottom indicate the step s of the approximation.

Figure D.11: Ratio 4 for CIFAR, 8 for flowers and Cats and 16 for CELEB

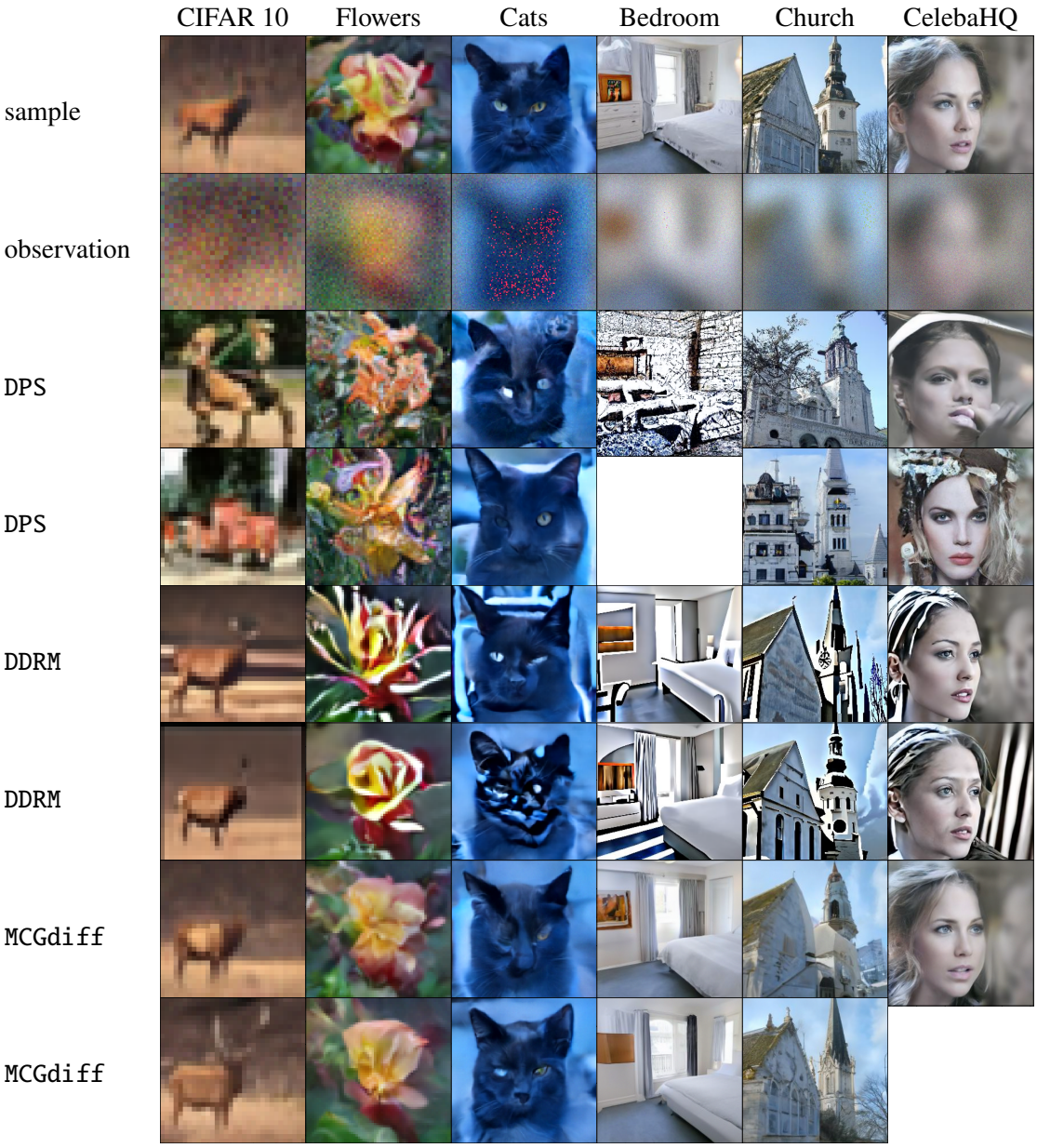

CIFAR 10
Flowers
Cats
Bedroom
Church
CelebaHQ
sample
observation
DPS
DPS
DDRM
DDRM
MCGdiff
MCGdiff

Figure D.12

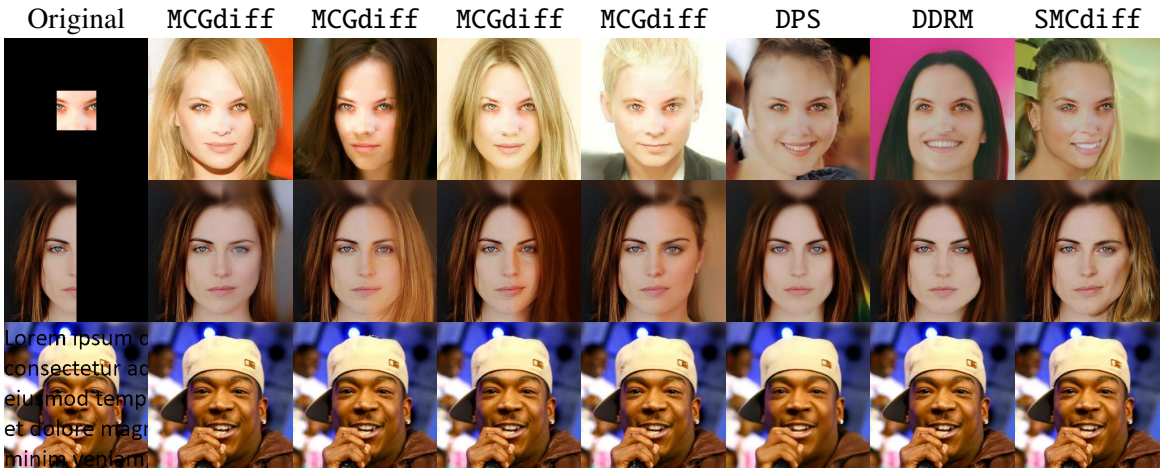


Figure D.13: Inpainting with different masks on the CelebA test set.

Appendix E

Appendix of Chapter 6

E.1 Additional Theoretical Results on DDM

In this section we prove two important aspects mentioned in section 6.3. Namely, that the inference process matches the marginals of the forward process ($q^{\eta}_{k|0}(x_k|x_0) = \mathsf{q}_{k|0}(x_k|x_0)$) and that for a certain choice of weighting coefficients, (6.3) consists of minimizing a certain KL, where for two densities f, g we define

$$\mathsf{KL}(f \parallel g) = \int \log\left(\frac{f(x)}{g(x)}\right) f(x)\mathrm{d}x\,. \tag{E.1}$$

This follows closely Song et al. (2021a), adapting it to our notation and to the variance exploding framework.

Lemma 70. *Let $\{\eta_k\}_{k\in\mathbb{N}}$ satisfy $\eta_k^2 \in [0, \upsilon_k^2]$ for all $k \in [1:K]$. Then*

$$q^{\eta}_{k|0}(x_k|x_0) = \mathsf{q}_{k|0}(x_k|x_0)\,.$$

Proof. We proceed by induction. By definition, equality holds for $k = K$. Assume that for $k+1$ the equality holds. Then, note that

$$\begin{aligned} q^{\eta}_{k|0}(x_k|x_0) &= \int q^{\eta}_{k|k+1,0}(x_k|x_{k+1}, x_0) q^{\eta}_{k+1|0}(x_{k+1}|x_0)\mathrm{d}x_{k+1} \\ &= \int \mathcal{N}\left(x_k; \boldsymbol{\mu}_k(x_0, x_{k+1}), \eta_k^2\,\mathrm{I}_d\right) \mathcal{N}\left(x_{k+1}; x_0, \upsilon_{k+1}^2\,\mathrm{I}\right) \mathrm{d}x_{k+1}\,, \end{aligned}$$

with $\boldsymbol{\mu}_k(x_0, x_{k+1}) = x_0 + (\upsilon_k^2/\upsilon_{k+1}^2 - \eta_k^2/\upsilon_{k+1}^2)^{1/2}(x_{k+1} - x_0)$. By standard Gaussian conjugation formulas, we have that $q^{\eta}_{k|0}(x_k|x_0) = \mathcal{N}\left(x_k; x_0, \upsilon_k^2\right)$, completing the proof. □

Note that taking $\eta_k^2 = \frac{\upsilon_k^2}{\upsilon_{k+1}^2}\rho_{k+1}^2$ yields $q^{\eta}_{k|k+1,0} = \mathsf{q}_{k|k+1,0}$ where

$$\mathsf{q}_{k|k+1,0}(x_k|x_0, x_{k+1}) := \frac{q_{k+1|k}(x_{k+1}|x_k) q_{k|0}(x_k|x_0)}{q_{k+1|0}(x_{k+1}|x_0)} = \mathcal{N}(x_k; x_0 + \frac{\upsilon_k^2}{\upsilon_{k+1}^2}(x_{k+1} - x_0), \frac{\upsilon_k^2}{\upsilon_{k+1}^2}\rho_{k+1}^2\,\mathrm{I})\,. \tag{E.2}$$

This shows that the inference process can be seen as a generalization of the forward noising process.

Lemma 71. *Let $\mu(x_{0:K}) = \mathsf{q}_{\mathrm{data}}(x_0) q^{\eta}_{1:K|0}(x_{1:K}|x_0)$. Then,*

$$\mathsf{KL}(\mu \parallel \mathsf{p}_{\theta}^{0:K}) = C + \sum_{k=1}^{K} \gamma_k^2 \mathbb{E}_{X_0\sim\mathsf{q}_{\mathrm{data}}, \epsilon\sim\mathcal{N}(0,\mathrm{I})}\left[\|\mathcal{D}^{\theta}_{0|k}(X_0 + \upsilon_k\epsilon, \upsilon_k) - X_0\|^2\right]\,, \tag{E.3}$$

where C is a constant independent of θ and

$$\gamma_k^2 = \eta_{k-1}^{-2}\left[1 - (\upsilon_{k-1}^2/\upsilon_k^2 - \eta_{k-1}^2/\upsilon_k^2)^{1/2}\right]^2 \quad \textit{for} \quad k > 1\,,$$
$$\gamma_1^2 = \eta_0^{-2}\,.$$

Proof. In this proof, we treat every constant not depending on θ as C. Note that the actual value of C can change from a line to the other. We start by rewriting

$$\begin{aligned}
\mathsf{KL}(\mu \parallel \mathsf{p}^\theta_{0:K}) &= \int \log\left(\frac{\mathsf{q}_{\mathrm{data}}(x_0)q^\eta_{1:K|0}(x_{1:K}|x_0)}{\mathsf{p}_{0:K}(x_{0:K})}\right)\mathsf{q}_{\mathrm{data}}(x_0)q^\eta_{1:K|0}(x_{1:K}|x_0)\mathrm{d}x_{0:K} \\
&= \sum_{k=1}^{K-1}\int \log\left(\frac{q^\eta_{k|k+1,0}(x_k|x_0,x_{k+1})}{p^\theta_{k|k+1}(x_k|x_{k+1})}\right) q^\eta_{k|k+1,0}(x_k|x_0,x_{k+1})q^\eta_{k+1|0}(x_{k+1}|x_0)\mathsf{q}_{\mathrm{data}}(x_0)\mathrm{d}x_{0,k,k+1} \\
&+ \int \log\left(\frac{\mathsf{q}_{\mathrm{data}}(x_0)q^\eta_{1|0}(x_1|x_0)}{p^\theta_{0|1}(x_0|x_1)}\right)\mathsf{q}_{\mathrm{data}}(x_0)q^\eta_{1|0}(x_1|x_0)\mathrm{d}x_{0:1} + C \\
&= \sum_{k=1}^{K-1}\int \mathsf{KL}(q^\eta_{k|k+1,0}(\cdot|x_0,x_{k+1}) \parallel p^\theta_{k|k+1}(\cdot|x_{k+1}))\mathsf{q}_{k+1|0}(x_0)x_{k+1}\mathsf{q}_{\mathrm{data}}(x_0)\mathrm{d}x_{0,k+1} \\
&- \int \log p^\theta_{0|1}(x_0|x_1)\mathsf{q}_{\mathrm{data}}(x_0)q^\eta_{1|0}(x_1|x_0)\mathrm{d}x_{0:1} + C\,,
\end{aligned}$$

where C is a constant that does not depend on θ. We know that

$$\mathsf{KL}(\mathcal{N}(\mu_1,\sigma_1^2\,\mathrm{I}) \parallel \mathcal{N}(\mu_2,\sigma_2^2\,\mathrm{I})) = 2^{-1}\left[2d\log(\sigma_2/\sigma_1) - d + d(\sigma_1/\sigma_2)^2 + \|\mu_2-\mu_1\|^2/\sigma_2^2\right]\,,$$

thus

$$\mathsf{KL}(q^\eta_{k|k+1,0}(\cdot|x_0,x_{k+1}) \parallel p^\theta_{k|k+1}(\cdot|x_{k+1})) = \eta_k^{-2}\left[1 - (\upsilon_k^2/\upsilon_{k+1}^2 - \eta_k^2/\upsilon_{k+1}^2)^{1/2}\right]^2\|\mathcal{D}^\theta_{0|k+1}(x_{k+1}) - x_0\|^2\,.$$

Note also that

$$\log p^\theta_{0|1}(x_0|x_1) = -\eta_0^{-2}\|\mathcal{D}^\theta_{0|1}(x_1) - x_0\|^2 + C\,.$$

The proof is finished by lemma 70. □

E.2 Preprocessing Implementation Details

Our preprocessing follows four stages.

- Align the recording-frequency of all ECGs to 250 Hz by performing down or up sampling. Thus, two consecutive points in the ECG are separated by 4ms.
- Extract R peaks from the ECG. The first principal component is extracted channel-wise from the entire ECG. Subsequently, this extracted component is processed through a Savitzky-Golay filter, characterized by an order of 3 and a window length of 15. The extraction of R-peaks is then carried out based on the methodology proposed in Brammer (2020).
- Select the window $[\mathrm{R} - 192\,\mathrm{ms}, R + 512\,\mathrm{ms}]$ containing the QRS. This window corresponds to 176 time-points as $(192 + 512)/4 = 176$.
- Normalize each ECG lead by dividing it by the maximum absolute value attained during the QRS.

Table E.1: Distribution of patients, gender and number of recorded beats among train, test and MI sets.

	Train	CV	Test	MI
All (patients)	22580	2723	2864	468
Male (patients)	11722	1399	1497	343
Female (patients)	10858	1324	1367	125
All (beats)	214460	25694	27221	44911
Mean (beats)	9.5 +/- 0.1	9.4 +/- 0.2	9.5 +/- 0.2	96 +/- 5

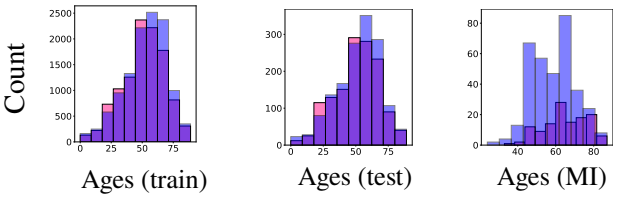


Figure E.1: Female (pink), male (blue) ages histograms in training (left), test (middle), MI (right) sets.

E.3 Architecture Details

We implement a very close architecture to Karras et al. (2022) and available at https://github.com/NVlabs/edm as well as training procedure. The main difference is that we replaced the 2D convolutional layers by 1D ones in every UNet. The final network use the following parameters:

- First embedding dimension: $c = 192$,
- Number of Unet blocks per resolution: 2,
- Number of resolutions: 1,
- Dropout probability 0.10,
- Attention resolution: [88, 44, 22].

For the training, the following configuration was used:

- learning rate: 10^{-4},
- Number of epochs: 10^4,
- Batch Size: 1024,
- Exponential moving average coefficient: 0.9999.

For the (forward diffusion) we used the following parameters:

- $\sigma_{\text{min}} = 2 \times 10^{-4}$,
- $\sigma_{\text{max}} = 80$,
- $\sigma_{\text{data}} = 0.5$,
- Importance law of σ for training: $\text{Log}\,\mathcal{N}(-1.2, 1.2^2\,\text{I})$.

E.4 Deeper or Unconditioned Denoisers

In this section we test two alternative architectures: a DDM unconditioned on the patient information $\mathcal{P}$ (1) and a deeper DDM (2).

To obtain comparable EMD for both conditioned and unconditioned ECGs with $\mathcal{P}$, unconditioned generated ECGs are concatenated with A, S, RR features randomly selected from the test set. We find

that conditioning over A, S, RR leads to smaller EMD.

The U-Net blocks can be stacked and a common usage in the literature is to combine several U-Net on different resolution levels, that are obtained by downsampling the data before feeding it to each block U-Net. We have experimented with using 2 resolution levels for the U-Net but found no signicant gains w.r.t. using only one level.

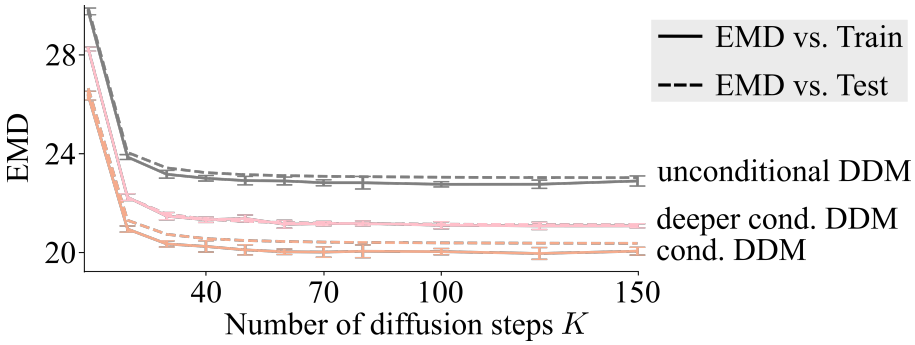


Figure E.2: EMD of generated ECGs vs. test (dotted) and train (plain), w.r.t diffusion steps. Small conditioned (resp. uncond.) network in orange and (resp. gray). Deeper conditioned network in pink. EMD of test (resp. noisy-test) vs. train in red (resp. blue). Error bars correspond to different training batches of size 2864.

E.5 SMC Algorithm

In this section we first provide the SMC algorithm 12.

Algorithm 12 SMC

Input: observation y, number of diffusion steps K, number of particles M
Operations involving index i are repeated for $i \in [1:M]$
Initialization: $\xi_K^i \sim \lambda$
for $k = K-1$ **to** 0 **do**
  $I_k^i \sim \mathrm{Cat}\big(\{\omega_k(\xi_{k+1}^j)/\sum_{i=1}^M \omega_k(\xi_{k+1}^i)\}_{j=1}^M\big)$
  $\xi_k^i \sim p_k^y(\cdot|\xi_{k+1}^{I_k^i})$
end for
Output: $\xi_0^{1:M}$

E.6 Heuristic for the Potential

Preliminary definitions.

We preface this section with some measure theory notations and definitions of a few quantities that will be used throughout.

For $d \in \mathbb{N}$, we denote $\mathcal{B}(\mathrm{R}^d)$ the Borel set in R^d. For a probability measure $\mu \in \mathbb{R}^d$ and $f : \mathbb{R}^d \to \mathbb{R}$ a bounded measurable function, we write $\mu(f) := \int f(x)\mu(\mathrm{d}x)$ the expectation of f under μ and if $K(\mathrm{d}x|z)$ is a transition kernel we write $K(f)(z) := \int f(x)K(\mathrm{d}x|z)$. For $a \in \mathbb{R}^d$, we define the Dirac distribution δ_a as the distribution such that for all $B \in \mathcal{B}(\mathrm{R}^d)$, $\delta_a(B) = 1$ if $a \in B$ else $\delta_a(B) = 0$.

Heuristic

In this section we give an heuristic for deriving the potential (6.13) for the VE framework. We assume that we measure partially a new ECG through a subset of indices $\mathcal{I} = \{(\ell, t) \in [1:L] \times [1:T]\} \neq$

$[1:L]\times[1:T]$. For any $(\ell,t)\in\mathcal{I}$, the observation follows $y\sim X_0[\ell,t]+\sigma\epsilon_{\ell,t}$ where σ is the *known* measurement noise, supposed uniform for the sake of simplicity. We aim at sampling x_0 from the posterior $X_0|y,\sigma$, with p.d.f.

$$\phi_0^y(x_0) := g_0^y(x_0)\mathsf{p}_0(x_0)/\mathcal{Z}$$

where

$$\begin{aligned} g_0^y(x_0) &:= \textstyle\prod_{(l,t)\in\mathcal{I}}\mathcal{N}(x_0[\ell,t];y[\ell,t],\sigma^2),\\ \mathsf{p}_0(x_0) &:= \textstyle\int\lambda(x_K)\prod_{j=K}^1 p_{j-1|j}(x_{j-1}|x_j)\mathrm{d}x_{1:K},\\ \mathcal{Z} &:= \textstyle\int g_0^y(x)\mathsf{p}_0(x)\mathrm{d}x. \end{aligned}$$

We suppose that there exists a diffusion step $\tau\in[0:K]$ such that $\upsilon_\tau^2=\sigma^2$, i.e., such that the level of measurement noise equals the level of diffusion noise. This assumption is realistic for a large number of diffusion steps K. We can then rewrite the posterior p.d.f. as follow

$$\begin{aligned} \phi_0^y(x_0) &= \textstyle\prod_{(l,t)\in\mathcal{I}}\mathcal{N}(x_0[\ell,t];y[\ell,t],\upsilon_\tau^2)\mathsf{p}_0(x_0)/\mathcal{Z}\\ &= \textstyle\prod_{(l,t)\in\mathcal{I}} q_{\tau|0}(y[\ell,t]|x_0[\ell,t])\mathsf{p}_0(x_0)/\mathcal{Z}\,. \end{aligned} \tag{E.4}$$

We make the idealistic assumption that for all $x_{0:K}$, $\mathsf{q}_{0:K}(x_{0:K})=\mathsf{p}_{0:K}(x_{0:K})$, which implies that $\mathsf{q}_k=\mathsf{p}_k$ and that

$$\begin{aligned} q_{k|0}(x_k|x_0)\mathsf{p}_0(x_0) &= q_{k|0}(x_k|x_0)\mathsf{q}_0(x_0)\\ &= p_{0|k}(x_0|x_k)\mathsf{p}_k(x_k)\,. \end{aligned} \tag{E.5}$$

However, we cannot directly replace the element-wise forward process in E.4 using this assumption as only the indices of $\mathcal{I}$ are taken into account. Hence, we introduce the following integral form of the likelihood

$$\begin{aligned} g_0^y(x_0) &= \textstyle\prod_{(l,t)\in\mathcal{I}} q_{\tau|0}(y[\ell,t]|x_0[\ell,t])\\ &= \textstyle\prod_{(l,t)\in\mathcal{I}}\int q_{\tau|0}(x_\tau[\ell,t]|x_0[\ell,t])\delta_{y[\ell,t]}(\mathrm{d}x_\tau[\ell,t])\\ &= \textstyle\int q_{\tau|0}(x_\tau|x_0)\psi(\mathrm{d}x_\tau)\,, \end{aligned} \tag{E.6}$$

where

$$\psi(\mathrm{d}x_\tau) := \textstyle\prod_{(l,t)\in\mathcal{I}}\delta_{y[\ell,t]}(\mathrm{d}x_\tau[\ell,t])\prod_{(l,t)\notin\mathcal{I}}\mathrm{d}x_\tau[\ell,t]\,,$$

which also means that ψ is a measure that has a singleton in every observed coordinate and the Lebesgue measure on the non observed coordinates. By plugging E.6 into E.4 and replacing the backward with the forward process we obtain

$$\begin{aligned} \phi_0^y(x_0) &= \textstyle\int q_{\tau|0}(x_\tau|x_0)\mathsf{p}_0(x_0)\psi(\mathrm{d}x_\tau)/\mathcal{Z}\\ &= \textstyle\int p_{0|\tau}(x_0|x_\tau)\mathsf{p}_\tau(x_\tau)\psi(\mathrm{d}x_\tau)/\mathcal{Z}\,. \end{aligned}$$

This shows that sampling from ϕ_0^y can be obtained by sampling first from

$$\phi_\tau := \mathsf{p}_\tau(x_\tau)\psi(\mathrm{d}x_\tau)/\mathcal{Z} \tag{E.7}$$

and then propagating the samples via $p_{0|\tau}(x_0|x_\tau)$, i.e., we define $\phi_k(x_k)\propto g_k(x_k)\mathsf{p}_{k-1}(x_{k-1})$ with $g_k(x_k):=1$ for $k<\tau$.

Now we derive a sequence $\{\phi_k\}_{k>\tau}$ to sample recursively from E.7. We introduce the extended distribution

$$\phi_{\tau:K}(\mathrm{d}x_{\tau:K}) := \mathsf{p}_{\tau:K}(x_{\tau:K})\psi(\mathrm{d}x_\tau)\mathrm{d}x_{\tau+1:K}/\mathcal{Z}\,,.$$

where $\mathsf{p}_{\tau:K}(x_{\tau:K}) := \lambda(x_K) \prod_{k=K}^{\tau+1} p_{k-1|k}(x_{k-1}|x_k)$. We can write the marginal distribution of $\phi_{\tau:K}$ for $k > \tau$, using the definition (6.5)

$$\phi_k(x_k) = \int p_{\tau|k}(x_\tau|x_k)\mathsf{p}_\tau(x_\tau)\psi(\mathrm{d}x_\tau)/\mathcal{Z}\,.$$

Then using the assumption in E.5

$$\begin{aligned}\phi_k(x_k) &= \int q_{k|\tau}(x_k|x_\tau)\mathsf{p}_k(x_k)\psi(\mathrm{d}x_k)/\mathcal{Z} \\ &= \prod_{(l,t)\in\mathcal{I}} q_{k|\tau}(x_k[\ell,t]|y[\ell,t])\mathsf{p}_k(x_k)/\mathcal{Z} \\ &= \prod_{(l,t)\in\mathcal{I}} \mathcal{N}(x[\ell,t]; y[\ell,t], v_k^2 - \sigma^2)\mathsf{p}_k(x_k)/\mathcal{Z}\,,\end{aligned}$$

where we recognize a product between the marginal law at time k and a potential function of the form

$$g_k(x) := \prod_{(l,t)\in\mathcal{I}} \mathcal{N}(x[\ell,t]; y[\ell,t], v_k^2 - \sigma^2)\,.$$

Note that ϕ_τ introduced in E.7 does not admit a density with respect to p_τ, because of the singleton measures on the observed coordinates. To mimic the effect of the singleton while still admitting a density with respect to p_τ, we use the following approximation

$$\phi_\tau(x_\tau) \approx \prod_{(l,t)\in\mathcal{I}} \mathcal{N}(x_s[\ell,t]; y[\ell,t], \varepsilon^2)\mathsf{p}_\tau(x_\tau)/\mathcal{Z}\,,$$

for a small ε.

Figure E.3 provides a visual representation of the sampling of ECGs x_0 from the posterior $X_0|y, \sigma = 0.1$, with $\mathcal{I} = [1:3] \times [1:T]$ using the sequence of instrumental laws $\{\phi_k\}_{k\in[0:K]}$. At the beginning of the generation (first column), the generated samples (in blue) are scattered and the standard deviation of the guiding function is high. As generation progresses (from left to right), the standard deviation of the guiding function decreases until it approaches 0. From then on (last two columns), $k < \tau$ and samples are generated by the backward process solely.

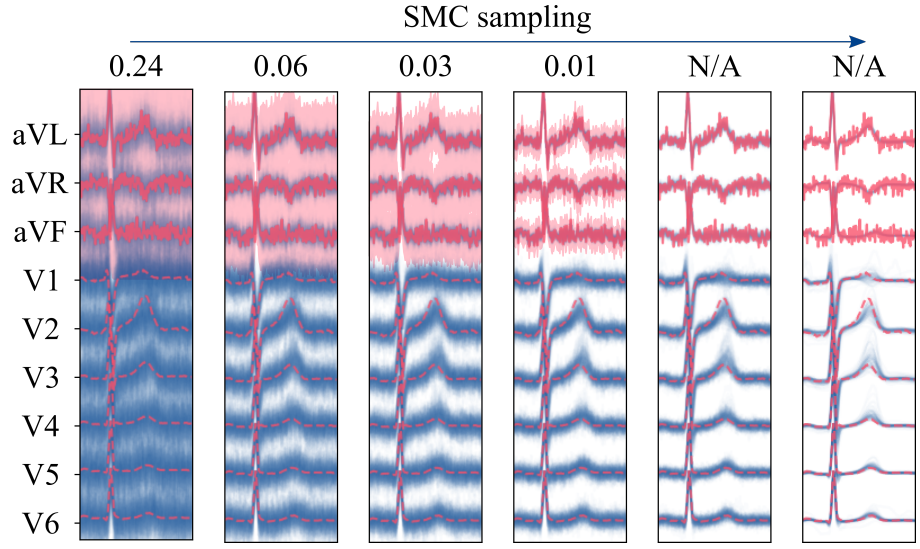


Figure E.3: Conditional generation example. Observation: (aVL, aVR, aVF) with $\sigma = 0.1$. Red solid/dashed lines: observed/real signal. Shaded zone: observed signal plus $3 \times$ std of the guiding function 6.13, std values on top. Blue: posterior samples.

E.7 Proposal Potential and Weight

Using conjugate formulas we compute the proposal kernel and the weights defined in (6.8) used in SMC algorithm

$$p_k^y(x_k|x_{k+1}) = \prod_{\ell\in\mathcal{V}_k}\prod_{k=1}^{\mathrm{T}_y}\mathcal{N}(x_k[\ell,t];\mu_{k,y}(x_{k+1})[\ell,t],\frac{\eta_k^2\sigma_{k,y}^2}{\eta_k^2+\sigma_{k,y}^2})\prod_{\ell\notin\mathcal{V}_k}\prod_{t\notin[1:\mathrm{T}_y]}\mathcal{N}(x_k[\ell,t];\mu_k(x_{k+1})[\ell,t],\eta_k^2)\,,$$

and

$$\omega_k^y(x_{k+1}) = \prod_{\ell\in\mathcal{V}_k}\prod_{t=1}^{\mathrm{T}_y}\mathcal{N}\left(\mu_k[\ell,t];y[\ell,t],\eta_{k+1}^2+\sigma_{k,y}^2\right)\Big/\prod_{\ell\in\mathcal{V}_{k+1}}\prod_{t=1}^{\mathrm{T}_y}\mathcal{N}(x_{k+1}[\ell,t];y[\ell,t],\sigma_{k+1,y}^2)\,,$$

where

$$\begin{aligned}
&\sigma_{k,y}^2 := \upsilon_k^2-(1-\delta)\sigma_\ell^2\\
&\mu_k := \boldsymbol{\mu}_k(x_{k+1},\mathcal{D}_{0|k+1}(x_{k+1}))\\
&\mu_{k,y}(x_{k+1})[\ell,t] := (\eta_k^2 y[\ell,t]+\sigma_{k,y}^2\mu_k[\ell,t])/(\eta_k^2+\sigma_{k,y}^2)\,.
\end{aligned}$$

E.8 Number of particles

As the number of particles, denoted as M, increases, we observe a corresponding decrease in the discrepancy between the target posterior distribution and the distribution of particles generated by algorithm 12. A critical question arises: what is the optimal value for M that strikes a balance between accuracy and computational efficiency? To approach this question, we first selected a patient from the test dataset and used algorithm 12 to generate 10^3 samples with a high particle count of $M = 10^4$. We consider these samples as our reference representing the target posterior distribution.

We then generated 10^3 samples with algorithm 12 for different values of M and calculated the Earth Mover's Distance (EMD) relative to the reference samples. This process helps us to evaluate the convergence of the distribution generated by the algorithm to the posterior as M varies. Figure E.4 illustrates the relationship between M and the EMD. From this analysis, $M = 50$ provides an effective equilibrium that provides a reasonable approximation to the posterior distribution while ensuring manageable inference times.

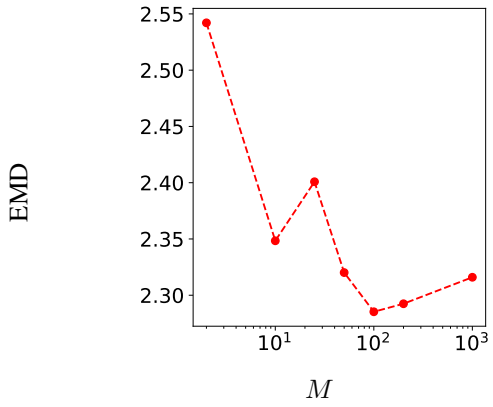


Figure E.4: EMD distance between 1000 *samples from algorithm 12 with* M *particles and* 1000 *samples of algorithm 12 with* 10^5 *particles, that is considered the standard samples.*

E.9 Baselines

In this section, we provide implementation details for testing the WGAN, DAE, AAE, and OOD baselines in the same setup as our approach.

In the paper Adib et al. (2022), the WGAN is conditioned on 15 categorical heart disease labels. These labels are embedded into a vector of size 100 and concatenated with the latent variable before being inputted into the generator. They are also embedded into a vector of length T (where T is the temporal length of the signal) and then concatenated with the cardiac signal (fake or real) before being inputted into the critic. Embedding maps variables with a finite number of possible values (i.e., categorical variables) into a vectorized representation. However, since in our DDM we condition on scalar variables such as the RR interval, in order to compare the results obtained with our DDM and the WGAN, we instead use a multi-layer perceptron (MLP) with the following architecture: a linear layer from 4 to 864, a 1D normalization layer, LeakyReLU, and a linear layer from 864 to 64. This MLP maps the 4-size feature vector $(\tilde{A}, \tilde{S}, \tilde{\mathrm{RR}})$ to a 64-vector, which is then used in the same way as the embedding was in the original paper.

In the paper Chiang et al. (2019), a DAE is used to denoise ECGs containing multiple heartbeats. Their proposed architecture consists of 6 convolutional layers with a kernel size of 16 and 6 deconvolutional layers with a kernel size of 16. Since in our experiments the input signals are single heartbeats, we use a kernel size of 4 instead of 16 to be able to apply the DAE to shorter signals. We pretrain the DAE to denoise corrupted signals with Gaussian noise with a standard deviation sampled from an exponential distribution with a rate parameter of 0.2, by minimizing the mean squared error between the real and denoised heartbeats, using the Adam optimizer for 50 epochs. We use this model for two experiments: ECG denoising and anomaly detection. For the latter, we use the mean squared error (MSE) between the input and decoded heartbeat as the anomaly score.

For the AAE, we employ the same architecture and training as Shan et al. (2022). We use this model for two tasks: ECG denoising and anomaly detection (which is the task solved by this model in Shan et al. (2022)). For the first task, we denoise corrupted ECG signals simply by encoding and decoding them with the AE module.

Finally, we also use the out-of-distribution score proposed by Ciosek et al. (2020) for anomaly detection (in addition of using it for generative evaluation in section 6.5.3). The anomaly score is the MSE between the output of the random fixed network and the trained network.

E.10 Additional Results

In this section we provide supplementary results for the experiments on ECG missing lead reconstruction and the prediction of corrected QT: we provide R^2-score between predicted and real lead ℓ, with 95%-CLT intervals over the test-set for missing lead reconstruction using `NC-MCGdiff` and Dower matrices in table E.2; we provide the R^2-score between QT measured vs. regressed (intercept: QT_0^c, slope: QT_1^c) as a function of RR, in generated samples, with 95%-CLT intervals over the test-set, for several corrected QT formulas in table E.3.

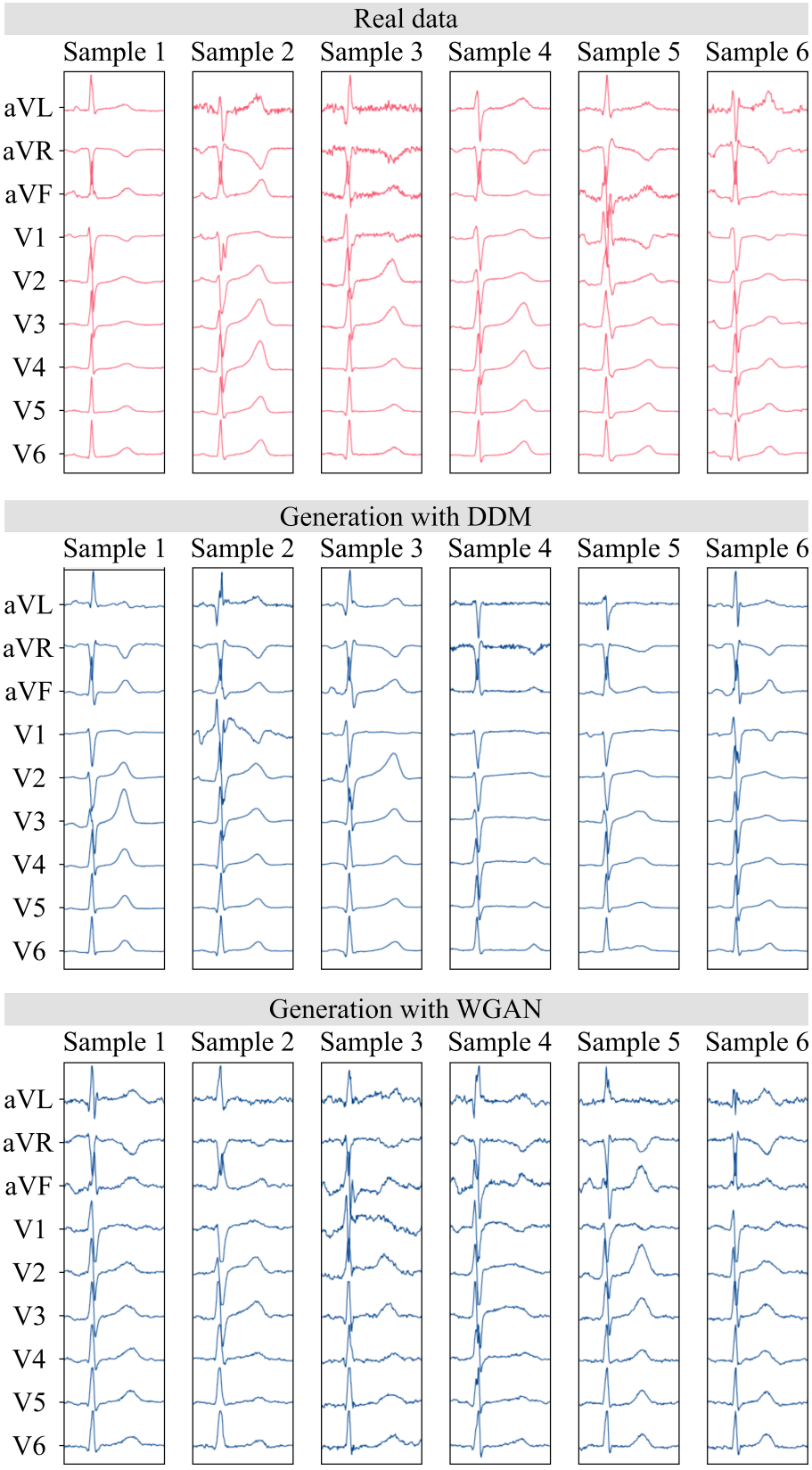


Figure E.5: Real and generated ECG heart beat with DDM and WGAN.

Table E.2: R^2-score between predicted and real lead ℓ, with 95%-CLT intervals over the test-set.

Lead (ℓ)	`NC-MCGdiff`	Dower
V1	0.98 ± 0.01	0.70 ± 0.05
V2	0.99 ± 0.00	0.78 ± 0.05
V3	0.99 ± 0.00	0.75 ± 0.06
V4	0.99 ± 0.00	0.87 ± 0.03
V5	0.98 ± 0.02	0.86 ± 0.08
V6	0.99 ± 0.01	0.85 ± 0.03

Table E.3: R^2-score between QT measured vs. regressed (intercept: QT_0^c, slope: QT_1^c) as a function of RR, in generated samples, with 95%-CLT intervals over the test-set.

Method	R^2-score	Expression
Framingham	0.88 ± 0.03	$\mathrm{QT} = \mathrm{QT}_0^c + 0.154(1 - \mathrm{RR})$
Bazett	0.47 ± 0.04	$\mathrm{QT} = \mathrm{QT}_1^c \sqrt{\mathrm{RR}}$
Baz. (offset)	0.98 ± 0.00	$\mathrm{QT} = \mathrm{QT}_0^c + \mathrm{QT}_1^c \sqrt{\mathrm{RR}}$
Fridericia	0.94 ± 0.02	$\mathrm{QT} = \mathrm{QT}_1^c \sqrt[3]{\mathrm{RR}}$
Frid. (offset)	0.98 ± 0.00	$\mathrm{QT} = \mathrm{QT}_0^c + \mathrm{QT}_1^c \sqrt[3]{\mathrm{RR}}$

ECOLE
DOCTORALE
DE MATHEMATIQUES
HADAMARD

Titre : Modèles génératifs pour le traitement des données du type électrocardiogramme: théorie et application.

Mots clés : Apprentissage par machine,Modèles génératifs,Cardiologie,Apprentissage profond,Apprentissage auto supervisé.

Résumé : Cette thèse apporte des contributions au vaste domaine des modèles génératifs, avec un intérêt particulier pour l'application de tels modèles aux données d'électrocardiogramme (ECG) dans le cadre de l'inférence et de la quantification de l'incertitude. Dans une première partie, nous développons deux méthodes novatrices pour réduire le biais dans les méthodes d'échantillonnage d'importance et de Monte Carlo séquentiel (SMC), qui sont deux outils importants de l'inférence bayésienne. Les algorithmes résultants peuvent être considérés tous deux comme des "enveloppes" autour d'algorithmes existants actuels, offrant une réduction de biais sans grande augmentation du temps de calcul. Nous présentons également de nouvelles bornes de convergence non asymptotiques pour l'utilisation de ces algorithmes dans l'apprentissage de paramètres dans les modèles de Markov cachés (HMM). Dans une deuxième partie, nous nous concentrons sur l'utilisation du SMC pour résoudre des problèmes inverses linéaires bayésiens, avec des modèles génératifs servant de priors informatifs. Cette approche est particulièrement intéressante pour améliorer la résolution des problèmes inverses rencontrés dans divers domaines scientifiques. Enfin, nous appliquons cette méthodologie à plusieurs problèmes inverses basés sur l'ECG, notamment la complétion de pistes manquantes et la détection hors distribution. Les résultats de ces applications démontrent l'efficacité et la polyvalence des modèles génératifs proposés pour relever des défis concrets dans le contexte de l'analyse des données ECG.

Title : Generative models for ECG data: theory and application.

Keywords : Machine Learning,Generative model,cardiology,Deep Learning,Self-supervised learning.

Abstract : This thesis contributes to the vast domain of Generative models, with a particular interest in applying such models to electrocardiogram (ECG) data for inference and uncertainty quantification. In a first part, we develop two novel methods for reducing bias in Importance Sampling and Sequential Monte Carlo (SMC) methods, which are two important tools of Bayesian inference. The issuing algorithms can both be viewed as a wrapper around current existing algorithms providing effortless bias reduction. We also provide new non-assymptotic convergence bounds for using such algorithms for parameter learning in Hidden Markov Models (HMM). In a second part, we focus on using SMC for solving Bayesian linear inverse problems with generative models serving as informative priors. Finally, we apply this method on several ECG based inverse problems, namely missing lead completion and out-of-distribution detection.

Institut Polytechnique de Paris
91120 Palaiseau, France