Optimal Transport based Simulation Methods for Deep Probabilistic Models

James Thornton

St Peter's College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity 2023

Acknowledgements

First and foremost I would like to express my utmost gratitude to my supervisors George Deligiannidis and Arnaud Doucet. I would also like to thank Valentin De Bortoli, whom I worked with closely throughout my PhD, in particular on the Schrödinger Bridge. I was extremely fortunate to learn and be guided by these mentors, both on a personal level and research-wise. My outlook on research, approach and views have been heavily shaped by them.

My research career has been marked by a few key pivotal moments; and the people behind them. I am grateful to Anthony Lee for his supervision during my undergraduate thesis. This ultimately led to my interest in pursuing a PhD. I would also like to thank my former colleagues at BlackRock for giving me the kick I needed to start the PhD. I learnt a lot, and have fond memories of those few long years at Drapers' Gardens. Once again I am grateful to Arnaud Doucet for giving me freedom to pursue my research interests, and encouraging me to broaden my research scope to the intersection of machine learning and Monte Carlo; then later optimal transport. In retrospect this was perhaps the best decision possible in terms of personal interest; timing and impact - coinciding with the emergence of diffusion models and leading to some early contributions. I thank Anthony Caterini for organizing the optimal transport (OT) reading group at Oxford, going through the fantastic book of Marco Cuturi and Gabriel Peyré. Prior to this I knew very little about optimal transport, but I had an immediate attraction, and OT became the recurring theme of my thesis.

I want to acknowledge St Peter's College for support during my time at Oxford and the Department of Statistics for providing a fantastic work environment - in particular Joanna Stoneham for ensuring everything ran smoothly. I would also like to express my appreciation to my cohort on the OxWaSP CDT programme.

Finally, I would like to thank my parents. It would have been very difficult without their support.

Abstract

Deep probabilistic models have emerged as state-of-the-art for high-dimensional, multi-modal data synthesis and density estimation tasks. By combining abstract probabilistic formulations with the expressivity and scalability of neural networks, deep probabilistic models have become a fundamental component of the machine learning toolbox. Such models still have a number of limitations however. For example, deep probabilistic models are often limited to gradient based training and hence struggle to incorporate non-differentiable operations; they are expensive to train and sample from; and often deep probabilistic models do not leverage prior geometric and problem-specific structural knowledge.

This thesis consists of four contributing pieces of work and advances the field of deep probabilistic models through optimal transport based simulation methods. First, by using regularized optimal transport via the Sinkhorn algorithm, we provide a theoretically grounded and differentiable approximation to resampling within particle filtering. This allows one to perform gradient based training of state space models, a class of sequential probabilistic model, with end-to-end differentiable particle filtering. Next, we explore initialization strategies for the Sinkhorn algorithm to address speed issues. We show that careful initializations result in dramatic acceleration of the Sinkhorn algorithm. This has applications in differentiable sorting; clustering within the latent space of a variational autoencoder; and within particle filtering. The remaining two works contribute to the field of diffusion based generative modelling through the Schrödinger Bridge. First, we connect diffusion models to the Schrödinger Bridge, coined the *Diffusion Schrödinger Bridge*. This methodology enables accelerated sampling; data-to-data simulation, and a novel way to compute regularized optimal transport for high dimensional, continuous state-space problems. Finally, we extend the Diffusion Schrödinger Bridge to the Riemannian manifold setting. This allows one to incorporate prior geometric knowledge and hence enable more efficient training and inference for diffusion models on Riemannian manifold valued data. This has applications in climate and Earth science.

Contents

Chapter 1

Introduction

Observed phenomena are often of high-dimension, distributionally multi-modal and arise through random, complex systems that are not fully understood. It is the role of scientists to better understand these observations in the physical and social world around us. Of particular interest to statistical machine learning researchers is developing data-driven models to simulate and predict random quantities of interest, and their interactions. Such data-driven machine learning approaches typically consist of utilizing large datasets and expressive model parameterizations to bridge the gap between domain knowledge and empirical evidence.

With the advancement of technology, we are now able to capture, store and access significant and ever increasing volumes of data. Coupled with modern compute resources, vast amounts of data can be used by practitioners to train large and flexible neural network parameterized models. The synergy between expressive neural networks and theoretically grounded statistical methods offers the promise of a principled, scalable way to accurately represent complex relationships between random quantities of interest. Deep probabilistic models live at this intersection and have been applied to many domains with great success, including: classical machine learning tasks in vision and language; statistical inference problems in time-series, prediction or density estimation; and in natural science applications.

Despite rapid progress there are still many open challenges. Deep probabilistic models are resource expensive, both in terms of training and in deployment. It is not clear how to train and simulate from probabilistic models efficiently; especially with iterative diffusion models. Nor is it always clear how to incorporate existing problem specific knowledge or structure into deep probabilistic models. Intuitively and experimentally, incorporating existing knowledge allows one to develop more effective loss objectives and more efficient models, hence reduces the data requirement and parameter count for learning performant models. Ultimately, incorporating domain knowledge reduces the gap that neural networks need to bridge by learning from data. There are also many challenges in combining theoretically grounded statistical methods, which may involve non-differentiable operations such as sorting, clustering or resampling; with the expressiveness of deep networks, which are typically limited to gradient based training.

This thesis contributes to the advancement of deep probabilistic models, primarily by introducing new methodology based on the use of optimal transport and simulation methods.

1.1 Contributions

This thesis consists of multiple works which can roughly be split into two sections. The first half of this thesis consists of using discrete, entropically regularized optimal transport (OT) to derive novel and improved training schemes for deep probabilistic models. The second half of the thesis explores new methodology at the intersection of entropically regularized optimal transport for continuous state-space and diffusion models through the Diffusion Schrödinger Bridge and Riemannian extension.

Simulation methods. Simulation is a ubiquitous term, used rather liberally and often interchangeably with Monte Carlo. In this work, *simulation* is used to refer to the evaluation of stochastic procedures, encompassing: Monte Carlo integration, Markov chain Monte Carlo, probabilistic models and the simple realisation of random

variables. Throughout this thesis it shall be used in two contexts. Firstly simulation is used in the Monte Carlo integration sense to refer to approximating losses which are often expressed as intractable integrals over random variables. Secondly, running the generative process of a probabilistic model is also referred to as simulation. In some approaches, simulating from the probabilistic model is used directly in approximating the training loss, e.g. GANs. However, other scalable methods like diffusion models have different simulation procedures during training and deployment.

Differentiable Particle Filtering. The first publication [19] of this thesis is detailed in **Chapter 3** and considers utilizing the Sinkhorn algorithm to enable end-to-end differentiable particle filtering. This permits training neural network parameterized state-space models, thus leveraging the sequential structure of the problem, as well as the expressive power of neural networks in a principled manner.

Particle filters are a class of Monte Carlo methods used to perform state inference and likelihood estimation in state space models [30]. Given sequential unobserved latent states $(X_t)_t$ and observations $(Y_t)_t$ indexed by time $t \in \{1, \ldots, T\}$, a state-space model is a sequential probabilistic model characterized by a transition model over latent states, expressed as a density $f_\theta(x_t|x_{t-1})$, and an observation model $g_\theta(y_t|x_t)$. This has applications across scientific domains including robotics, econometrics and epidemiology [18, 29, 31, 33]. The particle filter provides an asymptotically unbiased log-likelihood estimate of the observations, $\log p(y_{1:T})$, which can then be used to learn the parameters of the transition and observation models in a principled manner. Particle filtering consists of sequential application three primary operations:

1. Proposal. Propose particles for the hidden state at each time t, outputting the proposal particle distribution.
2. Weighting. Assign an importance weight to each proposed particle according to the proposal density, state-space model transition density and observation density. This step gives the weighted filtering particle distribution.

3. Resampling. Resample proposed particles according to importance weights to prevent weight degeneracy. This step gives the unweighted filtering particle distribution.

Typical resampling procedures are non-differentiable, this limits the use of particle filters in training deep neural network parameterized state-space models through gradient-based optimization. One may instead reframe the resampling operation in particle filtering as sampling through a coupling between the empirical proposal and weighted filtering particle distributions [68]. Minimizing variance along this coupling is equivalent to optimal transport. Sampling across such couplings retains theoretical guarantees of standard resampling but also 'reduces' the discontinuities and hence increases the 'smoothness' of the likelihood function. Computing this transport coupling using the Sinkhorn algorithm is differentiable, then taking the average across the rows of the coupling matrix, also known as the barycentric projection or ensemble transform [68], rather than sampling the coupling introduces a slight, quantifiable bias but enables differentiable resampling, hence end-to-end differentiable particle filtering.

Initializing Sinkhorn Potentials. Despite the success of embedding Sinkhorn layers in neural networks, it may take many iterations for the Sinkhorn algorithm to converge, moreover, each iteration of the Sinkhorn algorithm has complexity $\mathcal{O}(n^2)$, where n is the number of atoms in each of the discrete marginal measures. Hence it can be time-consuming for the Sinkhorn algorithm to converge. This issue is compounded when, like in the case of differentiable particle filtering, there are multiple Sinkhorn layers embedded in the forward pass of the probabilistic model.

The convergence speed of the Sinkhorn algorithm depends on two terms. Firstly, on some conditioning constant of the Gibbs kernel $e^{-c_{i,j}/\varepsilon}$, for ground cost $(c_{i,j})_{i,j}$ and secondly on how close the initial Sinkhorn potentials are to the optimal potentials, see [66, Theorem 4.1]. There have been many attempts to accelerate the Sinkhorn algorithm including using Anderson acceleration [17] or momentum approaches,

[56, 90]. In the second publication of this thesis [91], **Chapter 4**, we investigate acceleration through initialization.

If the initialized Sinkhorn potentials are at the optima then no further iterations are required. Informally, if the transport problem $\mathrm{OT}_1 = (\alpha, \beta, c, \epsilon)$ is 'close' to a similar problem $\mathrm{OT}_2 = (\tilde{\alpha}, \tilde{\beta}, c, \tilde{\epsilon})$, then the optimal potentials will also be close, [61]. The premise of the work in **Chapter 4**, it to construct sequences of OT problems that are cheap to solve or approximate, but which converge to the original OT problems of interest, in terms of the marginal measures or regularization parameter. We then use the cheaper solutions to the approximate problem to initialize the original, more difficult problem. A number of initializers are proposed for a variety of common problems involving Sinkhorn layers within neural networks, in particular: for sorting [21], Gaussian and Gaussian mixture initalizers for clustering of latent embeddings, such as within autoencoders for example, as used in [13, 19, 41]; and a subsample initializer for cases where n, the number of points of the discrete measures, is large. The initializers show dramatic speed-up for a variety of tasks.

Diffusion Schrödinger Bridge. In the third publication of this thesis [7], **Chapter 5**, a novel generalization of diffusion models is introduced and builds a connection between optimal transport and diffusion models. The core idea behind this work is that each reverse diffusion learns a diffusion process that minimizes the Kullback–Leibler divergence to the forward process; iterating this time-reversal corresponds to the iterative proportional fitting procedure (IPF) [38] which is the generalization of the Sinkhorn algorithm [20, 81] to continuous state-space. The IPF procedure converges to the solution to the Schrödinger bridge problem [74], which also provides an approximate solution to high dimensional, regularized OT.

Unlike for traditional diffusion model training schemes, the iterative time-reversal approach does not require the forward noising process to converge to a simple prior distribution, but instead forces convergence by learning a new forward process during alternate IPF steps. This means that the corresponding reverse process can be significantly shorter than in regular diffusion model approaches, leading to faster

simulation. Given the forward process is no longer required to converge, one is no longer restricted to Gaussian prior and can instead initialize the reverse process from another related dataset - resulting in data-to-data simulation. This can be used for example in image-to-image restoration tasks or more generally for other conditional generative modeling tasks. Indeed, the third manuscript of this thesis was one of the first, if not the first, work to introduce image-to-image diffusion models; one of the first acceleration techniques for diffusion models; and the first diffusion model with non-linear forward process. Although surpassed in performance by other methods, it remains complementary to the other approaches that have since been used.

Riemannian Diffusion Schrödinger Bridge. Many real-world data lives on Riemannian manifolds. This includes Earth and climate data [53, 64]; protein or molecular modelling [76] and robotics [34, 75]. By incorporating this geometric prior knowledge, it is hoped that one may obtain more efficient generative probabilistic models, often requiring fewer parameters to train and easier to sample.

The next manuscript of this thesis [93] is detailed in **Chapter 6** and extends the Diffusion Schrödinger bridge methodology to the Riemannian setting. The training and sampling of diffusion models on Riemannian manifolds differs from those of traditional Euclidean diffusion models [23]. Instead of a linear diffusion for the forward noising process as is typical in Euclidean diffusion models, one instead requires Brownian motion and its extensions on a manifold. Such manifold constrained diffusion processes often do not have a closed-form and require simulation. One can sample a diffusion path on a manifold using geodesic-random walk, which is the Riemannian counterpart to the Euler–Maruyama method. Brownian motion on a compact manifold converges to a uniform distribution, which is then used to initialize the reverse, generative process. In [23, 93] and **Chapter 6**, we detail how to perform the time-reversal for Riemannian manifold Brownian motion.

Iterative time-reversal on Riemannian manifolds and the Riemannian Diffusion Schrödinger Bridge is then introduced in [93] and **Chapter 6**. This procedure

consists of a time-reversal for guided diffusions on manifolds, this permits data-to-data generation and enables practitioners to condition generative models to be close to a known dataset. Additionally, the Riemannian Diffusion Schrödinger Bridge enables acceleration for Riemannian diffusion models, where many acceleration methods for the Euclidean space are no longer applicable.

1.2 Thesis Structure

This thesis is presented in an integrated format whereby chapters 3- 6 are each self-contained published papers, followed by individual author contributions. Background literature on optimal transport and diffusion models is presented in **Chapter 2**, alongside insights and common sources of confusion.

- **Chapter 3**: "Differentiable Particle Filtering via Entropy Regularized Optimal Transport" - **James Thornton***, Adrien Corenflos*, George Deligiannidis, Arnaud Doucet, published at the *International Conference on Machine Learning* (ICML), 2021 (oral).
- **Chapter 4**: "Rethinking Initialization of the Sinkhorn Algorithm" - **James Thornton** and Marco Cuturi, published at the *International Conference on Artificial Intelligence and Statistics* (AISTATS), 2023 (oral).
- **Chapter 5**: "Diffusion Schrödinger Bridge with Applications to Score-Based Generative Modeling" - Valentin De Bortoli, **James Thornton**, Jeremy Heng and Arnaud Doucet, published at the *Conference on Neural Information Processing Systems* (NeurIPS) 2021 (spotlight).
- **Chapter 6**: "Riemannian Diffusion Schrödinger Bridge" - **James Thornton**, Michael Hutchinson, Emile Mathieu, Valentin De Bortoli, Yee Whye Teh and Arnaud Doucet, published at the *Workshop on Continuous Time Methods for Machine Learning* (ICML) 2022.
  This work was split from [23] by the same authors, which was published at NeurIPS 2022 (outstanding paper award).

1.3 Non-Included Works

I have been extremely fortunate to work with many fantastic collaborators during my PhD, on a variety of projects spanning Monte Carlo methods and statistical machine learning. Below are some of projects I have been involved with that have not been included in this thesis - in the interest of brevity or cohesion to the main theme of optimal transport.

The Masked Bouncy Particle Sampler: A Parallel, Chromatic, Piecewise-Deterministic Markov Chain Monte Carlo Method. The first work during the taught year of my PhD with my supervisors George Deligiannidis and Arnaud Doucet considered piece-wise deterministic Markov processes (PDMP) for Markov chain Monte Carlo, resulting in an unpublished paper [92].

Piecewise deterministic Markov processes provide the foundation for a promising class of non-reversible, continuous-time Markov chain Monte Carlo procedures and have been shown experimentally to enjoy attractive scaling properties in high-dimensional settings. This work introduces the Masked Bouncy Particle Sampler (MBPS), a flexible MCMC procedure within the PDMP framework that exploits model structure and modern parallel computing resources using chromatic spatial partitioning ideas from the discrete-time MCMC literature. We extend the basic procedure by introducing a dynamic factorization scheme of the target distribution to reduce boundary effects commonly associated to fixed partitioning. We establish the validity of the proposed methods theoretically and provide experimental evidence that the Masked Bouncy Particle Sampler delivers significant efficiency gains over other state-of-the-art sampling schemes for certain high-dimensional sparse models.

Simulating Diffusion Bridges via Score-Matching. This work, [46], was carried out with Jeremy Heng, Valentin de Bortoli and Arnaud Doucet as a follow-up piece to [7], detailed in **Chapter 5**. [46] was one of the early contributions to diffusion bridges via time-reversal diffusion models. Consider an unconstrained reference diffusion, p_{forward}, with fixed initialization x_0, the time-reversal, p_{backward} initialized

at x_T recovers the backward point-to-point diffusion bridge from x_T to x_0. The second time-reversal, i.e. reversal of p_{backward}, recovers the corresponding forward diffusion bridge from x_0 to x_T, this gives access to the Doob-h transform. These diffusion bridges may be used as proposal bridges for path-space rejection sampling.

Later work [57, 83], amortizes this process across pairs (x_0, x_T) for some given coupling. Although not typically a Schrödinger bridge, see comment in Section 2.2.2, this approach has shown excellent performance across trajectory inference tasks.

Riemannian Score-Based Generative Modelling. This paper, [23] with Valentin de Bortoli, Emile Mathieu, Michael Hutchinson, Yee Whye Teh and Arnaud Doucet, extends the time-reversal paradigm of diffusion models to data and diffusion trajectories constrainted to Riemannian manifolds. Included work [93], detailed in **Chapter 6** was originally part of [23], but split out. In this work, [23], a time-reversal is established for diffusions constrained to Riemannain manifolds along with convergence results for compact manifolds. Novel training and sampling schemes are derived to learn and sample from this time-reversal using specifically constructed score-networks for Riemannian manifolds. This method exhibits state-of-the art performance for generative modelling and likelihood computation tasks for manifold valued data in high-dimension.

Chapter 2

Background

2.1 Optimal Transport: From Sinkhorn to Schrödinger

This section provides a brief introduction to optimal transport (OT). The summary here is rather informal, and a more complete account may be found in [66, 72, 96]. Optimal transport is a rich area of active research, centred on computing the distance between two general measures by constructing a 'transport map' from one distribution to the other, which minimises the local pointwise 'cost', averaged according to the transport plan. The OT literature has deep roots in mathematics and connections across various surprising areas of machine learning - from generative modeling to differentiable proxies of common operations in deep learning.

Notation. In the interest of clarity and completeness, the following notation is introduced. For measurable space $(\mathsf{E}, \mathcal{E})$, let $\mathscr{P}(\mathsf{E})$ denote the space of probability measures on $(\mathsf{E}, \mathcal{E})$. Let $\mathcal{C}(E, H)$ denote the collection of continuous functions from E to H. The pushforward of $T : \mathcal{X} \to \mathcal{Y}$ on measure $\alpha \in \mathscr{P}(\mathcal{X})$ is denoted $T_\sharp \alpha$ defined by $T_\sharp \alpha(Y) = \alpha(T^{-1}(Y))$ for $Y \subset \mathcal{Y}$.

2.1.1 Monge, Kantorovich, Brenier

The Monge Problem and Transport Maps. For supports $\mathcal{X}$ and $\mathcal{Y}$, consider a pair of measures $\alpha \in \mathscr{P}(\mathcal{X})$, $\beta \in \mathscr{P}(\mathcal{Y})$ and a ground cost function $c : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^+$. Support $\mathcal{X} = \mathcal{Y} = \mathbb{R}^d$ where $d \in \mathbb{N}$, is typically used in practice.

Monge Problem

The Monge optimal transport problem [60] for triplet (α, β, c) is given by:

$$\min_{T:\mathcal{X}\to\mathcal{Y}, T_\sharp\alpha=\beta} \mathcal{L}^c_{\text{Monge}}(T) \qquad \mathcal{L}^c_{\text{Monge}}(T) := \int_{\mathcal{X}} c(x, T(x))\alpha(\mathrm{d}x) \tag{2.1}$$

where $T : \mathcal{X} \to \mathcal{Y}$ is known as the transport map or Monge map.

The Kantorovich Relaxation. The Monge problem assumes existence of a deterministic transport map, which is often not the case for settings such as discrete marginal measures. The Kantorovich relaxation [52] reformulates the optimal transport problem in terms of couplings between the two marginal measures, here α and β.

Kantorovich Formulation

For triplet (α, β, c), the Kantorovich relaxation is given by:

$$\min_{\pi\in\mathcal{S}(\alpha,\beta)} \mathcal{L}^c_{\text{Kant}}(\pi) \qquad \mathcal{L}^c_{\text{Kant}}(\pi) := \int_{\mathcal{X}\times\mathcal{Y}} c(x, y)\pi(\mathrm{d}x, \mathrm{d}y) \tag{2.2}$$

$\mathcal{S}(\alpha, \beta) = \{\pi \in \mathscr{P}(\mathcal{X} \times \mathcal{Y} : \int \pi(\mathrm{d}x, \cdot) = \beta, \int \pi(\cdot, \mathrm{d}y) = \alpha\}$ denotes the collection of couplings between α and β.

The p-norm is a standard ground cost on product space where $\mathcal{X} = \mathcal{Y}$ leading to Wasserstein-p distance, $\mathcal{W}_p$. This has the attractive property, that convergence in Wasserstein-p distance is equivalent to convergence up to the p-moment.

$$\mathcal{W}_p(\alpha, \beta)^p = \min_{\pi\in\mathcal{S}(\alpha,\beta)} \int_{x\in\mathcal{X}, y\in\mathcal{Y}} ||x - y||^p \pi(\mathrm{d}x, \mathrm{d}y) \tag{2.3}$$

Theorem 1 *(Theorem 6.8, [96]) For random variables $X_k \sim \alpha_k$, $Y \sim \beta$, the*

following are equivalent:

$$\mathcal{W}_p(\alpha_k, \beta) \underset{k\to\infty}{\longrightarrow} 0 \qquad\qquad ||\mathbb{E}[X_k^p] - \mathbb{E}[Y^p]||_p \underset{k\to\infty}{\longrightarrow} 0$$

Dual Formulation. The Kantorovich formulation (2.2) admits a dual form which expresses the problem as a maximisation across potentials $f : \mathcal{X} \to \mathbb{R}$, $g : \mathcal{Y} \to \mathbb{R}$. This formulation is often easier to work with, especially in the regularized case.

Dual Formulation

$$\max_{f,g\in\mathcal{R}_c} \mathcal{U}^c(f,g) \qquad \mathcal{U}^c(f,g) := \int_{\mathcal{X}} f(x)\alpha(\mathrm{d}x) + \int_{\mathcal{Y}} g(y)\beta(\mathrm{d}y) \tag{2.4}$$

where $\mathcal{R}_c = \{f \in \mathcal{C}(\mathcal{X}, \mathbb{R}), g \in \mathcal{C}(\mathcal{Y}, \mathbb{R}) : f(x) + g(y) \leq c(x, y)\}$.

Brenier's Theorem. For the setting where $\mathcal{X} = \mathcal{Y} = \mathbb{R}^d$, $c(x, y) = \|x - y\|^2$, and one of the measures, i.e. α is absolutely continuous with respect to the Lebesgue measure, then it was shown by [9] that the optimal coupling π^* in the Kantorovich formulation (2.2) is unique, given by $\pi^* = (\mathbb{I}_d, T)_\sharp \alpha$, where T corresponds to the Monge transport map. Moreover,

- the transport map is given by $T = \nabla\varphi$ for convex function $\varphi : \mathcal{X} \to \mathbb{R}$, known as the Brenier potential,
- Brenier potential, φ, and Kantorovich potential f, are related via identity $\varphi(x) = \frac{\|x\|^2}{2} - f(x)$.

Discrete Optimal Transport. Consider discrete probability measures $\alpha = \sum_{i=1}^N a_i \delta_{x_i}$ and $\beta = \sum_{j=1}^M b_j \delta_{y_j}$ on $\mathcal{X} = \mathbb{R}^{d_x}$ with weights $\mathbf{a} = (a_i)_{i\in[N]}$, $\mathbf{b} = (b_j)_{j\in[M]}$, and atoms $\mathbf{x} = (x_i)_{i\in[N]}$, $\mathbf{y} = (y_j)_{j\in[M]}$. In this case, the squared 2-Wasserstein metric between α and β is given by

Discrete Primal OT

$$\min_{\mathbf{P}\in\mathcal{S}(\mathbf{a},\mathbf{b})} \mathcal{L}^c_{\text{Kant}}(\mathbf{P}) \qquad \mathcal{L}^c_{\text{Kant}}(\mathbf{P}) = \textstyle\sum_{i=1}^N \sum_{j=1}^M c_{i,j} p_{i,j}, \tag{2.5}$$

$\mathbf{P} = (p_{i,j})_{i,j} \in \mathcal{S}(\mathbf{a}, \mathbf{b}) := \{\mathbf{P} \in [0, 1]^{N\times M} | \ \sum_{j=1}^M p_{i,j} = a_i, \ \sum_{i=1}^N p_{i,j} = b_j\}$.

The coupling matrix $\mathbf{P}$ relates to coupling $\pi \in \mathcal{S}(\alpha, \beta)$ through $\pi(\mathrm{d}x, \mathrm{d}y) = \sum_{i,j}\ p_{i,j}\delta_{x_i}(\mathrm{d}x)\delta_{y_j}(y)$. Any element $\pi \in \mathcal{S}(\alpha, \beta)$ allows us to "transport" β to α (and vice-versa), i.e.

$$\pi(\mathrm{d}x|y) = \textstyle\sum_{i,j}\ b_j^{-1} p_{i,j}\delta_{y_j}(y)\delta_{x_i}(\mathrm{d}x). \tag{2.6}$$

The minimization problem (2.5) may be solved through linear programming at computational complexity $\mathcal{O}(N^3 \log N)$ [4], though note that there is not necessarily a unique minimizing argument.

The discrete dual formulation is given by:

Discrete Dual OT

$$\max_{\mathbf{f},\mathbf{g}\in\mathcal{R}(C)} \mathcal{U}^c(\mathbf{f}, \mathbf{g}) \qquad\qquad \mathcal{U}^c(\mathbf{f}, \mathbf{g}) = \mathbf{a}^T\mathbf{f} + \mathbf{b}^T\mathbf{g}, \tag{2.7}$$

where $\mathcal{R}(\mathbf{C}) = \{\mathbf{f}, \mathbf{g} \in \mathbb{R}^N | f_i + g_j \leq c_{i,j}, i, j \in [N]\}$, $\mathbf{f} = (f_i)$, $\mathbf{g} = (g_i)$, and $\mathbf{C} = (c_{i,j})$.

The potentials $\mathbf{f}$ and $\mathbf{g}$ are related to the general case by evaluating at the support points, $f_i = f(x_i)$, $g_j = g(y_j)$.

2.1.2 Entropic Regularization

Entropic OT is a variant of optimal transport, with the addition of an entropic regularization term to the primal objective 2.2, as shown in 2.8. There are many benefits of this penalty term, in particular the resulting coupling is more stable and cheaper to compute than the non-regularized coupling. The solution of the entropic OT problem may be obtained via a simple alternative minimization scheme known as iterative proportional fitting [38], which for the discrete setting is known as the Sinkhorn algorithm [20, 81]. Moreover, the solution to the entropic OT problem converges to the non-regularized OT solution as the regularization term as $\varepsilon \to 0$ [98].

Entropic Primal Formulation

$$\min_{\pi \in \mathcal{S}(\alpha,\beta)} \mathcal{L}_\epsilon^c(\pi) \qquad \mathcal{L}_\epsilon^c(\pi) := \mathcal{L}_{\text{Kant}}^c(\pi) - \varepsilon R(\pi) \tag{2.8}$$

where R is an entropic regularizer, such as entropy $R(\pi) := H(\pi)$ or negative mutual information $R(\pi) := -\text{KL}(\pi||\alpha \otimes \beta)$. Recall $\mathcal{L}_{\text{Kant}}^c(\pi) = \mathbb{E}_\pi[c(\mathbf{X}, \mathbf{Y})]$ and $\text{KL}(\pi||\alpha \otimes \beta) := \int_{\mathcal{X}\times\mathcal{Y}} \log \frac{\mathrm{d}\pi}{\mathrm{d}\alpha\otimes\beta}(x, y)\pi(\mathrm{d}x, \mathrm{d}y)$, $H(\pi) := \int_{\mathcal{X}\times\mathcal{Y}} \log \mathrm{d}\pi(x, y)\pi(\mathrm{d}x, \mathrm{d}y)$.

Alternative regularization terms, $R(\cdot)$, have been used in the literature, [40], including quadratic instead of entropic and variations of the entropic regularization term presented here [19, 20, 36, 66], see Table 2.1.

The corresponding dual formulation for $R(\pi) = -\text{KL}(\pi\|\alpha \otimes \beta)$ may be written:

Entropic Dual Formulation

$$\max_{f\in\mathcal{C}(\mathcal{X},\mathbb{R}),g\in\mathcal{C}(\mathcal{Y},\mathbb{R})} \mathcal{U}_\epsilon^c(f, g) \quad \mathcal{U}_\epsilon^c(f, g) := \mathcal{U}^c(f, g) - \varepsilon \int_{\mathcal{X}\times\mathcal{Y}} e^{\frac{f(x)+g(y)-c(x,y)}{\varepsilon}} \alpha(\mathrm{d}x)\beta(\mathrm{d}y) \tag{2.9}$$

Notice how the dual potential functions f and g are no longer constrained by $f(x) + g(y) \leq c(x, y)$. This is because the optimal solution to the entropically regularized dual problem satisfies this constraint by construction. The primal-dual relationship for primal (2.8) and dual (2.9) between optimal coupling π^* and potentials $\mathbf{f}^*, \mathbf{g}^*$ is given by [40, Proposition 7]:

$$\pi^*(\mathrm{d}x, \mathrm{d}y) = e^{\frac{f^*(x)+g^*(y)-c(x,y)}{\epsilon}} \alpha(\mathrm{d}x)\beta(\mathrm{d}y), \tag{2.10}$$

hence given π^* is a probability measure $f^*(x) + g^*(y) \leq c(x, y)$ a.s. This makes the entropic dual much easier to work with in the sense of optimization.

Discrete Regularized OT. The discrete regularized OT problem is of particular interest in this work given the prevalence of empirical measures in machine learning, but also given that it is a convex problem that can be solved efficiently.

Discrcete Entropic Primal and Dual Formulations

$$\min_{\mathbf{P}\in\mathcal{S}(\mathbf{a},\mathbf{b})} \mathcal{L}_\epsilon^c(\mathbf{P}) \qquad \mathcal{L}_\epsilon^c(\mathbf{P}) = \sum_{i=1}^N \sum_{j=1}^M c_{i,j}p_{i,j} - \varepsilon R(\mathbf{P}), \tag{2.11}$$

$$\max_{\mathbf{f},\mathbf{g}\in\mathbb{R}^N\times\mathbb{R}^M} \mathcal{U}_\epsilon^c(\mathbf{f},\mathbf{g}) \qquad \mathcal{U}_\epsilon^c(\mathbf{f},\mathbf{g}) = \mathbf{a}^T\mathbf{f} + \mathbf{b}^T\mathbf{g} - \varepsilon R^*(\mathbf{f},\mathbf{g},\varepsilon) \tag{2.12}$$

There are many equivalent choices of regularizer $R(\cdot)$ described in the literature with corresponding R^*; parameterization of $\mathbf{f}, \mathbf{g}$, and identity between coupling matrix $\mathbf{P} =: P(\mathbf{f}, \mathbf{g}, \varepsilon)$. Some of the common formulations are given in Table 2.1. Given regularization R, the dual term R^* and coupling matrix, $P(\mathbf{f}, \mathbf{g}, \varepsilon)$ are derived simply via the Langrangian multiplier method.

Table 2.1: Variations of regularization terms

Formulation	R	$[P(\mathbf{f},\mathbf{g},\varepsilon)]_{i,j}$	R^*
1	$-\sum_{i,j} p_{i,j} \log p_{i,j}$	$e^{\frac{f_i+g_j-c_{i,j}}{\varepsilon}-1}$	$\sum_{i,j} e^{\frac{f_i+g_j-c_{i,j}}{\varepsilon}-1}$
2	$-\sum_{i,j} p_{i,j}(\log p_{i,j} - 1)$	$e^{\frac{f_i+g_j-c_{i,j}}{\varepsilon}}$	$\sum_{i,j} e^{\frac{f_i+g_j-c_{i,j}}{\varepsilon}}$
3	$-\sum_{i,j} p_{i,j}(\log \frac{p_{i,j}}{a_i b_j})$	$a_i b_j e^{\frac{f_i+g_j-c_{i,j}}{\varepsilon}-1}$	$\sum_{i,j} a_i b_j e^{\frac{f_i+g_j-c_{i,j}}{\varepsilon}-1}$
4	$-\sum_{i,j} p_{i,j}(\log \frac{p_{i,j}}{a_i b_j})$	$a_i b_j e^{\frac{f_i+g_j-c_{i,j}}{\varepsilon}}$	$\sum_{i,j} a_i b_j \left[e^{\frac{f_i+g_j-c_{i,j}}{\varepsilon}} - 1\right]$

With reference to Table 2.1, Formulation 1 is presented in [20]; Formulation 2 in [66, 91] and in **Chapter 4**; Formulation 3 (primal) and 4 (dual) in [19, 36] and **Chapter 3**. The slight variations are a source of confusion but notice all formulations provide the same optimal $\mathbf{P}$ or $(\mathbf{f}, \mathbf{g})$ in (2.11) and (2.12), given that the different R terms vary only by constants $\log(a_i, b_j)$ or $\sum_{i,j} p_{i,j} = 1$, hence result in the same optimal couplings. Reparameterizing $f_i \leftarrow f_i - \frac{\epsilon}{2}$, $g_j \leftarrow g_j - \frac{\epsilon}{2}$ in the R^* terms translates from Formulation 3 to 4.

2.1.3 The Sinkhorn Algorithm

The Sinkhorn algorithm [20, 66, 81] provides an efficient, GPU-friendly and differentiable approach to solving the discrete, regularized OT problem (2.11), (2.12). This procedure has many interpretations. For what follows, denote the Gibbs kernel $K = (e^{-\frac{c_{i,j}}{\varepsilon}})_{i,j}$, and cost-matrix $C = (c_{i,j})_{i,j}$.

Block gradient ascent. For Formulation 2 in Table 2.1, $\mathcal{U}_\epsilon^c = \mathbf{a}^T\mathbf{f} + \mathbf{b}^T\mathbf{g} - \varepsilon e^{\frac{\mathbf{f}}{\varepsilon}} \odot K e^{\frac{\mathbf{g}}{\varepsilon}}$. First order conditions read:

$$\nabla_{\mathbf{f}}\mathcal{U}_\epsilon^c(\mathbf{f},\mathbf{g}) = \mathbf{a} - e^{\frac{\mathbf{f}}{\varepsilon}} \odot K e^{\frac{\mathbf{g}}{\varepsilon}} = \mathbf{0} \qquad \nabla_{\mathbf{g}}\mathcal{U}_\epsilon^c(\mathbf{f},\mathbf{g}) = \mathbf{b} - e^{\frac{\mathbf{g}}{\varepsilon}} \odot K^T e^{\frac{\mathbf{f}}{\varepsilon}} = \mathbf{0} \tag{2.13}$$

Rearranging and solving for $\mathbf{f}$ and $\mathbf{g}$ respectively gives block updates in $\mathbf{f}, \mathbf{g}$, for $l \geq 0$:

$$\mathbf{f}^{l+1} = \varepsilon(\log \mathbf{a} - \log K e^{\frac{\mathbf{g}^l}{\varepsilon}}) \qquad \mathbf{g}^{l+1} = \varepsilon(\log \mathbf{b} - \log K^T e^{\frac{\mathbf{f}^{l+1}}{\varepsilon}}) \tag{2.14}$$

for some initialization of $\mathbf{f}, \mathbf{g}$, discussed in [91]. Applying these updates corresponds to a block gradient ascent to the concave dual formulation, hence solves the optimization problem.

Note that these iterative updates will differ slightly depending on the regularization term - the Sinkhorn algorithm as written in **Chapter 3** can be derived in a similar way using regularization Formulation 4.

Iterative marginal projections. Again using Formulation 2, recall the primal dual relationship, $p_{i,j} = e^{\frac{f_i + g_j - c_{i,j}}{\epsilon}}$. Denote $\mathbf{u} = e^{\frac{\mathbf{f}}{\varepsilon}}$, $\mathbf{v} = e^{\frac{\mathbf{g}}{\varepsilon}}$ and hence $\mathbf{P} = \text{diag}(\mathbf{u}) K \text{diag}(\mathbf{v})$, where $\text{diag}(\mathbf{u})$ is the matrix with diagonal entries $\mathbf{u}$. The marginal constraints for $\mathbf{P} \in \mathcal{S}(\mathbf{a}, \mathbf{b})$ can then be rewritten as $\mathbf{u} \odot\ K\mathbf{v} = \mathbf{a}$, $\mathbf{v} \odot\ K^T\mathbf{u} = \mathbf{b}$. Rearranging, gives block updates in $\mathbf{u}, \mathbf{v}$, for $l \geq 0$:

$$\mathbf{u}^{l+1} = \frac{\mathbf{a}}{K\mathbf{v}^l} \qquad\qquad \mathbf{v}^{l+1} = \frac{\mathbf{b}}{K^T\mathbf{u}^{l+1}} \tag{2.15}$$

where again the initialization of $\mathbf{u}, \mathbf{v}$ through $\mathbf{f}, \mathbf{g}$ is discussed in [91]. Note that this substitution of $\mathbf{u} = e^{\frac{\mathbf{f}}{\varepsilon}}$, $\mathbf{v} = e^{\frac{\mathbf{g}}{\varepsilon}}$ and rearrangement recovers (2.14).

Iterative KL projections. As per [66], define $Proj_A^{\text{KL}}(Q) = \arg\min_{P \in A} \text{KL}(P||Q)$ as the projection of measures $Q \in \mathscr{P}(\mathcal{X})$ to subset $A \subset \mathscr{P}(\mathcal{X})$. Let $C_a = \{P | \sum_i p_{i,j} = a_i\}$ and $C_b = \{P|\ \sum_j p_{i,j} = b_j\}$. Hence the collection of coupling matrices is $\mathcal{S}(\mathbf{a}, \mathbf{b}) = C_a \cap C_b$.

Under regularization Formulation 3 or 4, $\mathcal{L}_\epsilon^c(\mathbf{P}) = \langle C, \mathbf{P}\rangle + \varepsilon \text{KL}(\mathbf{P}||\alpha \otimes \beta)$. Under rearrangement $\min_{\mathbf{P}} \mathcal{L}_\epsilon^c(\mathbf{P}) = \min_{\mathbf{P}} \text{KL}(\mathbf{P}||K)$, hence the optimal coupling matrix

solving the regularized OT problem (2.11) solves: $\mathbf{P}^* = Proj^{\mathrm{KL}}_{U(\alpha,\beta)}(K)$, which may be obtained via iterative KL projections, also known as Bregman projections [8]

$$\mathbf{P}^{l+2} = Proj^{\mathrm{KL}}_{C_b}(\mathbf{P}^{l+1}) \qquad \mathbf{P}^{l+1} = Proj^{\mathrm{KL}}_{C_a}(\mathbf{P}^{l}). \tag{2.16}$$

2.1.4 The Schrödinger Bridge Problem

Consider a reference stochastic process with indexed densities $(p_t)_{t\in[0,T]}$. The Schrödinger bridge problem [74] entails finding a stochastic process, $(\pi_t)_{t\in[0,T]}$, 'close' to this reference stochastic process in terms of path-space Kullback–Leibler divergence, but with the addition of constrained terminal marginals, $\pi_0 = \alpha$, $\pi_T = \beta$. More formally:

Dynamic Schrödinger Bridge Problem

$$\underset{\pi\in\mathcal{S}(\alpha,\beta,[0,T])}{\operatorname{argmin}} \ \mathrm{KL}(\pi||p) \tag{2.17}$$

where $\mathcal{S}(\alpha,\beta,[0,T]) := \{\pi \in \mathscr{P}(\mathcal{X}^{[0,T]})|\pi_0 = \alpha, \pi_T = \beta\}$

Optimal Transport. A closely related problem is the static Schrödinger bridge problem, which aims to minimize the divergence only at the terminal marginals:

Static Schrödinger Bridge Problem

$$\underset{\pi^S\in\mathcal{S}(\alpha,\beta)}{\operatorname{argmin}} \mathrm{KL}(\pi^S||p_{0,T}) \tag{2.18}$$

where $\mathcal{S}(\alpha,\beta) := \{\pi \in \mathscr{P}(\mathcal{X}\times\mathcal{X})|\pi_0 = \alpha, \pi_T = \beta\}$

Given $\mathrm{KL}(\pi||p) = \mathrm{KL}(\pi_{0,T}|p_{0,T}) + \mathbb{E}_{\pi_{0,T}}[\mathrm{KL}(\pi_{|0,T}||p_{|0,T})]$, the optimal path for the dynamic problem π^* marginalizes to the optimal static coupling $\pi^{S,*}$:

$$\pi^*(x_{[0,T]}) = \pi^{S,*}(x_0, x_T)\pi^*(x_{(0,T)})$$

where $x_{(0,T)} = \{x_t|t \in (0,T)\} = x_{[0,T]} \setminus \{x_0, x_T\}$. By expansion:

$$\mathrm{KL}(\pi^S||p_{0,T}) = \int \pi^S(x_0,x_T)[\log \pi^S(x_0,x_T) - \log p_{0,T}(x_0,x_T)]\mathrm{d}x_0,\mathrm{d}x_T \tag{2.19}$$

$$= \mathbb{E}_{\pi^S}[-\log p_{0,T}] - H(\pi^S) \tag{2.20}$$

where $H(\pi_{0,T}) := -\int \pi^S(x_0,x_T)\log\pi^S(x_0,x_T)\mathrm{d}x_0,\mathrm{d}x_T$. Given fixed marginals, we have the following equivalence under $\arg\min$:

$$\underset{\pi^S\in\mathcal{S}(\alpha,\beta)}{\operatorname{argmin}} \mathbb{E}_{\pi^S}[-\log p_{0,T}] - H(\pi^S) = \underset{\pi^S\in\mathcal{S}(\alpha,\beta)}{\operatorname{argmin}} \mathbb{E}_{\pi^S}[-\log p_{T|0}] - H(\pi^S)$$

Therefore the static Schrödinger bridge problem is equivalent to the regularized optimal transport problem with ground cost $c(x, y) = -\log p_{T|0}(y|x)$ and regularization term $\epsilon = 1$:

$$\underset{\pi^S \in \mathcal{S}(\alpha,\beta)}{\operatorname{argmin}} \mathrm{KL}(\pi^S || p_{0,T}) = \underset{\pi^S \in \mathcal{S}(\alpha,\beta)}{\operatorname{argmin}} \mathbb{E}_{\pi^S} - \log p(\mathbf{X}_T|\mathbf{X}_0) - H(\pi^s).$$

For the Brownian motion reference process $-\log p_{T|0}(y|x) = \frac{\|x-y\|^2}{2\sigma^2}$, as $\mathbf{X}_T|x_0 \sim \mathcal{N}(x_0, \sigma^2\mathbb{I})$. This gives the standard squared-Euclidean ground cost with regularization $\varepsilon = 2\sigma^2$:

$$\underset{\pi^S \in \mathcal{S}(\alpha,\beta)}{\operatorname{argmin}} \mathrm{KL}(\pi^S || p_{0,T}) = \underset{\pi^S \in \mathcal{S}(\alpha,\beta)}{\operatorname{argmin}} \mathbb{E}_{\pi^S} \|\mathbf{X}_T - \mathbf{X}_0\|^2 - 2\sigma^2 H(\pi^s).$$

Iterative Proportional Fitting. Let $\pi_0 = p$, then for $n \geq 0$, the iterative proportional fitting (IPF) procedure [38] is given by:

$$\pi^{2n+1} = \arg\min_{\pi} \{\mathrm{KL}(\pi||\pi^{2n}), \quad \pi_T = \beta\} \tag{2.21}$$

$$\pi^{2n+2} = \arg\min_{\pi} \{\mathrm{KL}(\pi||\pi^{2n+1}), \quad \pi_0 = \alpha\}. \tag{2.22}$$

This procedure converges to the solution of the Schrödinger bridge problem [74]. By comparison to (2.16), it can be seen that the Sinkhorn algorithm is a particular case of IPF procedure for discrete measures, where the KL minimization is over doubly stochastic matrices rather than continuous time stochastic processes.

2.2 Diffusion Models: From Score-Matching to the Schrödinger Bridge

2.2.1 Deep Probabilistic Modeling via Time Reversal

Diffusion models, also known as score-based generative models, were first pioneered by [47, 82, 86, 88] and have since emerged as state of the art deep probabilistic models for a number of tasks. Diffusion models are first and foremost generative models and have been applied across modalities: image [24], video [5, 39, 79], audio [58], manifold valued data [23, 93]. Other applications of diffusion models include: classification [44]; inverse problems (e.g. [59]); in biology (e.g. [102]); density estimation [85], trajectory inference [7] - the list continues to grow.

As shown in this thesis and contributing papers [7, 93], diffusion models also have deep connections to OT. This connection both offers novel insights and methodology for generative modeling tasks; and secondly enables one to approximate high-dimensional optimal transport.

Diffusion generative models consist of three primary components:

1. An iterative forward process, initialized from a data distribution and terminating 'close' to a 'simple' distribution.
2. An easy to sample distribution 'close' to the terminal distribution of the forward process. Often referred to as a *prior*.
3. A learnt, iterative, backward process, initialized from the prior. The backward process is constructed to terminate 'close' to the given data distribution.

A core characteristic of diffusion models is in their highly scalable training procedure. Although sampling the backward process typically consists of many steps in an iterative process, the training procedure does not require taking gradients through the sampling iterations. But instead only a single step. Informally, the diffusion model training procedure consists of decomposing the generative modeling problem

into infinitely many small problems, each corresponding to a single time point, t; and then solving these problems jointly.

Time reversal. Denote the forward noising process by $(\mathbf{X}_t)_{t=0}^T$ with the following dynamics:

$$\mathrm{d}\mathbf{X}_t = f(\mathbf{X}_t, t)\mathrm{d}t + g(\mathbf{X}_t, t)\mathrm{d}\mathbf{B}_t, \quad \mathbf{X}_0 \sim p_0 = p_{\text{data}}, \tag{2.23}$$

where $(\mathbf{B}_t)_{t\in[0,T]}$ is a Brownian motion and $f : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$ is regular enough so that (strong) solutions exist.

Under conditions on f, it is well-known (see [14, 37, 45] for instance) that the reverse-time process $(\mathbf{Y}_t)_{t\in[0,T]} = (\mathbf{X}_{T-t})_{t\in[0,T]}$ satisfies

$$\mathrm{d}\mathbf{Y}_t = \left[-f(\mathbf{Y}_t, t) + gg^T(\mathbf{Y}_t, t)\nabla \log p_{T-t}(\mathbf{Y}_t) + \nabla_{\mathbf{Y}} \cdot gg^T(\mathbf{Y}_t, t)\right]\mathrm{d}t + g(\mathbf{Y}_t, t)\mathrm{d}\tilde{\mathbf{B}}_t \tag{2.24}$$

with initialization $\mathbf{Y}_0 \sim p_T$, where p_t denotes the marginal density of $\mathbf{X}_t$ and $\tilde{\mathbf{B}}$ denotes another Brownian motion, independent to the forward process.

Typically, the diffusion term g is chosen to be a function of only t, independent of state-dimension, which simplifies the time-reversal to:

$$\mathrm{d}\mathbf{Y}_t = \left[-f(\mathbf{Y}_t, t) + g^2(t)\nabla \log p_{T-t}(\mathbf{Y}_t)\right]\mathrm{d}t + g(t)\mathrm{d}\tilde{\mathbf{B}}_t \tag{2.25}$$

If one had access to the reverse diffusion terms, then one could simply simulate the diffusion in order to generate data, such as through (2.29). The score term, $\nabla \log p_{T-t}$ and hence reverse diffusion, is typically intractable however, so instead must often be learnt.

Learning the time-reversal. The intractable score may be expressed as the expectation over the tractable conditional density using score-matching [50].

$$\begin{aligned}
\nabla_{x_t} \log p(x_t) &= \frac{\nabla_{x_t} p(x_t)}{p(x_t)} = \frac{\nabla_{x_t} \int p(x_t, x_s)\mathrm{d}x_s}{p(x_t)} = \frac{\int \nabla_{x_t} p(x_t, x_s)\mathrm{d}x_s}{p(x_t)} \\
&= \frac{\int p(x_t, x_s)\nabla_x \log p(x_t, x_s)\mathrm{d}x_s}{p(x_t)} = \int p(x_s|x_t)\nabla_{x_t} \log p(x_t|x_s)\mathrm{d}x_s \\
&= \mathbb{E}_{s|t}[\nabla_{x_t} \log p(x_t|\mathbf{X}_s)]
\end{aligned} \tag{2.26}$$

here the first step uses the gradient of a logarithm, the next uses marginalization of an augmented density where swapping integral and gradient is possible given the density is finite. The middle line step uses derivative: $\nabla_{x_t} p(x_t, x_0) = p(x_t, x_s) \nabla_{x_t} \log p(x_t, x_s)$ and the final steps use Bayes' rule and a simplification.

By definition, the conditional expected value is the minimizer of the L2 distance:

$$\nabla \log p_t(x_t) = \arg\min_r \mathbb{E}_{p_{s,t}}[\|r - \nabla_{x_t} \log p_{t|s}(x_t|\mathbf{X}_s)\|^2]. \tag{2.27}$$

The above learns a single score-term at time t, instead, one can learn the score for all $t \in [0, T]$ jointly using a neural network $s_{\theta^*}(x_t, t) \approx \nabla \log p_t(x_t)$:

$$\theta^* = \arg\min_\theta \mathbb{E}_{p_{s,t}}[\|s_\theta(\mathbf{X}_t, t) - \nabla_{x_t} \log p_{t|s}(\mathbf{X}_t|\mathbf{X}_s)\|^2]. \tag{2.28}$$

For SDEs with affine drift, one may simply choose $s = 0$ as $\nabla_{x_t} \log p_{t|0}(\mathbf{X}_t|x_0)$ is analytically tractable. Additionally, for affine SDEs one can simulate $\mathbf{X}_t \sim p_{t|0}(\mathbf{X}_t|x_0)$ in closed-form without approximation. This means the diffusion model may be trained in a scalable manner.

Diffusion Model Training Procedure

1. Sample data, $x_0 \sim p_{\text{data}}$
2. Sample time, $t \sim \text{Uniform}([0, T])$
3. Sample forward SDE, $x_t \sim p_t(\cdot|x_0)$
4. Perform a gradient descent step e.g. on loss from (2.28).

Note: other losses and parameterizations also exist.

Simulation is typically required to approximate (2.28) for more complex forward diffusions, achieved through some discretization scheme. If one partitions the time domain $[0, T]$ by $0 = t_0 < t_1 < t_2 < \ldots < t_N = T$, then one may simulate $\mathbf{X}_{t_n}, \mathbf{X}_{t_{n-1}}$, using e.g. Euler Maruyama, then evaluate the single step conditional score term $\nabla \log p_{t_n|t_{n-1}}(\mathbf{X}_{t_n}|\mathbf{X}_{t_{n-1}})$, which is often available even if $\nabla_{x_t} \log p_{t|0}(\mathbf{X}_t|\mathbf{X}_0)$ is not.

Alternatively, one may use implicit score-matching (ISM) [50, 87] which does not require knowing $\nabla \log p_{t_n|t_{n-1}}(\mathbf{X}_{t_n}|\mathbf{X}_{t_{n-1}})$:

$$\nabla \log p_t = \arg\min_{s_\theta} \mathbb{E}_{p_t}\left[\frac{1}{2}\|s_\theta(x_t, t)\|^2 + \text{div}_{x_t}(s_\theta)\right].$$

Here div_x is the divergence operator, for high dimensional settings this may be estimated via Hutchinson's trace estimator [49]. A derivation and extension to drift-matching on manifolds is given in **Chapter 6**.

Sampling from the diffusion process. There are many way to sample from a diffusion process, arguably the most simple is via a time-discretization and the Euler-Maruyama scheme. This is general to both simulating the forward process for the ISM loss or sampling from the backward process to generate new samples.

Denote the backward drift term as $b_\theta(\mathbf{Y}_t, t) := -f(\mathbf{Y}_t, t) + g^2(t)\nabla s_\theta(\mathbf{Y}_t, t)$, where $s_\theta(\mathbf{Y}_t, t)$, the generative diffusion may then be written $\mathrm{d}\mathbf{Y}_t = b_\theta(\mathbf{Y}_t, t)\mathrm{d}t + g(t)\mathrm{d}\tilde{\mathbf{B}}_t$. For step-schedule $(\gamma_t)_t$, typically annealed toward 0, simulating the reverse diffusion corresponds to the following stepwise update:

$$\mathbf{Y}_{t+1} = \mathbf{Y}_t + \gamma_t b_\theta(\mathbf{Y}_t, t) + g(t)\sqrt{\gamma_t}\epsilon_t \qquad \epsilon_t \sim \mathcal{N}(\mathbf{0}, \mathbb{I}), \tag{2.29}$$

where recall $\mathbf{Y}_t$ has the reverse time index to $\mathbf{X}_t$.

The sampling procedure is often decomposed into 'predictor' and 'corrector' steps [88]. The predictor step corresponds to sampling from $p_{t-1|t}$, typically using the reverse-time SDE described above. The corrector step corresponds to sampling from a Markov chain converging to p_t, which may be approximated by Langevin dynamics:

$$\mathbf{Y}_t^{k+1} = \mathbf{Y}_t^k + \tilde{\gamma}_k s_\theta(\mathbf{Y}_t^k, t) + \sqrt{2\tilde{\gamma}_k}\epsilon_k \qquad \epsilon_k \sim \mathcal{N}(\mathbf{0}, \mathbb{I}). \tag{2.30}$$

where superscript k indexes the Langevin dynamics Markov chain with step sizes $(\tilde{\gamma}_k)_k$, and subscript t is the fixed time index of the backward SDE.

Langevin Dynamics vs Stochastic Gradient Langevin Dynamics

Many recent papers and popular blog posts [2, 26, 42, 100] confuse the stochasticity introduced in Langevin dynamics with the stochasticity from the stochastic gradient in Stochastic Gradient Langevin Dynamics (SGLD) [99]. Langevin dynamics is the broader Monte Carlo method for approximate sampling from a distribution $p(x)$ using

$$\mathbf{X}^{k+1} = \mathbf{X}^k + \tilde{\gamma}_k \nabla \log p(\mathbf{X}^k) + \sqrt{2\tilde{\gamma}_k}\epsilon_k \qquad \epsilon_k \sim \mathcal{N}(\mathbf{0}, \mathbb{I}). \tag{2.31}$$

SGLD is a particular case of Langevin dynamics targeting the posterior $p(x|\mathbf{y}) = p(x)\prod_{i=1}^N p(y_i|x)$, but where the the likelihood gradient terms are subsampled with minibatch size $n < N$, indexed by $(i_k)_k$:

$$\nabla \log p(x|\mathbf{y}) = \nabla \log p(x) + \sum_{i=1}^{N} \nabla \log p(y_i|x) \tag{2.32}$$

$$\approx \nabla \log p(x) + \frac{N}{n}\sum_{k=1}^{n} \nabla \log p(y_{i_k}|x). \tag{2.33}$$

Here the 'stochastic gradient' comes from minibatching the gradient terms.

In addition to Euler-Maruyama sampler, one may use a number of other sampling schemes for SDE solvers or ODE solvers, as discussed in the next section. Some notable sampling schemes include:, higher-order SDE samplers [27], adaptive step sizes [51], or for models with linear forward process, one may discretize and reparameterize the sampling scheme to iteratively predict x_0 [47, 84].

Choice of forward process. The field of diffusion models has rapidly expanded to encompass various forward processes beyond the original linear SDEs on finite-dimensional Euclidean space [47, 86, 88]. Extensions include: to Riemannian manifold constrained diffusions [23, 93]; non-linear neural network parameterized forward processes [7, 93], higher-order diffusions [28, 63], discrete state [12], infinite-dimensional [6, 55]; heat dissipation or blurring [2, 48, 69]. Although not technically diffusions, the basic forward-backward paradigm has been investigated with various other corruption processes [2, 22, 43, 101]. In particular, [3] provides a general framework for understanding a wide class of time-reversals.

By far the most commonly used forward noising process are linear SDEs, and

in particular scaled Brownian motion or the Ornstein Uhlenbeck process. For scalable training without simulation approximation, one is required to evaluate $\nabla_{x_t} \log p_{t|0}(x_t|x_0)$ and sample from perturbation kernel $p_{t|0}(\cdot|x_0)$ - this is possible with linear SDEs but not typically possible for other SDEs.

Linear Forward Processes. Let the moments of density $p_{t|0}(x_t|x_0)$ be denoted μ_t, Σ_t. Moments μ_t, Σ_t of linear $p_{t|0}(x_t|x_0)$ satisfy ODEs [73] which may then be solved for closed-form sampling.

By appealing to Fokker–Planck–Kolmogorov, [73, Chapter 5] shows that μ_t, Σ_t follow the following ODEs:

$$\begin{aligned} \frac{\mathrm{d}\mu_t}{\mathrm{d}t} &= \mathbb{E}\left[f(\mathbf{X}_t, t)\right] \\ \frac{\mathrm{d}\Sigma_t}{\mathrm{d}t} &= \mathbb{E}\left[f(\mathbf{X}_t, t)(\mathbf{X} - \mu_t)^T\right] + \mathbb{E}\left[(\mathbf{X} - \mu_t) f(\mathbf{X}_t, t)^T\right] + \mathbb{E}\left[g(\mathbf{X}_t, t) g(\mathbf{X}_t, t)^T\right]. \end{aligned} \tag{2.34}$$

For linear forward SDE let $\mathbf{F}(t)\mathbf{X}_t := f(\mathbf{X}_t, t)$, and diffusion matrix as $\mathbf{G}$, hence

$$\mathrm{d}\mathbf{X}_t = \mathbf{F}(t)\mathbf{X}_t \mathrm{d}t + \mathbf{G}(t)\mathrm{d}\mathbf{B}_t, \quad \mathbf{X}_0 \sim p_0 = p_{\text{data}}, \tag{2.35}$$

The distribution for density $p_{t|0}(x_t|x_0)$ will be Gaussian for linear forward SDEs hence fully specified by moments μ_t, Σ_t. The equations (2.34) may be rewritten for linear SDEs as:

$$\begin{aligned} \frac{\mathrm{d}\mu_t}{\mathrm{d}t} &= \mathbf{F}(t)\mu_t \\ \frac{\mathrm{d}\Sigma_t}{\mathrm{d}t} &= \mathbf{F}(t)\Sigma_t + \Sigma_t \mathbf{F}^T(t) + \mathbf{G}(t)\mathbf{G}^T(t). \end{aligned} \tag{2.36}$$

Although the solution to the Brownian motion SDE is trivial, the solutions to the ODEs in (2.36) may be used to compute the perturbation kernel for other linear SDE including higher order diffusions [28] and multivariate, time-inhomogeneous Ornstein Uhlenbeck processes with non-zero terminal mean and non-unit terminal variance, see below. Note that the Ornstein Uhlenbeck process is used heavily in discrete diffusions such as [47] and connections are given in Section 2.2.2.

The solution of (2.36) may be found by appropriate substitution verification or the integrating factor approach for simple linear SDEs. A more general approach, similar to a multivariate integrating factor, uses matrix exponentials, again detailed in [73, Chapter 6]. Mean μ_t may be evaluated as below in (2.37)

$$\mu_t = \exp\left[\int \mathbf{F}(t)\mathrm{d}t\right] x_0, \tag{2.37}$$

where the integral in the exponential is element-wide. For time homogeneous $\mathbf{F}(t) = \mathbf{F}$ the integral $\int \mathbf{F}(t)\mathrm{d}t$ is trivial. More commonly $\mathbf{F}(t) = \beta(t)\mathbf{F}$ for integrable time-scaling function $\beta(t)$[1], the solution is also simple to compute.

The covariance matrix Σ_t may be computed using filtering techniques, see e.g. [73, Chapter 6]. One may construct matrices $\mathbf{C}_t, \mathbf{D}_t$, such that $\Sigma_t = \mathbf{C}_t\mathbf{D}_t^{-1}$, with dynamics following ODEs

$$\begin{bmatrix} \frac{\mathrm{d}\mathbf{C}_t}{\mathrm{d}t} \\ \frac{\mathrm{d}\mathbf{D}_t}{\mathrm{d}t} \end{bmatrix} = \begin{bmatrix} \mathbf{F}(t) & \mathbf{G}(t)\mathbf{G}^T(t) \\ \mathbf{0} & -\mathbf{F}^T(t) \end{bmatrix} \begin{bmatrix} \mathbf{C}_t \\ \mathbf{D}_t \end{bmatrix} \tag{2.38}$$

with initialization $\mathbf{C}_0 = \Sigma_0$, $\mathbf{D}_0 = \mathbb{I}_d$, where Σ_0 typically has entries all 0 in our setting, for fixed $\mathbf{X}_0 = x_0$.

Again, (2.38) may be solved similar to as in (2.37) with matrix exponential

$$\begin{bmatrix} \mathbf{C}_t \\ \mathbf{D}_t \end{bmatrix} = \exp\left[\int \begin{bmatrix} \mathbf{F}(t) & \mathbf{G}(t)\mathbf{G}^T(t) \\ \mathbf{0} & -\mathbf{F}^T(t) \end{bmatrix} \mathrm{d}t\right] \begin{bmatrix} \mathbf{C}_0 \\ \mathbf{D}_0 \end{bmatrix} \tag{2.39}$$

where again the integral in the exponential of (2.39) is element-wise.

As far as I am aware, this very practical approach has not been utilized in the diffusion model literature other than in [80], whereas other complex higher-order SDEs are solved via lengthy algebraic manipulation and substitution verification [28]. For practical purposes, many software packages facilitate matrix exponential computations or close approximations, some software packages even provide symbolic matrix exponential solutions.

Time inhomogeneous Ornstein Uhlenbeck Process

If $\mathrm{d}x = -\frac{\beta(t)}{2}\left(\frac{x-\mu}{\sigma^2}\right)\mathrm{d}t + \sqrt{\beta(t)}\mathrm{d}B_t$ then $\mathbf{X}_t|x_0 \sim \mathcal{N}(\mu_t, \Sigma_t)$ where

$$\mu_t = e^{-\frac{1}{2\sigma^2}\int_0^t \beta(t')\mathrm{d}t'} x_0 + \left(1 - e^{-\frac{1}{2\sigma^2}\int_0^t \beta(t')\mathrm{d}t'}\right)\mu$$
$$\Sigma_t = \sigma^2\left(1 - e^{-\frac{1}{\sigma^2}\int_0^t \beta(t')\mathrm{d}t'}\right)$$

Often in the literature practitioners set $\sigma \leftarrow 1, \mu \leftarrow 0$, and instead scale the data x_0 to lie within a fixed interval such as $[-1, 1]$ or $[-0.5, 0.5]$.

Time scaling. Let time-scaling function be denoted $\beta : [0, T] \to \mathbb{R}^+$ [1]. Under discretization scheme given by (2.29) choosing $\beta(t)$ appropriately with fixed step-size is equivalent to no time-scaling but a corresponding varying step-size. Empirically it has been found useful to decrease $\beta(t) \to 0$ as $t \to 0$, i.e. approaching data. By Ito, time-scaling transforms the SDE as follows:

$$\mathrm{d}\mathbf{X}_t = f(\mathbf{X}_t, t)\mathrm{d}t + g(\mathbf{X}_t, t)\mathrm{d}\mathbf{B}_t \tag{2.40}$$
$$\mathrm{d}\tilde{\mathbf{X}}_t = \beta(t)f(\tilde{\mathbf{X}}_t, t)\mathrm{d}t + \sqrt{\beta(t)}g(\tilde{\mathbf{X}}_t, t)\mathrm{d}\mathbf{B}_t. \tag{2.41}$$

Although lots of empirical and heuristic work has been done to find a suitable $\beta(\cdot)$ [24, 54], as far as I am aware identifying an optimal time-scale function is still an open problem.

VPSDE vs OU, VESDE vs Brownian Motion

The terms variance preserving SDE (VPSDE) and variance exploding SDE (VESDE) were introduced in [88] to refer to time-inhomogeneous Ornstein Uhlenbeck (OU) and Brownian motion (BM) SDEs.

The terms VPSDE and VESDE are commonly used in the literature often without historical acknowledgement of simply being OU and BM processes, and sometimes treated as distinct from OU and BM [10].

[1] To be consistent with the literature, β is used as the time-scaling function, beware however that in other sections of this thesis β is used as a marginal measure in OT.

2.2.2 The Many Faces of Diffusion Models

As discussed in Section 2.2.1, treating diffusion models as the time-reversal of an SDE is a broad and elegant framework. There are many connections, generalizations, and similarities to other well studied generative probabilistic models. Such other models can often be trained in a scalable manner using the diffusion model corruption / iterative refinement training paradigm. A non-exhaustive list is detailed below:

- Time reversal of an SDE
- Variational Markov Chain
- Autoencoder with Fixed Encoder
- Sequential Energy-based Model
- Amortized Langevin Dynamics
- Sequential Denoising Autoencoders
- Continuous-time Normalizing Flow
- Schrödinger Bridge.

Amortized Langevin Dynamics

Before the time-reversal generalization [7, 88], earlier work [86] proposed to learn the score term directly using the implicit score-matching loss [87], or denoising score matching [84] for a sequence of noise levels that converges to 0. The denoising approach is in similar vein to annealing and corresponds to learning the scores $\nabla \log p_t$ for a sequence of t, where p_t is the density of the data distribution convolved with Gaussian noise of variance corresponding to time t.

A naive approach may to approximate $\nabla \log p_0$ with $s_\theta(\cdot, 0)$, then directly apply Langevin dynamics to simulate from the data distribution:

$$x_{k+1} = x_k + \gamma_k s_\theta(x_k, 0) + \sqrt{2\gamma_k}\epsilon_k$$

where $\epsilon_k \sim \mathcal{N}(\mathbf{0}, \mathbb{I})$. Here k indexes the iteration of the Markov chain Monte Carlo procedure with score $\nabla \log p$ targetting p_0. This has practical limitations, in particular it suffers from mode-collapse in generation. In order to sample from multi-modal distributions an annealed-type approach is typically used - one would apply Langevin dynamics for each noise level t for a number of steps, before continuing onto the next lower noise level.

Within the predictor-corrector framework, this approach corresponds to a sequence of 'corrector' steps without the 'predictor' step. Here each step is an iteration of Langevin dynamics where the gradient of the potential is approximated with the neural network parameterized score.

Variational Markov Chain

Closely related to the time-reversal of an SDE is an approach derived by [82] which is then built upon with the addition of scalable training procedure by [47]. The primary difference to reverse-time SDEs is the explicit discretization and a training objective derived from a variational perspective, rather than through reverse SDE. Both approaches lead to equivalent losses however, up to weighting of component terms.

Consider the forward noising process as a discrete Markov chain: $p(x_{0:N}) = p(x_0)\prod_{k=1}^{N-1} p(x_{k+1}|x_k)$, where $p_0 = p_{\text{data}}$ and $x_k|x_{k-1} \sim \mathcal{N}\left(\sqrt{1-\beta_k}x_k,\ \beta_k\mathbb{I}\right)$ for some positive schedule $(\beta_k)_k$. This forward process is a discrete approximation to the Ornstein Uhlenbeck process, hence $p_N \approx \mathcal{N}(\mathbf{0}, \mathbb{I})$ by construction.

> The forward DDPM noising process is a discrete approximation to the Ornstein Uhlenbeck Process
>
> The DDPM forward discretized SDE is applied independently per dimension, so here we just consider the univariate case. Consider the moments of an Ornstein Uhlenbeck (OU) process, let $\mu = 0$, $\sigma = 1$ and piecewise constant β, $\beta(t') = \beta_t$ for $t' \in (t, t+1)$, i.e.$\int_t^{t+1} \beta(t')\mathrm{d}t' = \beta_t$. By a Taylor approximation $\mathbb{E}[\mathbf{X}_{t+1}|x_t] = e^{-\frac{1}{2}\int_t^{t+1}\beta(t')\mathrm{d}t'}x_t = \sqrt{e^{-\beta_t}} \approx \sqrt{1-\beta_t}$ and variance $\mathbb{V}[\mathbf{X}_{t+1}|x_t] = 1 - e^{-\beta_t} \approx \beta_t$. Therefore, $x_t|x_{t-1} \sim \mathcal{N}\left(\sqrt{1-\beta_t}x_t,\ \beta_t\right)$.
>
> Similarly, the closed-form perturbation kernel may be derived from the OU process. $\mathbb{E}[\mathbf{X}_t|x_0] = e^{-\frac{1}{2}\int_0^t\beta(t')\mathrm{d}t'}x_0 = e^{-\frac{1}{2}\sum_{k=1}^{t}\int_k^{k+1}\beta(t')\mathrm{d}t'}x_0 = e^{-\frac{1}{2}\sum_{t'=1}^{t}\beta_{t'}}x_0 = \sqrt{\prod_{t'=1}^{t} e^{-\beta_{t'}}}x_0 \approx \sqrt{\prod_{t'=1}^{t}(1-\beta_t)}x_0 = \sqrt{\bar{\alpha}_t}x_0$ and variance $\mathbb{V}[\mathbf{X}_t|x_0] = 1-\bar{\alpha}_t$ where $\bar{\alpha}_t = \prod_{t'=1}^{t}(1-\beta_t)$.

Now consider a parameterized reverse process $q_\theta(x_{0:N}) = q(x_N)\prod_{k=0}^{N} p_\theta(x_{k-1}|x_k)$, where $q(x_N) = \mathcal{N}(\mathbf{0}, \mathbb{I})$ and $p_\theta(x_{k-1}|x_k) = \mathcal{N}(\mu_\theta(x_k, k), \sigma_k^2\mathbb{I})$.

Learning the generative model may be formulated as a KL divergence minimization,

using $x_N, \ldots, x_1$ as latent variables. As shown in [82]

$$\mathbb{E}[-\log q_\theta(x_0)] \leq \mathbb{E}_p\left[-\log \frac{q_\theta(x_{0:N})}{p(x_{1:N}|x_0)}\right] \tag{2.42}$$

$$= \mathbb{E}_p\left[-\log q(x_N) - \sum_{k\geq 1} \log \frac{q_\theta(x_{k-1}|x_k)}{p(x_k|x_{k-1})}\right] \tag{2.43}$$

$$= \mathbb{E}_p\left[-\log q(x_N) - \sum_{k\geq 1} \log \frac{q_\theta(x_{k-1}|x_k)}{p(x_{k-1}|x_k, x_0)} \cdot \frac{p(x_{k-1}|x_0)}{p(x_k|x_0)}\right] \tag{2.44}$$

$$= \mathbb{E}_p\left[-\log \frac{q(x_N)}{p(x_N|x_0)} - \sum_{k>1} \log \frac{q_\theta(x_{k-1}|x_k)}{p(x_{k-1}|x_k, x_0)} - \log q_\theta(x_0|x_1)\right] \tag{2.45}$$

$$= \mathbb{E}_p\left[\mathrm{KL}(p(x_N|x_0)||q(x_N)) + \sum_{k>1} \mathrm{KL}(p(x_{k-1}|x_k, x_0)||q_\theta(x_{k-1}|x_k)) - \log q_\theta(x_0|x_1)\right] \tag{2.46}$$

The first inequality (2.42) above is derived from standard variational inference methods. The second line (2.43) is computed using the logarithm of $q_\theta(x_{0:N}) = q(x_N)\prod_{k=0}^{N} p_\theta(x_{k-1}|x_k)$. Line (2.44) is derived from identity $p(x_k|x_{k-1})p(x_{k-1}|x_0) = p(x_{k-1}|x_k, x_0)p(x_k|x_0)$, which is a consequence of Bayes' rule. Line (2.45) is derived from a telescopic sum of the second logarithmic product term, and then a shift in index within the summation. The final equation is simply according to the definition of the Kullback–Leibler divergence.

The first term in (2.46) may be ignored as it is not dependent on θ. Choosing $x_0 \sim q_\theta(\cdot|x_1)$ to be Gaussian results in the last term of (2.46) having a similar form to the other KL terms. By Bayes' rule, the single-step bridge posterior $p(x_{k-1}|x_k, x_0)$ is Gaussian denoted $x_{k-1}|x_k, x_0 \sim \mathcal{N}(\tilde{\mu}_k, \tilde{\Sigma}_k)$ where

$$\tilde{\mu}_k = \frac{\beta_k\sqrt{\bar{\alpha}_{k-1}}}{1-\bar{\alpha}_k}x_0 + \frac{1-\bar{\alpha}_{k-1}}{1-\bar{\alpha}_k}\sqrt{1-\beta_k}x_k, \qquad \tilde{\Sigma}_k = \frac{1-\bar{\alpha}_{k-1}}{1-\bar{\alpha}_k}\beta_k\mathbb{I}.$$

For choice $\sigma_k^2 = \beta_k$, the remaining divergence terms may be expressed as follows, up to a constant:

$$\mathrm{KL}(p(x_{k-1}|x_k, x_0)||q_\theta(x_{k-1}|x_k)) = \frac{1}{2\sigma_k^2}\|\tilde{\mu}_k - \mu_\theta(x_k, k)\|^2 + \text{const.} \tag{2.47}$$

Given both x_k, the input to the network, and $\tilde{\mu}_k$ are linear combinations of x_0 and noise $\epsilon \sim \mathcal{N}(0, \mathbb{I})$, for different network parameterizations $\mu_\theta, \epsilon_\theta, x_{0,\theta}$:

$$|\tilde{\mu}_k - \mu_\theta(x_k, k)\|^2 \propto \|\epsilon - \epsilon_\theta(x_k, k)\|^2 \propto \|x_0 - x_{0,\theta}(x_k, k)\|^2, \tag{2.48}$$

hence one may re-write the loss terms for ϵ, x_0 and $\tilde{\mu_k}$ prediction, or indeed any other linear combination of x_0 and ϵ.

Sequential Denoising Autoencoders

As discussed above, one such parameterization of the loss for the variational discrete Markov chain is in predicting x_0 for different noise levels, indexed by t. This is essentially sequential denoising autoencoding [25].

Variational Autoencoder with Fixed Encoder

Another autoencoding interpretation is to view the forward noising process as an encoder, transforming data to a latent Gaussian distribution, and view the learnt reverse process as a decoder. This interpretation views the forward process as a fixed, infinite depth encoder; and the backward process as an infinite depth decoder [95].

An interesting observation given in the Appendix D.4 of **Chapter 5** and in [7] is that in the Schrödinger bridge setting, there is no *variational gap* in the autoencoder interpretation as the encoder is no longer fixed but is trained to map strictly to a Gaussian.

Continuous-time Normalizing Flow

There is a close relationship between the SDE interpretation of generative diffusion models and continuous normalizing flows. [88] details how one may transform the generative SDE (2.49) into a continuous-time normalizing flow (2.50) with the same marginal distributions $p_t(x_t)$ at each time point $t \in [0, T]$, given below:

$$\textbf{SDE}: \qquad \mathrm{d}\mathbf{Y}_t = \left[-f(\mathbf{Y}_t, t) + g^2(t)\nabla \log p_{T-t}(\mathbf{Y}_t)\right] \mathrm{d}t + g(t)\mathrm{d}\tilde{\mathbf{B}}_t, \tag{2.49}$$

$$\textbf{ODE}: \qquad \mathrm{d}\mathbf{Y}_t = \left[-f(\mathbf{Y}_t, t) + \tfrac{g^2(t)}{2}\nabla \log p_{T-t}(\mathbf{Y}_t)\right] \mathrm{d}t. \tag{2.50}$$

This permits likelihood computation in a similar way to continuous normalizing flows/ neural ODEs using the instantaneous change of variables formula of [15]. Likelihood computation by converting the generative SDE to probability flow ODE makes the assumption that the SDE has fully converged from data to Gaussian, this may not necessarily be true in practice.

Instead of sampling from the SDE to generate samples, one may also sample from the corresponding ODE. This opens up many forms of deterministic samplers and deterministic distillation [70].

Sequential Energy Based Models

Given the score, $\nabla \log p_t$, is a gradient, instead of learning the score directly with a neural network $s_\theta(x_t, t) \approx \nabla \log p_t(x_t)$, one could learn a sequence of energy functions $(E_\theta(\cdot, t))_t$, such that $-\nabla_x E_\theta(x_t, t) \approx \nabla \log p_t(x_t)$, where the gradient of the energy function may be computed using auto-gradient tools. This was first successfully achieved in [71] by using a specific parameterization and similar network architecture to those used in diffusion models. Although a conservative vector field, the generative performance remains similar to diffusion models but sampling time is significantly slower, due to requiring auto-gradient computation at each step.

There are however a number of benefits of learning an energy based formulation, it permits composition of multiple score-networks through approximate Metropolis rejection steps during the sampling procedure, such as through Hamiltonian Monte Carlo [32]. The downside is that it is difficult to parameterize E_θ, slow to train as it requires calling auto-gradient twice through E_θ at each training step, and slow to generate new samples as each step of the iterating sampling procedure also requires calling auto-gradient, which is expensive.

Diffusion Schrödinger Bridge

The final interpretation discussed in this section is that of the Schrödinger bridge. This will be given a proper treatment in **Chapter 5**. Essentially, the time-reversal

involved in training a diffusion model is equivalent to finding the diffusion which minimizes the Kullback–Leibler divergence in path-space to the forward process, with fixed marginal. As discussed in 2.1.4, a general approach to solving the Schrödinger bridge problem is through iterated Kullback–Leibler projections, hence iterated time-reversals, known as iterative proportional fitting or IPF [38]. One may therefore perform each step of the IPF procedure by training a diffusion model [7].

For some special cases, iterated time-reversal is not needed.

- Converged Forward Process. If the initial forward process converges to the desired terminal marginal distribution, then only a single time-reversal is sufficient for the IPF procedure to converge, hence a diffusion model with converged forward noising process is a Schrödinger bridge. Importantly, this means the Schrödinger bridge methodology is a generalization of diffusion models. This special case corresponds to infinite regularization in the corresponding OT problem. Although this provides good generative performance, it results in the independent, and hence non-meaningful, coupling between the two marginal measures. Slight variations in the initialization for the reverse process $x_T \sim \alpha$, may result in vastly different generated samples.

- Degenerate Marginals. If one of the marginals is a Dirac measure, $\beta = \delta_{x_0}$[2], then a single time-reversal will also result in a Schrödinger Bridge. This approach coincides with diffusion bridge [46] when the other measure is also a Dirac measure. This special case has been used for generative modeling with diffusion models in [97] with a slight variation. Instead of a Schrödinger bridge between marginals, the approach of [97] considers a two part process where the generative diffusion is initialized from a single point, x_T to a noised data distribution, then a second denoising process to generate samples.

- Bridging an optimal coupling, with linear reference process. Recall KL decomposition $\mathrm{KL}(\pi|p) = \mathrm{KL}(\pi_{0,T}||p_{0,T}) + \mathbb{E}_{\pi_{0,T}}[\mathrm{KL}(\pi_{|0,T}||p_{|0,T})]$. If the optimal

[2]Here β refers to marginal measure in OT not time-scaling.

coupling is known then the first term, $\mathrm{KL}(\pi_{0,T}||p_{0,T})$, is minimized, hence not required in order to learn the Schrödinger bridge. For the specific case of a linear reference process, such as Brownian motion or an Ornstein Uhlenbeck process, one may sample from and evaluate the density of the reference diffusion bridge, $p_{|0,T}$ in closed form. Indeed, given an optimal coupling $\pi_{0,T}$ between marginals α, β, one may train a Schrödinger bridge by first sampling from the coupling $x_0, x_T \sim \pi_{0,T}$, then learning a the time-reversal of the reference bridge conditioned on x_0, x_T, using score-matching techniques. The time-reversal of the bridge minimizes $\mathbb{E}_{\pi_{0,T}}[\mathrm{KL}(\pi_{|0,T}|p_{|0,T})]$. This approach was first exploited by [94]. Other than reference process limitations, the primary challenge for this method is that computing the optimal coupling is often more difficult than the time-reversal of a diffusion, especially for high dimensional data. [67] follows the same approach, using Sinkhorn on minibatches with the hope that the coupling is approximately optimal.

Schrödinger bridge vs diffusion bridge

As discussed above, the Schrödinger bridge is essentially a diffusion bridge, close to a given reference bridge, averaged across samples from an optimal coupling. However, importantly, here the coupling is optimal with respect to the reference diffusion marginals i.e. minimizes $\mathrm{KL}(\pi_{0,T}||p_{0,T})$.

Recent works [57, 83] build diffusion bridges [46] with reference Brownian motion, minimizing $\mathbb{E}_{\pi_{0,T}}[\mathrm{KL}(\pi_{|0,T}||p_{|0,T})]$, but with data driven couplings. Although well performing, strictly speaking it is not clear that this approach yields a *Schrödinger bridge*. Given that the couplings provided and not computed, there is no clear reason they would form an optimal coupling corresponding to the chosen reference diffusion, i.e. the coupling likely does not minimize $\mathrm{KL}(\pi_{0,T}||p_{0,T})$.

- Gaussian to Gaussian. When both marginal measures are Gaussian, there exists a unique and closed-form solution to the Schrödinger bridge problem [10, 16], hence does not require IPF. While limited in application itself, [10] has shown it to be a good initializer for DSB [7] and provides an alternate reference diffusion for the Schrödinger bridge.

Chapter 3

Differentiable Particle Filtering via Entropy Regularized Optimal Transport

Differentiable Particle Filtering via Entropy-Regularized Optimal Transport

Adrien Corenflos [*1] **James Thornton** [*2] **George Deligiannidis** [2] **Arnaud Doucet** [2]

Abstract

Particle Filtering (PF) methods are an established class of procedures for performing inference in non-linear state-space models. Resampling is a key ingredient of PF, necessary to obtain low variance likelihood and states estimates. However, traditional resampling methods result in PF-based loss functions being non-differentiable with respect to model and PF parameters. In a variational inference context, resampling also yields high variance gradient estimates of the PF-based evidence lower bound. By leveraging optimal transport ideas, we introduce a principled differentiable particle filter and provide convergence results. We demonstrate this novel method on a variety of applications.

1. Introduction

In this section we provide a brief introduction to state-space models (SSMs) and PF methods. We then illustrate one of the well-known limitations of PF (Kantas et al., 2015): resampling steps are required in order to compute low-variance estimates, but these estimates are not differentiable w.r.t. to model and PF parameters. This hinders end-to-end training. We discuss recent approaches to address this problem in econometrics, statistics and machine learning (ML), outline their limitations and our contributions.

1.1. State-Space Models

SSMs are an expressive class of sequential models, used in numerous scientific domains including econometrics, ecology, ML and robotics; see e.g. (Chopin & Papaspiliopoulos, 2020; Douc et al., 2014; Doucet & Lee, 2018; Kitagawa & Gersch, 1996; Lindsten & Schön, 2013; Thrun et al., 2005). SSM may be characterized by a latent $\mathcal{X}$-valued Markov process $(X_t)_{t\geq 1}$ and $\mathcal{Y}$-valued observations $(Y_t)_{t\geq 1}$ satisfying $X_1 \sim \mu_\theta(\cdot)$ and for $t \geq 1$

$$X_{t+1}|\{X_t = x\} \sim f_\theta(\cdot|x), \; Y_t|\{X_t = x\} \sim g_\theta(\cdot|x), \quad (1)$$

where $\theta \in \Theta$ is a parameter of interest. Given observations $(y_t)_{t\geq 1}$ and parameter values θ, one may perform state inference at time t by computing the posterior of X_t given $y_{1:t} \coloneqq (y_1, ..., y_t)$ where

$$p_\theta(x_t|y_{1:t-1}) = \int f_\theta(x_t|x_{t-1}) p_\theta(x_{t-1}|y_{1:t-1}) \mathrm{d}x_{t-1},$$

$$p_\theta(x_t|y_{1:t}) = \frac{g_\theta(y_t|x_t) p_\theta(x_t|y_{1:t-1})}{\int g_\theta(y_t|x_t) p_\theta(x_t|y_{1:t-1}) \mathrm{d}x_t},$$

with $p_\theta(x_1|y_0) \coloneqq \mu_\theta(x_1)$.

The log-likelihood $\ell(\theta) = \log p_\theta(y_{1:T})$ is then given by

$$\ell(\theta) = \sum_{t=1}^{T} \log p_\theta(y_t|y_{1:t-1}),$$

with $p_\theta(y_1|y_0) \coloneqq \int g_\theta(y_1|x_1)\mu_\theta(x_1)\mathrm{d}x_1$ and for $t \geq 2$

$$p_\theta(y_t|y_{1:t-1}) = \int g_\theta(y_t|x_t) p_\theta(x_t|y_{1:t-1}) \mathrm{d}x_t.$$

The posteriors $p_\theta(x_t|y_{1:t})$ and log-likelihood $p_\theta(y_{1:T})$ are available analytically for only a very restricted class of SSM such as linear Gaussian models. For non-linear SSM, PF provides approximations of such quantities.

1.2. Particle Filtering

PF are Monte Carlo methods entailing the propagation of N weighted particles $(w_t^i, X_t^i)_{i\in[N]}$, here $[N] \coloneqq \{1, ..., N\}$, over time to approximate the filtering distributions $p_\theta(x_t|y_{1:t})$ and log-likelihood $\ell(\theta)$. Here $X_t^i \in \mathcal{X}$ denotes the value of the i^{th} particle at time t and $\mathbf{w}_t \coloneqq (w_t^1, ..., w_t^N)$ are weights satisfying $w_t^i \geq 0$, $\sum_{i=1}^N w_t^i = 1$. Unlike variational methods, PF methods provide consistent approximations under weak assumptions as $N \to \infty$ (Del Moral, 2004). In the general setting, particles are sampled according to proposal distributions $q_\phi(x_1|y_1)$ at time $t = 1$ and $q_\phi(x_t|x_{t-1}, y_t)$ at time $t \geq 2$ prior to weighting and resampling. One often chooses $\theta = \phi$ but this is not necessarily the case (Le et al., 2018; Maddison et al., 2017; Naesseth et al., 2018).

[*]Equal contribution , order at discretion of authors. [1]Department of Electrical Engineering and Automation, Aalto University [2]Department of Statistics, University of Oxford. Correspondence to: Adrien Corenflos <adrien.corenflos@aalto.fi>, James Thornton <james.thornton@spc.ox.ac.uk>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

Algorithm 1 Standard Particle Filter

1: Sample $X_1^i \overset{\text{i.i.d.}}{\sim} q_\phi(\cdot|y_1)$ for $i \in [N]$
2: Compute $\omega_1^i = \frac{p_\theta(X_1^i, y_1)}{q_\phi(X_1^i|y_1)}$ for $i \in [N]$
3: $\hat{\ell}(\theta) \leftarrow \frac{1}{N}\sum_{i=1}^N \omega_1^i$
4: **for** $t = 2, ..., T$ **do**
5: Normalize weights $w_{t-1}^i \propto \omega_{t-1}^i$, $\sum_{i=1}^N w_{t-1}^i = 1$
6: Resample $\tilde{X}_{t-1}^i \sim \sum_{i=1}^N w_{t-1}^i \delta_{X_{t-1}^i}$ for $i \in [N]$
7: Sample $X_t^i \sim q_\phi(\cdot|\tilde{X}_{t-1}^i, y_t)$ for $i \in [N]$
8: Compute $\omega_t^i = \frac{p_\theta(X_t^i, y_t|\tilde{X}_{t-1}^i)}{q_\phi(X_t^i|\tilde{X}_{t-1}^i, y_t)}$ for $i \in [N]$
9: Compute $\hat{p}_\theta(y_t|y_{1:t-1}) = \frac{1}{N}\sum_{i=1}^N \omega_t^i$
10: $\hat{\ell}(\theta) \leftarrow \hat{\ell}(\theta) + \log \hat{p}_\theta(y_t|y_{1:t-1})$
11: **end for**
12: **Return:** log-likelihood estimate $\hat{\ell}(\theta) = \log \hat{p}_\theta(y_{1:T})$

A generic PF is described in Algorithm 1 where $p_\theta(x_1, y_1) := \mu_\theta(x_1) g_\theta(y_1|x_1)$ and $p_\theta(x_t, y_t|x_{t-1}) := f_\theta(x_t|x_{t-1}) g_\theta(y_t|x_t)$. Resampling is performed in step 6 of Algorithm 1; it ensures particles with high weights are replicated and those with low weights are discarded, allowing one to focus computational efforts on 'promising' regions. The scheme used in Algorithm 1 is known as multinomial resampling and is unbiased (as are other traditional schemes such as stratified and systematic (Chopin & Papaspiliopoulos, 2020)), i.e.

$$\mathbb{E}\left[\frac{1}{N}\sum_{i=1}^N \psi(\tilde{X}_t^i)\right] = \mathbb{E}\left[\sum_{i=1}^N w_t^i \psi(X_t^i)\right], \qquad (2)$$

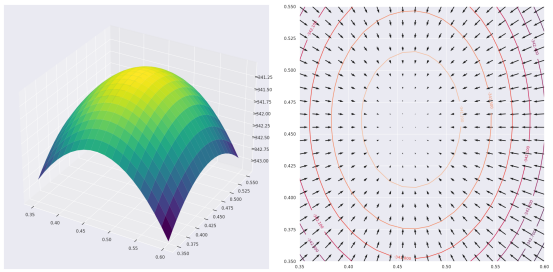
for any $\psi : \mathcal{X} \to \mathbb{R}$. This property guarantees $\exp(\hat{\ell}(\theta))$ is an unbiased estimate of the likelihood $\exp(\ell(\theta))$ for any N.

Henceforth, let $\mathcal{X} = \mathbb{R}^{d_x}$, $\theta \in \Theta = \mathbb{R}^{d_\theta}$ and $\phi \in \Phi = \mathbb{R}^{d_\phi}$. We assume here that $\theta \mapsto \mu_\theta(x)$, $\theta \mapsto f_\theta(x'|x)$ and $\theta \mapsto g_\theta(y_t|x)$ are differentiable for all x, x' and $t \in [T]$ and $\theta \mapsto \ell(\theta)$ is differentiable. These assumptions are satisfied by a large class of SSMs. We also assume that we can use the reparameterization trick (Kingma & Welling, 2014) to sample the particles; i.e. we have $\Gamma_\phi(y_1, U) \sim q_\phi(x_1|y_1)$, $\Psi_\phi(y_t, x_{t-1}, U) \sim q_\phi(x_t|x_{t-1}, y_t)$ for some mappings Γ_ϕ, Ψ_ϕ differentiable w.r.t. ϕ and $U \sim \lambda$, λ being independent of ϕ.

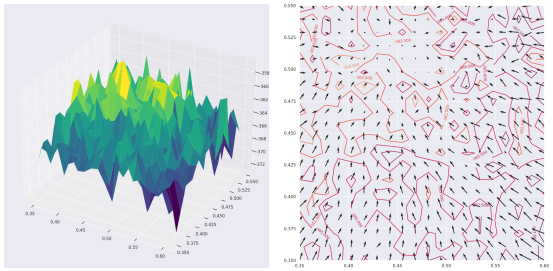
1.3. Related Work and Contributions

Let $\mathbf{U}$ be the set of all random variables used to sample and resample the particles. The distribution of $\mathbf{U}$ is (θ, ϕ)-independent as we use the reparameterization trick[1]. However, even if we sample and fix $\mathbf{U} = \mathbf{u}$, resampling involves sampling from an atomic distribution and introduces discontinuities in the particles selected when θ, ϕ vary.

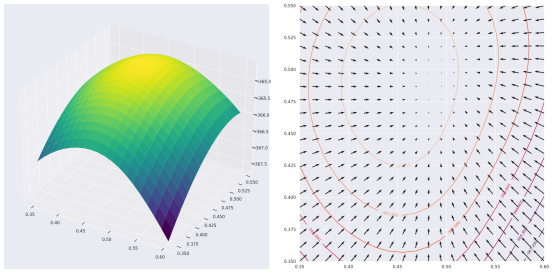
[1]For example, multinomial resampling relies on N uniform random variables.



(a) Kalman Filter



(b) Standard PF



(c) Differentiable PF

Figure 1. Left: Log-likelihood $\ell(\theta)$ and PF estimates $\hat{\ell}(\theta; \phi, \mathbf{u})$ for linear Gaussian SSM, given in Section 5.1, with $d_\theta = 2$ $d_x = 2$, and $T = 150$, $N = 50$. Right: $\nabla_\theta \ell(\theta)$ and $\nabla_\theta \hat{\ell}(\theta; \phi, \mathbf{u})$.

For $d_x = 1$, Malik & Pitt (2011) make $\theta \mapsto \hat{\ell}(\theta; \phi, \mathbf{u})$ continuous w.r.t. θ by sorting the particles and then sampling from a smooth approximation of their cumulative distribution function. For $d_x > 1$, Lee (2008) proposes a smoother but only piecewise continuous estimate. DeJong et al. (2013) returns a differentiable log-likelihood estimate $\hat{\ell}(\theta; \phi, \mathbf{u})$ by using a marginal PF (Klaas et al., 2005), where importance sampling is performed on a collapsed state-space. However, the standard marginal PF uses the proposal $q_\phi(x_t) := \sum_{i=1}^N w_{t-1}^i q_\phi(x_t|X_{t-1}^i, y_t)$ from which one cannot generally sample smoothly for arbitrary mixture components. As a consequence they instead suggest using a simple Gaussian distribution for $q_\phi(x_t)$, which can lead to poor estimates for multimodal posteriors. Moreover, in contrast to standard PF, this marginal PF cannot be applied in scenarios where the transition density can only be sampled from (e.g. using the reparameterization trick) but not evaluated pointwise (Murray et al., 2013), as the importance weight would be intractable. The implicit

reparameterization method of Graves (2016) may be used to obtain low variance gradient estimates with a mixture proposal. This method however is only compatible for component distributions with tractable conditional CDFs, such as Gaussian distributions. An alternative unbiased estimate of the likelihood based on dynamic programming may also be obtained (Finke et al., 2016; Aitchison, 2019). As emphasized by Aitchison (2019), this estimate is differentiable. This approach is again limited however to a restricted class of proposal distributions, such as an unweighted mixture proposal, which may perform poorly for slow-mixing time-series.

In the context of robot localization, a modified resampling scheme has been proposed in (Karkus et al., 2018; Ma et al., 2020a;b) referred to as 'soft-resampling' (SPF). SPF has parameter $\alpha \in [0,1]$ where $\alpha = 1$ corresponds to regular PF resampling and $\alpha = 0$ is essentially sampling particles uniformly at random. The resulting PF-net is said to be differentiable but computes gradients that ignore the non-differentiable component of the resampling step. Jonschkowski et al. (2018) proposed another PF scheme which is said to be differentiable but simply ignores the non-differentiable resampling terms and proposes new states based on the observation and some neural network. This approach however does not propagate gradients through time. Finally, Zhu et al. (2020) propose a differentiable resampling scheme based on transformers but they report that the best results are achieved when not backpropagating through it, due to exploding gradients. Hence no fully differentiable PF is currently available in the literature (Kloss et al., 2020).

PF methods have also been fruitfully exploited in Variational Inference (VI) to estimate θ, ϕ (Le et al., 2018; Maddison et al., 2017; Naesseth et al., 2018). As $\mathbb{E}_{\mathbf{U}}[\exp\left(\hat{\ell}(\theta;\phi,\mathbf{U})\right)] = \exp(\ell(\theta))$ is an unbiased estimate of $\exp(\ell(\theta))$ for any N, ϕ for standard PF, then one has indeed by Jensen's inequality

$$\ell^{\mathrm{ELBO}}(\theta,\phi) := \mathbb{E}_{\mathbf{U}}[\hat{\ell}(\theta;\phi,\mathbf{U})] \leq \ell(\theta). \tag{3}$$

The standard ELBO corresponds to $N = 1$ and many variational families for approximating $p_\theta(x_{1:T}|y_{1:T})$ have been proposed in this context (Archer et al., 2015; Krishnan et al., 2017; Rangapuram et al., 2018). The variational family induced by a PF differs significantly as $\ell^{\mathrm{ELBO}}(\theta,\phi) \to \ell(\theta)$ as $N \to \infty$ and thus yields a variational approximation converging to $p_\theta(x_{1:T}|y_{1:T})$. This attractive property comes at a computational cost; i.e. the PF approach trades off fidelity to the posterior with computational complexity. While unbiased gradient estimates of the PF-ELBO (3) can be computed, they suffer from high variance as the resampling steps require having to use REINFORCE gradient estimates (Williams, 1992). Consequently, Hirt & Dellaportas (2019); Le et al. (2018); Maddison et al. (2017); Naesseth et al. (2018) use biased gradient estimates which ignore these terms, yet report improvements as N increases over standard VI approaches and Importance Weighted Auto-Encoders (IWAE) (Burda et al., 2016).

Finally, if one is only interested in estimating θ (and not some distinct ϕ), then particle techniques approximating pointwise the score vector $\nabla_\theta \ell(\theta)$ are also available (Poyiadjis et al., 2011; Kantas et al., 2015).

The contributions of this paper are four-fold.

- We propose the first fully Differentiable Particle Filter (DPF) which can use general proposal distributions. DPF provides a differentiable estimate of $\ell(\theta)$, see Figure 1-c, and more generally differentiable estimates of PF-based losses. Empirically, in a VI context, DPF-ELBO gradient estimates also exhibit much smaller variance than those of PF-ELBO.
- We provide quantitative convergence results on the differentiable resampling scheme and establish consistency results for DPF.
- We show that existing techniques provide inconsistent gradient estimates and that the non-vanishing bias can be very significant, leading practically to unreliable parameter estimates.
- We demonstrate that DPF empirically outperforms recent alternatives for end-to-end parameter estimation on a variety of applications.

Proofs of results are given in the Supplementary Material.

2. Resampling via Optimal Transport

2.1. Optimal Transport and the Wasserstein Metric

Since Optimal Transport (OT) (Peyré & Cuturi, 2019; Villani, 2008) is a core component of our scheme, the basics are presented here. Given two probability measures α, β on $\mathcal{X} = \mathbb{R}^{d_x}$ the squared 2-Wasserstein metric between these measures is given by

$$\mathcal{W}_2^2(\alpha,\beta) = \min_{\mathcal{P}\in\mathcal{U}(\alpha,\beta)} \mathbb{E}_{(U,V)\sim\mathcal{P}}\left[||U-V||^2\right], \tag{4}$$

where $\mathcal{U}(\alpha,\beta)$ the set of distributions on $\mathcal{X}\times\mathcal{X}$ with marginals α and β, and the minimizing argument of (4) is the OT plan denoted $\mathcal{P}^{\mathrm{OT}}$. Any element $\mathcal{P} \in \mathcal{U}(\alpha,\beta)$ allows one to "transport" α to β (and vice-versa) i.e.

$$\beta(\mathrm{d}v) = \int \mathcal{P}(\mathrm{d}u, \mathrm{d}v) = \int \mathcal{P}(\mathrm{d}v|u)\alpha(\mathrm{d}u).$$

For atomic probability measures $\alpha_N = \sum_{i=1}^N a_i \delta_{u_i}$ and $\beta_N = \sum_{j=1}^N b_j \delta_{v_j}$ with weights $\mathbf{a} = (a_i)_{i\in[N]}$, $\mathbf{b} = (b_j)_{j\in[N]}$, and atoms $\mathbf{u} = (u_i)_{i\in[N]}$, $\mathbf{v} = (v_j)_{j\in[N]}$, one can show that

$$\mathcal{W}_2^2(\alpha_N,\beta_N) = \min_{\mathbf{P}\in\mathcal{S}(\mathbf{a},\mathbf{b})} \sum_{i=1}^N \sum_{j=1}^N c_{i,j} p_{i,j}, \tag{5}$$

where any $\mathcal{P} \in \mathcal{U}(\alpha_N, \beta_N)$ is of the form

$$\mathcal{P}(\mathrm{d}u, \mathrm{d}v) = \sum_{i,j} p_{i,j} \delta_{u_i}(\mathrm{d}u)\delta_{v_j}(\mathrm{d}v),$$

$c_{i,j} = ||u_i - v_j||^2$, $\mathbf{P} = (p_{i,j})_{i,j\in[N]}$ and $\mathcal{S}(\mathbf{a}, \mathbf{b}) = \{\mathbf{P} \in [0,1]^{N\times N} : \sum_{j=1}^N p_{i,j} = a_i,\ \sum_{i=1}^N p_{i,j} = b_j\}$. In such cases, one has

$$\mathcal{P}(\mathrm{d}v|u = u_i) = \sum_j a_i^{-1} p_{i,j}\delta_{v_j}(\mathrm{d}v). \tag{6}$$

The optimization problem (5) may be solved through linear programming. It is also possible to exploit the dual formulation

$$\mathcal{W}_2^2(\alpha_N, \beta_N) = \max_{\mathbf{f},\mathbf{g}\in\mathcal{R}(C)} \mathbf{a}^t\mathbf{f} + \mathbf{b}^t\mathbf{g}, \tag{7}$$

where $\mathbf{f} = (f_i)$, $\mathbf{g} = (g_i)$, $\mathbf{C} = (c_{i,j})$ and $\mathcal{R}(\mathbf{C}) = \{\mathbf{f}, \mathbf{g} \in \mathbb{R}^N | f_i + g_j \le c_{i,j}, i, j \in [N]\}$.

2.2. Ensemble Transform Resampling

The use of OT for resampling in PF has been pioneered by Reich (2013). Unlike standard resampling schemes (Chopin & Papaspiliopoulos, 2020; Doucet & Lee, 2018), it relies not only on the particle weights but also on their locations.

At time t, after the sampling step (Step 7 in Algorithm 1), $\alpha_N^{(t)} = \frac{1}{N}\sum_{i=1}^N \delta_{X_t^i}$ is a particle approximation of $\alpha^{(t)} := \int q_\phi(x_t|x_{t-1}, y_t)p_\theta(x_{t-1}|y_{1:t-1})\mathrm{d}x_{t-1}$ and $\beta_N^{(t)} = \sum w_t^i \delta_{X_t^i}$ is an approximation of $\beta^{(t)} := p_\theta(x_t|y_{1:t})$. Under mild regularity conditions, the OT plan minimizing $\mathcal{W}_2(\alpha^{(t)}, \beta^{(t)})$ is of the form $\mathcal{P}^{\mathrm{OT}}(\mathrm{d}x, \mathrm{d}x') = \alpha^{(t)}(\mathrm{d}x)\delta_{\mathbf{T}^{(t)}(x)}(\mathrm{d}x')$ where $\mathbf{T}^{(t)} : \mathcal{X} \to \mathcal{X}$ is a deterministic map; i.e if $X \sim \alpha^{(t)}$ then $\mathbf{T}^{(t)}(X) \sim \beta^{(t)}$. It is shown in (Reich, 2013) that one can one approximate this transport map with the 'Ensemble Transform' (ET) denoted $\mathbf{T}_N^{(t)}$. This is found by solving the OT problem (5) between $\alpha_N^{(t)}$ and $\beta_N^{(t)}$ and taking an expectation w.r.t. (6), that is

$$\tilde{X}_t^i = N\sum_{k=1}^N p_{i,k}^{\mathrm{OT}} X_t^k := \mathbf{T}_N^{(t)}(X_t^i), \tag{8}$$

where we slightly abuse notation as $\mathbf{T}_N^{(t)}$ is a function of $X_t^{1:N}$. Reich (2013) uses this update instead of using $\tilde{X}_t^i \sim \sum_{i=1}^N w_t^i \delta_{X_t^i}$. This is justified by the fact that, as $N \to \infty$, $\mathbf{T}_N^{(t)}(X_t^i) \to \mathbf{T}^{(t)}(X_t^i)$ in some weak sense (Reich, 2013; Myers et al., 2021). Compared to standard resampling schemes, the ET only satisfies (2) for affine functions ψ.

This OT approach to resampling involves solving the linear program (4) at cost $O(N^3 \log N)$ (Bertsimas & Tsitsiklis, 1997). This is not only prohibitively expensive but moreover the resulting ET is not differentiable. To address these problems, one may instead rely on entropy-regularized OT (Cuturi, 2013).

3. Differentiable Resampling via Entropy-Regularized Optimal Transport

3.1. Entropy-Regularized Optimal Transport

Entropy-regularized OT may be used to compute a transport matrix that is differentiable with respect to inputs and computationally cheaper than the non-regularized version, i.e. we consider the following regularized version of (5) for some $\epsilon > 0$ (Cuturi, 2013; Peyré & Cuturi, 2019)

$$\mathcal{W}_{2,\epsilon}^2(\alpha_N, \beta_N) = \min_{\mathbf{P}\in\mathcal{S}(\mathbf{a},\mathbf{b})} \sum_{i,j=1}^N p_{i,j}\left(c_{i,j} + \epsilon \log \frac{p_{i,j}}{a_i b_j}\right). \tag{9}$$

The function minimized in (9) is strictly convex and hence admits a unique minimizing argument $\mathbf{P}_\epsilon^{\mathrm{OT}} = (p_{\epsilon,i,j}^{\mathrm{OT}})$. $\mathcal{W}_{2,\epsilon}^2(\alpha_N, \beta_N)$ can also be computed using the regularized dual; i.e. $\mathcal{W}_{2,\epsilon}^2(\alpha_N, \beta_N) = \max_{\mathbf{f},\mathbf{g}} \mathrm{DOT}_\epsilon(\mathbf{f}, \mathbf{g})$ with

$$\mathrm{DOT}_\epsilon(\mathbf{f}, \mathbf{g}) := \mathbf{a}^t\mathbf{f} + \mathbf{b}^t\mathbf{g} - \epsilon\mathbf{a}^t\mathbf{M}\mathbf{b} \tag{10}$$

where $(\mathbf{M})_{i,j} := \exp\left(\epsilon^{-1}(f_i + g_j - c_{i,j})\right) - 1$ and $\mathbf{f}, \mathbf{g}$ are now unconstrained. For the dual pair $(\mathbf{f}^*, \mathbf{g}^*)$ maximizing (10), we have $\nabla_{\mathbf{f},\mathbf{g}}\mathrm{DOT}_\epsilon(\mathbf{f}, \mathbf{g})|_{(\mathbf{f}^*,\mathbf{g}^*)} = \mathbf{0}$. This first-order condition leads to

$$f_i^* = \mathcal{T}_\epsilon(\mathbf{b}, \mathbf{g}^*, \mathbf{C}_{i:}), \qquad g_i^* = \mathcal{T}_\epsilon(\mathbf{a}, \mathbf{f}^*, \mathbf{C}_{:i}), \tag{11}$$

where $\mathbf{C}_{i:}$ (resp. $\mathbf{C}_{:i}$) is the i^{th} row (resp. column) of $\mathbf{C}$. Here $\mathcal{T}_\epsilon : \mathbb{R}^N \times \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}^N$ denotes the mapping

$$\mathcal{T}_\epsilon(\mathbf{a}, \mathbf{f}, \mathbf{C}_{:,i}) = -\epsilon \log \sum_k \exp\left\{\log a_k + \epsilon^{-1}(f_k - c_{k,i})\right\}.$$

One may then recover the regularized transport matrix as

$$p_{\epsilon,i,j}^{\mathrm{OT}} = a_i b_j \exp\left(\epsilon^{-1}(f_i^* + g_j^* - c_{i,j})\right). \tag{12}$$

The dual can be maximized using the Sinkhorn algorithm introduced for OT in the seminal paper of Cuturi (2013). Algorithm 2 presents the implementation of Feydy et al. (2019) where the fixed point updates based on Equation (11) have been stabilized.

Algorithm 2 Sinkhorn Algorithm

Function $\mathbf{Potentials}(\mathbf{a}, \mathbf{b}, \mathbf{u}, \mathbf{v})$
Local variables: $\mathbf{f}, \mathbf{g} \in \mathbb{R}^N$
Initialize: $\mathbf{f} = \mathbf{0}$, $\mathbf{g} = \mathbf{0}$
Set $\mathbf{C} \leftarrow \mathbf{u}\mathbf{u}^t + \mathbf{v}\mathbf{v}^t - 2\mathbf{u}\mathbf{v}^t$
while stopping criterion not met **do**
  for $i \in [N]$ **do**
    $f_i \leftarrow \frac{1}{2}\left(f_i + \mathcal{T}_\epsilon(\mathbf{b}, \mathbf{g}, \mathbf{C}_{i:})\right)$
    $g_i \leftarrow \frac{1}{2}\left(g_i + \mathcal{T}_\epsilon(\mathbf{a}, \mathbf{f}, \mathbf{C}_{:i})\right)$
  end for
end while
Return $\mathbf{f}, \mathbf{g}$

The resulting dual vectors $(\mathbf{f}^*, \mathbf{g}^*)$ can then be differentiated for example using automatic differentiation through the Sinkhorn algorithm loop (Flamary et al., 2018), or more efficiently using "gradient stitching" on the dual vectors at convergence, which we do here (see Feydy et al. (2019) for details). The derivatives of $\mathbf{P}_\epsilon^{\mathrm{OT}}$ are readily accessible by combining the derivatives of (11) with the derivatives of (12), using automatic differentiation at no additional cost.

3.2. Differentiable Ensemble Transform Resampling

We obtain a differentiable ET (DET), denoted $\mathbf{T}_{N,\epsilon}^{(t)}$, by computing the entropy-regularized OT using Algorithm 3 for the weighted particles $(\mathbf{X}_t, \mathbf{w}_t, N)$ at time t

$$\tilde{X}_t^i = N \sum_{k=1}^N p_{\epsilon,i,k}^{\mathrm{OT}} X_t^k := \mathbf{T}_{N,\epsilon}^{(t)}(X_t^i). \tag{13}$$

Algorithm 3 DET Resampling

Function $\mathbf{EnsembleTransform}(\mathbf{X}, \mathbf{w}, N)$
$\mathbf{f}, \mathbf{g} \leftarrow \mathbf{Potentials}(\mathbf{w}, \frac{1}{N}\mathbf{1}, \mathbf{X}, \mathbf{X})$
for $i \in [N]$ **do**
  for $j \in [N]$ **do**
    $p_{\epsilon,i,j}^{\mathrm{OT}} = \frac{w_i}{N} \exp\left(\frac{f_i + g_j - c_{i,j}}{\epsilon}\right)$
  end for
end for
Return $\tilde{\mathbf{X}} = N\mathbf{P}_\epsilon^{\mathrm{OT}}\mathbf{X}$

Compared to the ET, the DET is differentiable and can be computed at cost $O(N^2)$ as it relies on the Sinkhorn algorithm. This algorithm converges quickly (Altschuler et al., 2017) and is particularly amenable to GPU implementation.

The DPF proposed in this paper is similar to Algorithm 1 except that we sample from the proposal q_ϕ using the reparameterization trick and Step 6 is replaced by the DET. While such a differentiable approximation of the ET has previously been suggested in ML (Cuturi & Doucet, 2014; Seguy et al., 2018), it has never been realized before that this could be exploited to obtain a DPF. In particular, we obtain differentiable estimates of expectations w.r.t. the filtering distributions with respect to θ and ϕ and, for a fixed "seed" $\mathbf{U} = \mathbf{u}$ [2], we obtain a differentiable estimate of the log-likelihood function $\theta \mapsto \hat{\ell}_\epsilon(\theta; \phi, \mathbf{u})$.

Like ET, DET only satisfies (2) for affine functions ψ. Unlike $\mathbf{P}^{\mathrm{OT}}$, $\mathbf{P}_\epsilon^{\mathrm{OT}}$ is sensitive to the scale of $\mathbf{X}_t$. To mitigate this sensitivity, one may compute $\delta(\mathbf{X}_t) = \sqrt{d_x} \max_{k\in[d_x]} \mathrm{std}_i(X_{t,k}^i)$ for $\mathbf{X}_t \in \mathbb{R}^{N\times d_x}$ and rescale $\mathbf{C}$ accordingly to ensure that ϵ is approximately independent of the scale and dimension of the problem.

[2] Here $\mathbf{U}$ denotes only the set of θ, ϕ-independent random variables used to generate particles as, contrary to standard PF, DET resampling does not rely on any additional random variable.

4. Theoretical Analysis

We show here that the gradient estimates of PF-based losses ignoring gradients terms due to resampling are not consistent and can suffer from a large non-vanishing bias. On the contrary, we establish that DPF provides consistent and differentiable estimates of the filtering distributions and log-likelihood function. This is achieved by obtaining novel quantitative convergence results for the DET.

4.1. Gradient Bias from Ignoring Resampling Terms

We first provide theoretical results on the asymptotic bias of the gradient estimates computed from PF-losses, by dropping the gradient terms from resampling, as adopted in (Hirt & Dellaportas, 2019; Jonschkowski et al., 2018; Karkus et al., 2018; Le et al., 2018; Ma et al., 2020b; Maddison et al., 2017; Naesseth et al., 2018). We limit ourselves here to the ELBO loss. Similar analysis can be carried out for the non-differentiable resampling schemes and losses considered in robotics.

Proposition 4.1. *Consider the PF in Algorithm 1 where ϕ is distinct from θ then, under regularity conditions, the expectation of the ELBO gradient estimate $\hat{\nabla}_\theta \ell^{\mathrm{ELBO}}(\theta, \phi)$ ignoring resampling terms considered in (Le et al., 2018; Maddison et al., 2017; Naesseth et al., 2018) converges as $N \to \infty$ to*

$$\mathbb{E}[\hat{\nabla}_\theta \ell^{\mathrm{ELBO}}(\theta, \phi)] \to \int \nabla_\theta \log p_\theta(x_1, y_1)\, p_\theta(x_1|y_1) \mathrm{d}x_1 + \sum_{t=2}^T \int \nabla_\theta \log p_\theta(x_t, y_t|x_{t-1})\, p_\theta(x_{t-1:t}|y_{1:t}) \mathrm{d}x_{t-1:t}$$

whereas Fisher's identity yields

$$\nabla_\theta \ell(\theta) = \int \nabla_\theta \log p_\theta(x_1, y_1)\, p_\theta(x_1|y_{1:T}) \mathrm{d}x_1 + \sum_{t=2}^T \int \nabla_\theta \log p_\theta(x_t, y_t|x_{t-1})\, p_\theta(x_{t-1:t}|y_{1:T}) \mathrm{d}x_{t-1:t}. \tag{14}$$

Hence, whereas we have $\nabla_\theta \ell^{\mathrm{ELBO}}(\theta, \phi) \to \nabla_\theta \ell(\theta)$ as $N \to \infty$ under regularity assumptions, the asymptotic bias of $\hat{\nabla}_\theta \ell^{\mathrm{ELBO}}(\theta, \phi)$ only vanishes if $p_\theta(x_{t-1:t}|y_{1:t}) = p_\theta(x_{t-1:t}|y_{1:T})$; i.e. for models where the X_t are independent. When $y_{t+1:T}$ do not bring significant information about X_t given $y_{t:T}$, as for the models considered in (Le et al., 2018; Maddison et al., 2017; Naesseth et al., 2018), this is a reasonable approximation which explains the good performance reported therein. However, we show in Section 5 that this bias can also lead practically to inaccurate parameter estimation.

4.2. Quantitative Bounds on the DET

Weak convergence results for the ET have been established in (Reich, 2013; Myers et al., 2021) and the DET in (Seguy

et al., 2018). We provide here the first quantitative bound for the ET ($\epsilon = 0$) and DET ($\epsilon > 0$) which holds for any $N \geq 1$ by building upon results of (Li & Nochetto, 2021) and (Weed, 2018). We use the notation $\nu(\psi) := \int \psi(x)\nu(\mathrm{d}x)$ for any measure ν and function ψ.

Proposition 4.2. *Consider atomic probability measures $\alpha_N = \sum_{i=1}^N a_i \delta_{Y^i}$ with $a_i > 0$ and $\beta_N = \sum_{i=1}^N b_i \delta_{X^i}$, with support $\mathcal{X} \subset \mathbb{R}^d$. Let $\tilde{\beta}_N = \sum_{i=1}^N a_i \delta_{\tilde{X}^i_{N,\epsilon}}$ where $\tilde{\mathbf{X}}_{N,\epsilon} = \Delta^{-1}\mathbf{P}^{\mathrm{OT}}_\epsilon \mathbf{X}$ for $\Delta = \mathrm{diag}(a_1, ..., a_N)$ and $\mathbf{P}^{\mathrm{OT}}_\epsilon$ is the transport matrix corresponding to the ϵ-regularized OT coupling, $\mathcal{P}^{\mathrm{OT},N}_\epsilon$, between α_N and β_N. Let α, β be two other probability measures, also supported on $\mathcal{X}$, such that there exists a unique λ-Lipschitz optimal transport map $\mathbf{T}$ between them. Then for any bounded 1-Lipschitz function ψ, we have*

$$\left|\beta_N(\psi) - \tilde{\beta}_N(\psi)\right| \leq 2\lambda^{1/2}\mathcal{E}^{1/2}\left[\mathfrak{d}^{1/2} + \mathcal{E}\right]^{1/2} + \max\{\lambda, 1\}\left[\mathcal{W}_2(\alpha_N, \alpha) + \mathcal{W}_2(\beta_N, \beta)\right], \quad (15)$$

where $\mathfrak{d} := \sup_{x,y\in\mathcal{X}} |x - y|$ and $\mathcal{E} = \mathcal{W}_2(\alpha_N, \alpha) + \mathcal{W}_2(\beta_N, \beta) + \sqrt{2\epsilon \log N}$.

If $\mathcal{W}_2(\alpha_N, \alpha), \mathcal{W}_2(\beta_N, \beta) \to 0$ and we choose $\epsilon_N = o(1/\log N)$ the bound given in (15) vanishes with $N \to \infty$. This suggested dependence of ϵ on N comes from the entropic radius, see Lemma C.1 in the Supplementary and (Weed, 2018), and is closely related to the fact that entropy-regularized OT is sensitive to the scale of $\mathbf{X}$. Equivalently one may rescale $\mathbf{X}$ by a factor $\log N$ when computing the cost matrix. In particular when α_N and β_N are Monte Carlo approximations of α and β, we expect $\mathcal{W}_2(\alpha_N, \alpha), \mathcal{W}_2(\beta_N, \beta) = O(N^{-1/d})$ with high probability (Fournier & Guillin, 2015).

4.3. Consistency of DPF

The parameters θ, ϕ are here fixed and omitted from notation. We now establish consistency results for DPF, showing that both the resulting particle approximations $\tilde{\beta}^{(t)}_N = \frac{1}{N}\sum_{i=1}^N \delta_{\tilde{X}^i_t}$ of $\beta^{(t)} = p(x_t|y_{1:t})$ and the corresponding log-likelihood approximation $\log \hat{p}_N(y_{1:T})$ of $\log p(y_{1:T})$ are consistent. In the interest of simplicity, we limit ourselves to the scenario where the proposal is the transition, $q = f$, so $\omega(x_{t-1}, x_t, y_t) = g(y_t|x_t)$, known as the bootstrap PF and study a slightly non-standard version of it proposed in (Del Moral & Guionnet, 2001); see Appendix D for details. Consistency is established under regularity assumptions detailed in the Supplementary. Assumption B.1 is that the space $\mathcal{X} \subset \mathbb{R}^d$ has a finite diameter $\mathfrak{d}$. Assumption B.2 implies that the proposal mixes exponentially fast in the Wasserstein sense at a rate κ, which is reasonable given compactness, and essential for the error to not accumulate. Assumption B.3 assumes a bounded importance weight function i.e. $g(y_t|x_t) \in [\Delta, \Delta^{-1}]$, again not unreasonable given compactness. Assumption B.4 states that at each time step, the optimal transport problem between $\alpha^{(t)}$ and $\beta^{(t)}$ is solved uniquely by a deterministic, globally Lipschitz map. Uniqueness is crucial for the quantitative stability results provided in the following proposition.

Proposition 4.3. *Under Assumptions B.1, B.2, B.3 and B.4, for any $\delta > 0$, with probability at least $1 - 2\delta$ over the sampling steps, for any bounded 1-Lipschitz ψ, for any $t \in [1:T]$, the approximations of the filtering distributions and log-likelihood computed by the bootstrap DPF satisfy*

$$|\tilde{\beta}^{(t)}_N(\psi) - \beta^{(t)}(\psi)| \leq \mathfrak{G}^{(t)}_{\epsilon,\delta/T,N,d}\left(\lambda(c, C, d, T, N, \delta)\right),$$

$$\left|\log \frac{\hat{p}_N(y_{1:T})}{p(y_{1:T})}\right| \leq \frac{\kappa}{\Delta}\max_{t\in[1:T]} \mathrm{Lip}\left[g(y_t \mid \cdot)\right] \times \sum_{t=1}^T \mathfrak{G}^{(t)}_{\epsilon,\delta/T,N,d}\left(\lambda(c, C, d, T, N, \delta)\right),$$

for $\lambda(c, C, d, T, N, \delta) = \sqrt{f_d^{-1}\left(\frac{\log(CT/\delta)}{cN}\right)}$ where c, C are finite constants independent of T, and $\mathrm{Lip}[f]$ is the Lipschitz constant of the function f, and $\mathfrak{G}^{(t)}_{N,\epsilon}, f_d$ defined in Appendix D are two functions such that if we set $\epsilon_N = o(1/\log N)$ then we have in probability

$$|\tilde{\beta}^{(t)}_N(\psi) - \beta^{(t)}(\psi)| \to 0, \quad \left|\log \frac{\hat{p}_N(y_{1:T})}{p(y_{1:T})}\right| \to 0.$$

The above bounds are certainly not sharp. A glimpse into the behavior of the above bounds in terms of T can be obtained through careful consideration of the quantities appearing in Proposition D.1 in the supplement. In particular, for κ small enough, it suggests that the bound on the error of the log-likelihood estimator grows linearly with T as for standard PF under mixing assumptions. Sharper bounds are certainly possible, e.g. using a L_1 version of Theorem 3.5 in (Li & Nochetto, 2021). It would also be of interest to weaken the assumptions, in particular, to remove the bounded space assumption although it is very commonly made in the PF literature to obtain quantitative bounds; see e.g. (Del Moral, 2004; Douc et al., 2014). Although this is not made explicit in the expressions above, there is an exponential dependence of the bounds on the state dimension d_x. This is unavoidable however and a well-known limitation of PF methods.

Finally note that DPF provides a biased estimate of the likelihood contrary to standard PF, so we cannot guarantee that the expectation of its logarithm, $\ell^{\mathrm{ELBO}}_\epsilon(\theta, \phi) := \mathbb{E}_{\mathbf{U}}[\hat{\ell}_\epsilon(\theta; \phi, \mathbf{U})]$. is actually a valid ELBO. However in all our experiments, see e.g. Section 5.1, $|\ell^{\mathrm{ELBO}}_\epsilon(\theta, \phi) - \ell^{\mathrm{ELBO}}(\theta, \phi)|$ is significantly smaller than $\ell(\theta) - \ell^{\mathrm{ELBO}}(\theta, \phi)$ so $\ell^{\mathrm{ELBO}}_\epsilon(\theta, \phi) < \ell(\theta)$. Hence we keep the ELBO terminology.

5. Experiments

In Section 5.1, we assess the sensitivity of the DPF to the regularization parameter ϵ. All other DPF experiments presented here use the DET Resampling detailed in Algorithm 3 with $\epsilon = 0.5$, which ensures stability of the gradient calculations while adding little bias to the calculation of the ELBO compared to standard PF. Our method is implemented in both PyTorch and TensorFlow, the code to replicate the experiments as well as further experiments may be found at `https://github.com/JTT94/filterflow`.

5.1. Linear Gaussian State-Space Model

We consider here a simple two-dimensional linear Gaussian SSM for which the exact likelihood can be computed exactly using the Kalman filter

$$X_{t+1}|\{X_t = x\} \sim \mathcal{N}\left(\text{diag}(\theta_1\ \theta_2)x, 0.5\mathbf{I}_2\right),$$
$$Y_t|\{X_t = x\} \sim \mathcal{N}(x, 0.1\mathbf{I}_2).$$

We simulate $T = 150$ observations using $\theta = (\theta_1, \theta_2) = (0.5, 0.5)$, for which we evaluate the ELBO at $\theta = (0.25, 0.25)$, $\theta = (0.5, 0.5)$, and $\theta = (0.75, 0.75)$. More precisely, using a standard PF with $N = 25$ particles, we compute the mean and standard deviation of $\frac{1}{T}(\hat{\ell}(\theta; \mathbf{U}) - \ell(\theta))$ over 100 realizations of $\mathbf{U}$. The mean is an estimate of the ELBO minus the true log-likelihood (rescaled by $1/T$). We then perform the same calculations for the DPF using the same number of particles and $\epsilon = 0.25, 0.5, 0.75$. As mentioned in Section 3.2 and Section 4.3, the DET resampling scheme is only satisfying Equation (2) for affine functions ψ so the DPF provides a biased estimate of the likelihood. Hence we cannot guarantee that the expectation of the corresponding log-likelihood estimate is a true ELBO. However, from Table 1, we observe that the difference between the ELBO estimates computed using PF and DPF is negligible for the three values of ϵ. The standard deviation of the log-likelihood estimates is also similar.

Table 1. Mean & std of $\frac{1}{T}(\hat{\ell}(\theta; \mathbf{U}) - \ell(\theta))$

θ_1, θ_2		0.25	0.5	0.75
PF	mean	-1.13	-0.93	-1.05
	std	0.20	0.18	0.17
DPF ($\epsilon = 0.25$)	mean	-1.14	-0.94	-1.07
	std	0.20	0.18	0.19
DPF ($\epsilon = 0.5$)	mean	-1.14	-0.94	-1.08
	std	0.20	0.18	0.18
DPF ($\epsilon = 0.75$)	mean	-1.14	-0.94	-1.08
	std	0.20	0.18	0.18

Recall here that alternative techniques estimating the score vector $\nabla_\theta \ell(\theta)$ by approximating (14) using particle smoothing algorithms (Poyiadjis et al., 2011; Kantas et al., 2015) could also be used to estimate θ.

5.2. Learning the Proposal Distribution

We consider a similar example as in (Naesseth et al., 2018) where one learns the parameters ϕ of the proposal using the ELBO for the following linear Gaussian SSM:

$$X_{t+1}|\{X_t = x\} \sim \mathcal{N}\left(\mathbf{A}x, \mathbf{I}_{d_x}\right), \tag{16}$$
$$Y_t|\{X_t = x\} \sim \mathcal{N}(\mathbf{I}_{d_y,d_x}x, \mathbf{I}_{d_y}), \tag{17}$$

with $\mathbf{A} = (0.42^{|i-j|+1})_{1\leq i,j\leq d_x}$, $\mathbf{I}_{d_y,d_x}$ is a $d_y \times d_x$ matrix with 1 on the diagonal for the d_y first rows and zeros elsewhere. For $\phi \in \mathbb{R}^{d_x+d_y}$, we consider

$$q_\phi(x_t|x_{t-1}, y_t) = \mathcal{N}(x_t|\Delta_\phi^{-1}\left(\mathbf{A}x_{t-1} + \Gamma_\phi y_t\right), \Delta_\phi),$$

with $\Delta_\phi = \text{diag}(\phi_1, \ldots, \phi_{d_x})$ and a $d_x \times d_y$ matrix $\Gamma_\phi = \text{diag}_{d_x,d_y}(\phi_1, \ldots, \phi_{d_x})$ with ϕ_i on the diagonal for d_x first rows and zeros elsewhere. The locally optimal proposal $p(x_t|x_{t-1}, y_t) \propto g(y_t|x_t)f(x_t|x_{t-1})$ in (Doucet & Johansen, 2009) corresponds to $\phi = \mathbf{1}$, the vector with unit entries of dimension $d_\phi = d_x + d_y$.

For $d_x = 25, d_y = 1$, $M = 100$ realizations of $T = 100$ observations using (16)-(17), we learn ϕ on each realization using 100 steps of stochastic gradient ascent with learning rate 0.1 on the $\ell^{\text{ELBO}}(\phi)$ using regular PF with biased gradients as in (Maddison et al., 2017; Le et al., 2018; Naesseth et al., 2018) and $\ell^{\text{ELBO}}(\phi)$ with four independent filters using DPF. We use $N = 500$ for regular PF and $N = 25$ for DPF so as to match the computational complexity. While $p(x_t|x_{t-1}, y_t)$ is not guaranteed to maximize the ELBO, our experiments showed that it outperforms optimized proposals. We therefore report the RMSE of $\phi - \mathbf{1}$ and the average Effective Sample Size (ESS) (Doucet & Johansen, 2009) as proxy performance metrics. On both metrics, DPF outperforms regular PF. The RMSE over 100 experiments is 0.11 for DPF vs 0.22 for regular PF while the average ESS after convergence is around 60% for DPF vs 25% for regular PF. The average time per iteration was around 15 seconds for both DPF and PF.

5.3. Variational Recurrent Neural Network (VRNN)

A VRNN is an SSM introduced by (Chung et al., 2015) to improve upon LSTMs (Long Short Term Memory networks) with the addition of a stochastic component to the hidden state, this extends variational auto-encoders to a sequential setting. Indeed let latent state be $X_t = (R_t, Z_t)$ where R_t is an RNN state and Z_t a latent Gaussian variable, here Y_t is a vector of binary observations. The VRNN is detailed as follows. RNN_θ denotes the forward call of an LSTM cell which at time t emits the next RNN state R_{t+1} and output O_{t+1}. E_θ, h_θ, μ_θ, σ_θ are fully connected neural networks; detailed fully in the Supplementary Material.

This model is trained on the polyphonic music benchmark datasets (Boulanger-Lewandowski et al., 2012), whereby Y_t represents which notes are active. The observation sequences are capped to length 150 for each dataset, with each observation of dimension 88. We chose latent states Z_t and R_t to be of dimension $d_z = 8$ and $d_r = 16$ respectively so $d_x = 24$. We use $q_\phi(x_t|x_{t-1}, y_t) = f_\theta(x_t|x_{t-1})$.

$$
\begin{aligned}
(R_{t+1}, O_{t+1}) &= \text{RNN}_\theta(R_t, Y_{1:t}, E_\theta(Z_t)), \\
Z_{t+1} &\sim \mathcal{N}(\mu_\theta(O_{t+1}), \sigma_\theta(O_{t+1})), \\
\hat{p}_{t+1} &= h_\theta(E_\theta(Z_{t+1}), O_{t+1}), \\
Y_t|X_t &\sim \text{Ber}(\hat{p}_t).
\end{aligned}
$$

Table 2. ELBO $\pm$ Standard Deviation evaluated using Test Data.

	MUSEDATA	JSB	NOTTINGHAM
DPF	$\mathbf{-7.59}_{\pm 0.01}$	$\mathbf{-7.67}_{\pm 0.08}$	$\mathbf{-3.79}_{\pm 0.02}$
PF	$-7.60_{\pm 0.06}$	$-7.92_{\pm 0.13}$	$-3.81_{\pm 0.02}$
SPF	$-7.73_{\pm 0.14}$	$-8.17_{\pm 0.07}$	$-3.91_{\pm 0.05}$

The VRNN model is trained by maximizing $\ell_\epsilon^{\text{ELBO}}(\theta)$ using DPF. We compare this to the same model trained by maximizing $\ell^{\text{ELBO}}(\theta)$ computed with regular PF (Maddison et al., 2017) and also trained with 'soft-resampling' (SPF) introduced by (Karkus et al., 2018) and described in Section 1.3, SPF is used here with parameter $\alpha = 0.1$. Unlike regular resampling, SPF partially preserves a gradient through the resampling step, however SPF still involves a non-differentiable operation, again resulting in a biased gradient. SPF also produces higher variance estimates as the resampled approximation is not uniformly weighted, essentially interpolating between PF and IWAE. Each of the methods are performed with $N = 32$ particles. Although DET is computationally more expensive than the other resampling schemes, the computational times of DPF, PF, and SPF are very similar due to most of the complexity coming from neural network operations. The learned models are then evaluated on test data using multinomial resampling for comparable ELBO results. Due to the fact that our observation model is $\text{Ber}(\hat{p}_t)$, this recovers the negative log-predictive cross-entropy.

Table 2 illustrates the benefit of using DPF over regular PF and SPF for the JSB dataset. Although DPF remains competitive compared to other heuristic approaches, the difference is relatively minor for the other datasets. We speculate that the performance of the heuristic methods is likely due to low predictive uncertainty for the next observation given the previous one.

5.4. Robot Localization

Consider the setting of a robot/agent in a maze (Jonschkowski et al., 2018; Karkus et al., 2018). Given the agent's initial state, S_1, and inputs a_t, one would like

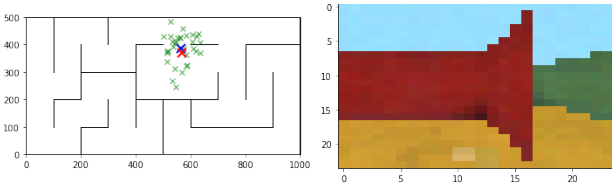


Figure 2. Left: Particles $(X_t^{(1),i}, X_t^{(2),i})$ (green), PF estimate of $\mathbb{E}[X_t|y_{1:t}]$ (blue), true state X_t^* (red). Right: Observation, O_t.

to infer the location of the agent at any specific time given observations O_t. Let the latent state be denoted $S_t = (X_t^{(1)}, X_t^{(2)}, \gamma_t)$ where $(X_t^{(1)}, X_t^{(2)})$ are location coordinates and γ_t the robot's orientation. In our setting observations O_t are images, which are encoded to extract useful features using a neural network E_θ, where $Y_t = E_\theta(O_t)$. This problem requires learning the relationship between the robot's location, orientation and the observations. Given actions $a_t = (v_t^{(1)}, v_t^{(2)}, \omega_t)$, we have

$$
\begin{aligned}
S_{t+1} &= F_\theta(S_t, a_t) + \nu_t, \quad \nu_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}, \Sigma_F), \\
Y_t &= G_\theta(S_t) + \epsilon_t, \quad \epsilon_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}, \sigma_G^2 \mathbb{I}_{e_d}),
\end{aligned}
$$

where $\Sigma_F = \text{diag}(\sigma_x^2, \sigma_x^2, \sigma_\theta^2)$ and the relationship between state S_t and image encoding Y_t may be parameterized by another neural network G_θ. We consider here a simple linear model of the dynamics

$$
F(S_t, a_t) = \begin{bmatrix} X_t^{(1)} + v_t^{(1)} \cos(\gamma_t) + v_t^{(2)} \sin(\gamma_t) \\ X_t^{(2)} + v_t^{(1)} \sin(\gamma_t) - v_t^{(2)} \cos(\gamma_t) \\ \gamma_t + \omega_t \end{bmatrix}.
$$

D_θ denotes a decoder neural network, mapping the encoding back to the original image. E_θ, G_θ and D_θ are trained using a loss function consisting of the PF-estimated log-likelihood $\hat{\mathcal{L}}_{\text{PF}}$; PF-based mean squared error (MSE), $\hat{\mathcal{L}}_{\text{MSE}}$; and auto-encoder loss, $\hat{\mathcal{L}}_{\text{AE}}$, given per-batch as in (Wen et al., 2020):

$$
\hat{\mathcal{L}}_{\text{MSE}} \coloneqq \frac{1}{T} \sum_{t=1}^{T} ||X_t^* - \sum_{i=1}^{N} w_t^i X_t^i||^2, \quad \hat{\mathcal{L}}_{\text{PF}} \coloneqq -\frac{1}{T} \hat{\ell}(\theta),
$$

$$
\hat{\mathcal{L}}_{\text{AE}} \coloneqq \sum_{t=1}^{T} ||D_\theta(E_\theta(O_t)) - O_t||^2,
$$

where $X_t^\star$ are the true states available from training data and $\sum_{i=1}^N w_t^i X_t^i$ are the PF estimates of $\mathbb{E}[X_t|y_{1:t}]$. The auto-encoder / reconstruction loss $\hat{\mathcal{L}}_{\text{AE}}$ ensures the encoder is informative and prevents the case whereby networks G_θ, E_θ map to a constant. The PF-based loss terms $\hat{\mathcal{L}}_{\text{MSE}}$ and $\hat{\mathcal{L}}_{\text{PF}}$ are not differentiable w.r.t. θ under traditional resampling schemes.

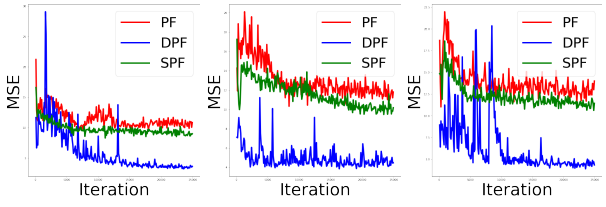
We use the setup from (Jonschkowski et al., 2018) with data from DeepMind Lab (Beattie et al., 2016). This consists of

Table 3. MSE and ± Standard Deviation evaluated on Test Data: Lower is better

	MAZE 1	MAZE 2	MAZE 3
DPF	$\mathbf{3.55}_{\pm \mathbf{0.20}}$	$\mathbf{4.65}_{\pm 0.50}$	$\mathbf{4.44}_{\pm \mathbf{0.26}}$
PF	$10.71_{\pm 0.45}$	$11.86_{\pm 0.57}$	$12.88_{\pm 0.65}$
SPF	$9.14_{\pm 0.39}$	$10.12_{\pm \mathbf{0.40}}$	$11.42_{\pm 0.37}$

3 maze layouts of varying sizes. We have access to 'true' trajectories of length $1,000$ steps for each maze. Each step has an associated state, action and observation image, as described above. The visual observation O_t consists of 32×32 RGB pixel images, compressed to 24×24, as shown in Figure 2. Random, noisy subsets of fixed length are sampled at each training iteration. To illustrate the benefits of our proposed method, we select the random subsets to be of length 50 as opposed to length 20 as chosen in (Jonschkowski et al., 2018). Training details in terms of learning rates, number of training steps and neural network architectures for E_θ, G_θ and D_θ are given in the Appendices.

We compare our method, DPF, to regular PF used in (Maddison et al., 2017) and Soft PF (SPF) used in (Karkus et al., 2018; Ma et al., 2020a;b), whereby the soft resampling is used with $\alpha = 0.1$. As most of the computational complexity arises from neural network operations, DPF is of similar overall computational cost to SPF and PF. As shown in Table 3 and Figure 3, DPF significantly outperforms previously considered PF methods in this experiment. The observation model becomes increasingly important for longer sequences due to resampling and weighting operations. Indeed, as shown in Figure 4, the error is small for both models at the start of the sequence, however the error at later stages in the sequence is visibly smaller for the model trained using DPF.

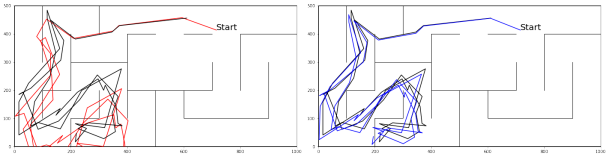


(a) Maze 1 (b) Maze 2 (c) Maze 3

Figure 3. MSE of PF (red), SPF (green) and DPF (blue) estimates, evaluated on test data during training.

6. Discussion

This paper introduces the first principled, fully differentiable PF (DPF) which can use general proposal distributions. It provides a differentiable estimate of the log-likelihood function and more generally differentiable estimates of PF-based losses. This permits parameter inference in state-space models and proposal distributions, using end-to-end gradient based optimization. This also allows the use of PF routines in general differentiable programming pipelines, in particular as a differentiable sampling method for inference in probabilistic programming languages (Dillon et al., 2017; Ge et al., 2018; van de Meent et al., 2018).



(a) Standard PF (b) Differentiable PF

Figure 4. Illustrative Example: PF estimate of path compared to true path (black) on a single 50-step trajectory from test data.

For a given number of particles N, existing PF methods ignoring resampling gradient terms have computational complexity $O(N)$. Training with these resampling schemes however is unreliable and performance cannot be improved by increasing N as gradient estimates are inconsistent and the limiting bias can be significant. DPF has complexity $O(N^2)$ during training. However, this cost is dwarfed when training large neural networks. Additionally, once the model is trained, standard PF may be ran at complexity $O(N)$. The benefits of DPF are confirmed by our experimental results where it was shown to outperform existing techniques, even when an equivalent computational budget was used. Moreover, recent techniques have been proposed to speed up the Sinkhorn algorithm (Altschuler et al., 2019; Scetbon & Cuturi, 2020) at the core of DPF and could potentially be used here to reduce its complexity.

Regularization parameter ϵ was not fine-tuned in our experiments. In future work, it would be interesting to obtain sharper quantitative bounds on DPF to propose principled guidelines on choosing ϵ, further improving its performance. Finally, we have focused on the use of the differentiable ensemble transform to obtain a differentiable resampling scheme. However, alternative OT approaches could also be proposed such as a differentiable version of the second order ET presented in (Acevedo et al., 2017), techniques based on point cloud optimization (Cuturi & Doucet, 2014; Peyré & Cuturi, 2019) relying on the Sinkhorn divergence (Genevay et al., 2018) or the sliced-Wasserstein metric. Alternative non-entropic regularizations, such as the recently proposed Gaussian smoothed OT (Goldfeld & Greenewald, 2020), could also lead to DPFs of interest.

Acknowledgments

Adrien Corenflos was supported by the Academy of Finland (projects 321900 and 321891). Arnaud Doucet is supported by the EPSRC CoSInES (COmputational Sta-

tistical INference for Engineering and Security) grant EP/R034710/1, James Thornton by the OxWaSP CDT through grant EP/L016710/1. Computing resources were provided through the Google Cloud Research Credits Programme. The authors thank Valentin De Bortoli and Yuyang Shi for their comments and Laurence Aitchison for bringing our attention to reference (Aitchison, 2019).

References

Acevedo, W., de Wiljes, J., and Reich, S. Second-order accurate ensemble transform particle filters. *SIAM Journal on Scientific Computing*, 39(5):A1834–A1850, 2017.

Aitchison, L. Tensor Monte Carlo: particle methods for the GPU era. *Advances in Neural Information Processing Systems*, 2019.

Altschuler, J., Niles-Weed, J., and Rigollet, P. Near-linear time approximation algorithms for optimal transport via Sinkhorn iteration. In *Advances in Neural Information Processing Systems*, pp. 1964–1974, 2017.

Altschuler, J., Bach, F., Rudi, A., and Niles-Weed, J. Massively scalable Sinkhorn distances via the Nyström method. In *Advances in Neural Information Processing Systems*, pp. 4429–4439, 2019.

Archer, E., Park, I. M., Buesing, L., Cunningham, J., and Paninski, L. Black box variational inference for state space models. *arXiv preprint arXiv:1511.07367*, 2015.

Beattie, C., Leibo, J. Z., Teplyashin, D., Ward, T., Wainwright, M., Küttler, H., Lefrancq, A., Green, S., Valdés, V., Sadik, A., Schrittwieser, J., Anderson, K., York, S., Cant, M., Cain, A., Bolton, A., Gaffney, S., King, H., Hassabis, D., Legg, S., and Petersen, S. DeepMind Lab, 2016.

Bertsimas, D. and Tsitsiklis, J. N. *Introduction to Linear Optimization*. Athena Scientific Belmont, MA, 1997.

Boulanger-Lewandowski, N., Bengio, Y., and Vincent, P. Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. In *International Conference on Machine Learning*, pp. 1881–1888, 2012.

Burda, Y., Grosse, R. B., and Salakhutdinov, R. Importance weighted autoencoders. In *International Conference on Learning Representations*, 2016.

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

Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A. C., and Bengio, Y. A recurrent latent variable model for sequential data. In *Advances in Neural Information Processing Systems*, pp. 2980–2988, 2015.

Cuturi, M. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, pp. 2292–2300, 2013.

Cuturi, M. and Doucet, A. Fast computation of Wasserstein barycenters. In *International Conference on Machine Learning*, pp. 685–693, 2014.

DeJong, D. N., Liesenfeld, R., Moura, G. V., Richard, J.-F., and Dharmarajan, H. Efficient likelihood evaluation of state-space representations. *Review of Economic Studies*, 80(2):538–567, 2013.

Del Moral, P. *Feynman-Kac Formulae*. Springer, 2004.

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

Dillon, J. V., Langmore, I., Tran, D., Brevdo, E., Vasudevan, S., Moore, D., Patton, B., Alemi, A., Hoffman, M., and Saurous, R. A. Tensorflow distributions. *arXiv preprint arXiv:1711.10604*, 2017.

Douc, R., Moulines, E., and Stoffer, D. *Nonlinear Time Series: Theory, Methods and Applications with R Examples*. CRC press, 2014.

Doucet, A. and Johansen, A. M. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of Nonlinear Filtering*, 12:656–704, 2009.

Doucet, A. and Lee, A. Sequential Monte Carlo methods. *Handbook of Graphical Models*, pp. 165–189, 2018.

Feydy, J., Séjourné, T., Vialard, F.-X., Amari, S.-I., Trouvé, A., and Peyré, G. Interpolating between optimal transport and MMD using Sinkhorn divergences. In *International Conference on Artificial Intelligence and Statistics*, 2019.

Finke, A., Doucet, A., and Johansen, A. M. On embedded hidden Markov models and particle Markov chain Monte Carlo methods. *arXiv preprint arXiv:1610.08962*, 2016.

Flamary, R., Cuturi, M., Courty, N., and Rakotomamonjy, A. Wasserstein discriminant analysis. *Machine Learning*, 107(12):1923–1945, 2018.

Fournier, N. and Guillin, A. On the rate of convergence in Wasserstein distance of the empirical measure. *Probability Theory and Related Fields*, 162(3):707–738, 2015.

Ge, H., Xu, K. X., and Ghahramani, Z. Turing: A language for flexible probabilistic inference. In *International Conference on Artificial Intelligence and Statistics*, pp. 1682–1690, 2018.

Genevay, A., Peyré, G., and Cuturi, M. Learning generative models with Sinkhorn divergences. In *International Conference on Artificial Intelligence and Statistics*, pp. 1608–1617, 2018.

Goldfeld, Z. and Greenewald, K. Gaussian-smooth optimal transport: Metric structure and statistical efficiency. *arXiv preprint arXiv 2001.09206*, 2020.

Graves, A. Stochastic backpropagation through mixture density distributions. *arXiv preprint arXiv:1607.05690*, 2016.

Hirt, M. and Dellaportas, P. Scalable Bayesian learning for state space models using variational inference with SMC samplers. In *International Conference on Artificial Intelligence and Statistics*, pp. 76–86, 2019.

Jonschkowski, R., Rastogi, D., and Brock, O. Differentiable particle filters: End-to-end learning with algorithmic priors. In *Proceedings of Robotics: Science and Systems*, 2018.

Kantas, N., Doucet, A., Singh, S. S., Maciejowski, J., and Chopin, N. On particle methods for parameter estimation in state-space models. *Statistical Science*, 30(3):328–351, 2015.

Karkus, P., Hsu, D., and Lee, W. S. Particle filter networks with application to visual localization. In *Conference on Robot Learning*, 2018.

Kingma, D. P. and Welling, M. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014.

Kitagawa, G. and Gersch, W. *Smoothness Priors Analysis of Time Series*, volume 116. Springer Science & Business Media, 1996.

Klaas, M., De Freitas, N., and Doucet, A. Toward practical N^2 Monte Carlo: the marginal particle filter. *Uncertainty in Artificial Intelligence*, 2005.

Kloss, A., Martius, G., and Bohg, J. How to train your differentiable filter. *arXiv preprint arXiv:2012.14313*, 2020.

Krishnan, R. G., Shalit, U., and Sontag, D. Structured inference networks for nonlinear state space models. In *AAAI Conference on Artificial Intelligence*, pp. 2101–2109, 2017.

Le, T. A., Igl, M., Rainforth, T., Jin, T., and Wood, F. Auto-encoding sequential Monte Carlo. In *International Conference on Learning Representations*, 2018.

Lee, A. Towards smooth particle filters for likelihood estimation with multivariate latent variables. Master's thesis, University of British Columbia, 2008.

Li, W. and Nochetto, R. H. Quantitative stability and error estimates for optimal transport plans. *IMA Journal of Numerical Analysis*, 2021.

Lindsten, F. and Schön, T. B. Backward simulation methods for Monte Carlo statistical inference. *Foundations and Trends® in Machine Learning*, 6(1):1–143, 2013.

Ma, X., Karkus, P., Hsu, D., and Lee, W. S. Particle filter recurrent neural networks. In *AAAI Conference on Artificial Intelligence*, 2020a.

Ma, X., Karkus, P., Ye, N., Hsu, D., and Lee, W. S. Discriminative particle filter reinforcement learning for complex partial observations. In *International Conference on Learning Representations*, 2020b.

Maddison, C. J., Lawson, D., Tucker, G., Heess, N., Norouzi, M., Mnih, A., Doucet, A., and Teh, Y. W. Filtering variational objectives. In *Advances in Neural Information Processing Systems*, 2017.

Malik, S. and Pitt, M. K. Particle filters for continuous likelihood evaluation and maximisation. *Journal of Econometrics*, 165(2):190–209, 2011.

Murray, L. M., Jones, E. M., and Parslow, J. On disturbance state-space models and the particle marginal Metropolis–Hastings sampler. *SIAM/ASA Journal on Uncertainty Quantification*, 1(1):494–521, 2013.

Myers, A., Thiery, A. H., Wang, K., and Bui-Thanh, T. Sequential ensemble transform for Bayesian inverse problems. *Journal of Computational Physics*, 427:110055, 2021.

Naesseth, C. A., Linderman, S. W., Ranganath, R., and Blei, D. M. Variational sequential Monte Carlo. In *International Conference on Artificial Intelligence and Statistics*, 2018.

Peyré, G. and Cuturi, M. Computational optimal transport. *Foundations and Trends® in Machine Learning*, 11(5-6): 355–607, 2019.

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

Rangapuram, S. S., Seeger, M. W., Gasthaus, J., Stella, L., Wang, Y., and Januschowski, T. Deep state space models for time series forecasting. In *Advances in Neural Information Processing Systems*, pp. 7785–7794, 2018.

Reich, S. A nonparametric ensemble transform method for Bayesian inference. *SIAM Journal on Scientific Computing*, 35(4):A2013–A2024, 2013.

Scetbon, M. and Cuturi, M. Linear time Sinkhorn divergences using positive features. In *Advances in Neural Information Processing Systems*, 2020.

Seguy, V., Damodaran, B. B., Flamary, R., Courty, N., Rolet, A., and Blondel, M. Large-scale optimal transport and mapping estimation. In *International Conference on Learning Representations*, 2018.

Thrun, S., Burgard, W., and Fox, D. *Probabilistic Robotics*. MIT Press, 2005.

van de Meent, J.-W., Paige, B., Hongseok, Y., and Wood, F. An introduction to probabilistic programming. *arXiv preprint arXiv:1809.10756*, 2018.

Villani, C. *Optimal Transport: Old and New*, volume 338. Springer Science & Business Media, 2008.

Weed, J. An explicit analysis of the entropic penalty in linear programming. In *Proceedings of the 31st Conference On Learning Theory*, 2018.

Wen, H., Chen, X., Papagiannis, G., Hu, C., and Li, Y. End-to-end semi-supervised learning for differentiable particle filters. *arXiv preprint arXiv:2011.05748*, 2020.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3-4):229–256, 1992.

Zhu, M., Murphy, K., and Jonschkowski, R. Towards differentiable resampling. *arXiv preprint arXiv:2004.11938*, 2020.

A. Proof of Proposition 4.1

A particle filter with multinomial resampling is defined by the following joint distribution

$$\overline{q}_{\theta,\phi}(x_{1:T}^{1:N}, a_{1:T-1}^{1:N}) = \prod_{i=1}^{N} q_\phi\left(x_1^i\right) \prod_{t=2}^{T}\prod_{i=1}^{N} w_{t-1}^{a_{t-1}^i} q_\phi\left(x_t^i | x_{t-1}^{a_{t-1}^i}\right)$$

where $a_{t-1}^i \in \{1, ..., N\}$ is the ancestral index of particle x_t^i and

$$\omega_{\theta,\phi}(x_1, y_1) = \frac{p_\theta(x_1, y_1)}{q_\phi(x_1)}, \quad \omega_{\theta,\phi}(x_{t-1}, x_t, y_t) = \frac{p_\theta(x_t, y_t | x_{t-1})}{q_\phi\left(x_t | x_{t-1}\right)}.$$

Finally, we have $w_t^i \propto \omega_{\theta,\phi}(x_{t-1}^{a_{t-1}^i}, x_t^i, y_t)$, $\sum_{i=1}^N w_t^i = 1$. We do not emphasize notationally that the weights $w_{t-1}^{a_{t-1}^i}$ are θ, ϕ and observations dependent.

The ELBO is given by

$$\ell^{\text{ELBO}}(\theta, \phi) = \mathbb{E}_{\overline{q}_{\theta,\phi}}\left[\log \widehat{p}_\theta(y_{1:T})\right] = \mathbb{E}_{\overline{q}_{\theta,\phi}}\left[\log\left(\frac{1}{N}\sum_{i=1}^{N}\omega_{\theta,\phi}(X_1^i, y_1)\right) + \sum_{t=2}^{T}\log\left(\frac{1}{N}\sum_{i=1}^{N}\omega_{\theta,\phi}(X_{t-1}^{A_{t-1}^i}, X_t^i, y_t)\right)\right].$$

We now compute $\nabla_\theta \ell^{\text{ELBO}}(\theta, \phi)$. We assume from now on that the regularity conditions allowing us to swap the expectation and differentiation operators are satisfied as in (Maddison et al., 2017; Le et al., 2018; Naesseth et al., 2018). We can split the gradient using the product rule and apply the log-derivative trick:

$$\begin{aligned}
\nabla_\theta \ell^{\text{ELBO}}(\theta, \phi) =& \mathbb{E}_{\overline{q}_{\theta,\phi}}\left[\nabla_\theta \log \widehat{p}_\theta(y_{1:T})\right] + \mathbb{E}_{\overline{q}_{\theta,\phi}}\left[\log \widehat{p}_\theta(y_{1:T}) \nabla_\theta \log \overline{q}_{\theta,\phi}(X_{1:T}^{1:N}, A_{1:T-1}^{1:N})\right] \\
=& \mathbb{E}_{\overline{q}_{\theta,\phi}}\left[\nabla_\theta \log\left(\frac{1}{N}\sum_{i=1}^{N}\omega_{\theta,\phi}(X_1^i, y_1)\right) + \sum_{t=2}^{T}\nabla_\theta \log\left(\frac{1}{N}\sum_{i=1}^{N}\omega_{\theta,\phi}(X_{t-1}^{A_{t-1}^i}, X_t^i, y_t)\right)\right] \quad (18) \\
&+ \mathbb{E}_{\overline{q}_{\theta,\phi}}\left[\log \widehat{p}_\theta(y_{1:T})\left\{\sum_{t=2}^{T}\sum_{i=1}^{N}\nabla_\theta \log w_{t-1}^{A_{t-1}^i}\right\}\right] \quad (19)
\end{aligned}$$

For the first part of the ELBO gradient (18), we have

$$\nabla_\theta \log\left(\frac{1}{N}\sum_{i=1}^{N}\omega_{\theta,\phi}(X_1^i), y_1\right) = \sum_{i=1}^{N} w_1^i \nabla_\theta \log w_{\theta,\phi}(X_1^i, y_1) = \sum_{i=1}^{N} w_1^i \nabla_\theta \log p_\theta(X_1^i, y_1)$$

and

$$\nabla_\theta \log\left(\frac{1}{N}\sum_{i=1}^{N}\omega_{\theta,\phi}(X_{t-1}^{A_{t-1}^i}, X_t^i, y_t)\right) = \sum_{i=1}^{N} w_t^i \nabla_\theta \log \omega_{\theta,\phi}(X_{t-1}^{A_{t-1}^i}, X_t^i, y_t) = \sum_{i=1}^{N} w_t^i \nabla_\theta \log p_\theta(X_t^i, y_t | X_{t-1}^{A_{t-1}^i}).$$

This gives

$$\begin{aligned}
\nabla_\theta \ell^{\text{ELBO}}(\theta, \phi) =& \mathbb{E}_{\overline{q}_{\theta,\phi}}\left[\sum_{i=1}^{N} w_1^i \nabla_\theta \log p_\theta(X_1^i, y_1) + \sum_{t=2}^{T}\sum_{i=1}^{N} w_t^i \nabla_\theta \log p_\theta(X_t^i, y_t | X_{t-1}^{A_{t-1}^i})\right] \quad (20) \\
&+ \mathbb{E}_{\overline{q}_{\theta,\phi}}\left[\log \widehat{p}_\theta(y_{1:T})\left\{\sum_{t=2}^{T}\sum_{i=1}^{N}\nabla_\theta \log w_{t-1}^{A_{t-1}^i}\right\}\right]. \quad (21)
\end{aligned}$$

When we ignore the gradient terms due to resampling corresponding to (21) as proposed in (Naesseth et al., 2018; Le et al., 2018; Maddison et al., 2017; Hirt & Dellaportas, 2019), we only use an unbiased estimate of the first term (20), i.e.

$$\hat{\nabla}_\theta \ell^{\text{ELBO}}(\theta, \phi) := \sum_{i=1}^{N} w_1^i \nabla_\theta \log p_\theta(X_1^i, y_1) + \sum_{t=2}^{T}\sum_{i=1}^{N} w_t^i \nabla_\theta \log p_\theta(X_t^i, y_t | X_{t-1}^{A_{t-1}^i}), \quad \text{where } (X_{1:T}^{1:N}, A_{1:T-1}^{1:N}) \sim \overline{q}_{\theta,\phi}(\cdot). \quad (22)$$

Now we assume that the mild assumptions ensuring almost sure convergence of the PF estimates are satisfied (see e.g. (Del Moral, 2004)). Under these assumptions, the estimator (22) converges almost surely as $N \to \infty$ towards

$$\int \nabla_\theta \log p_\theta(x_1, y_1) p_\theta(x_1|y_1) \mathrm{d}x_1 + \sum_{t=2}^{T} \int \nabla_\theta \log p_\theta(x_t, y_t|x_{t-1}) p_\theta(x_{t-1:t}|y_{1:t-1}) \mathrm{d}x_{t-1:t}. \tag{23}$$

Under an additional uniform integrability condition on $\hat{\nabla}_\theta \ell^{\mathrm{ELBO}}(\theta, \phi)$, we thus have that $\mathbb{E}_{\bar{q}_{\theta,\phi}}[\hat{\nabla}_\theta \ell^{\mathrm{ELBO}}(\theta, \phi)]$ converges towards (23). We recall that the true score is given by Fisher's identity and satisfies

$$\int \nabla_\theta \log p_\theta(x_1, y_1) p_\theta(x_1|y_{1:T}) \mathrm{d}x_1 + \sum_{t=2}^{T} \int \nabla_\theta \log p_\theta(x_t, y_t|x_{t-1}) p_\theta(x_{t-1:t}|y_{1:T}) \mathrm{d}x_{t-1:t}.$$

This concludes the proof of Proposition 4.1.

B. Notation and Assumptions

B.1. Filtering Notation

Recall $\mathcal{X} = \mathbb{R}^{d_x}$, denote the Borel sets of $\mathcal{X}$ by $\mathcal{B}(\mathcal{X})$ and $\mathcal{P}(\mathcal{X})$ the set of Borel probability measures on $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$. In an abuse of notation, we shall use the same notation for a probability measure and its density w.r.t. Lebesgue measure; i.e. $\nu(\mathrm{d}x) = \nu(x)\mathrm{d}x$. We also use the standard notation $\nu(\psi) = \int \psi(x)\nu(x)\mathrm{d}x$ for any test function ψ. In the interest of notational clarity, we will remove subscript θ, ϕ where unnecessary in further workings.

We denote $\{\alpha^{(t)}\}_{t\geq 0}$ the predictive distributions where $\alpha^{(t)}(x_t) = p(x_t|y_{1:t-1})$ for $t > 1$ and $\alpha^{(1)}(x_1) = \mu(x_1)$ while $\{\beta^{(t)}\}_{t\geq 1}$ denotes the filtering distributions; i.e. $\beta^{(t)}(x_t) = p(x_t|y_{1:t})$ for $t \geq 1$.

Using this notation, we have

$$\alpha^{(t)}(\psi) = \int \psi(x_t) f(x_t|x_{t-1}) \beta^{(t-1)}(x_{t-1}) \mathrm{d}x_{t-1} \mathrm{d}x_t \coloneqq \beta^{(t-1)} f(\psi), \tag{24}$$

$$\beta^{(t)}(\psi) = \frac{\alpha^{(t)}(g(y_t|\cdot)\psi)}{\alpha^{(t)}(g(y_t|\cdot))} = \frac{\beta^{(t-1)}(f(g(y_t|\cdot)\psi))}{\beta^{(t-1)}(f(g(y_t|\cdot)))}. \tag{25}$$

More generally, for a proposal distribution $q(x_t|x_{t-1}, y_t) \neq f(x_t|x_{t-1})$ with parameter $\phi \neq \theta$, the following recursion holds

$$\beta^{(t)}(\psi) = \frac{\beta^{(t-1)}(q(\omega_t\, \psi))}{\beta^{(t-1)}(q(\omega_t))} \tag{26}$$

$$\omega_t(x_{t-1}, x_t) \coloneqq \omega(x_{t-1}, x_t, y_t) = \frac{g(y_t|x_t) f(x_t|x_{t-1})}{q(x_t|x_{t-1}, y_t)}. \tag{27}$$

To simplify the presentation, we will present the analysis in the scenario where $\phi = \theta$ and $q(x_t|x_{t-1}, y_t) = f(x_t|x_{t-1})$ so we will analyze (24) for which $\omega_t(x_{t-1}, x_t) = g(y_t|x_t)$. In this case, the particle approximations of μ is denoted μ_N and for $t > 1$, $\alpha^{(t)}$ and $\beta^{(t)}$ are given by the random measures

$$\alpha_N^{(t)}(\psi) = \frac{1}{N}\sum_{i=1}^{N} \psi(X_t^i), \quad \beta_N^{(t)}(\psi) = \sum_{i=1}^{N} w_t^i \psi(X_t^i), \quad \tilde{\beta}_N^{(t)}(\psi) = \frac{1}{N}\sum_{i=1}^{N} \psi(\tilde{X}_t^i), \tag{28}$$

where $w_t^i \propto g(y_t|X_t^i)$ with $\sum_{i=1}^N w_t^i = 1$ and particles are drawn from $X_t^i \sim f(\cdot|\tilde{X}_{t-1}^i)$.

Here $\beta_N^{(t)}$ denotes the weighted particle approximation of $\beta^{(t)}$ while $\tilde{\beta}_N^{(t)}$ is the uniformly weighted approximation obtained after the DET transformation described in Section 3.2.

B.2. Optimal Transport Notation

Recall from Section 2.1, $\mathcal{P}_t^{\mathrm{OT}}$ denotes a transport between $\alpha^{(t)}$ and $\beta^{(t)}$ with accompanying map $\mathbf{T}^{(t)}$. $\mathcal{P}_t^{\mathrm{OT},N}$ denotes an optimal transport between particle approximations $\alpha_N^{(t)}$ and $\beta_N^{(t)}$ with corresponding transport matrix, $\mathbf{P}^{\mathrm{OT}}$ with i,j entry $p_{i,j}^{\mathrm{OT}}$. To simplify notation, we remove script t when not needed.

Similarly from Section 3.1, $\mathcal{P}_\epsilon^{\mathrm{OT,N}}$ denotes the regularized transport between $\alpha_N^{(t)}$ and $\beta_N^{(t)}$ with accompanying matrix $\mathbf{P}_\epsilon^{\mathrm{OT}}$ with i,j entry $p_{\epsilon,i,j}^{\mathrm{OT}}$. Recall $\tilde{\beta}_N^{(t)} = \frac{1}{N}\sum_{i=1}^N \delta_{\tilde{X}^i}$ is the uniformly weighted particle approximation for $\beta^{(t)}$ under the DET, i.e. $\tilde{X}^i = \mathbf{T}_{N,\epsilon}^{(t)}(X^i) = \int y \mathcal{P}_\epsilon^{\mathrm{OT},N}(\mathrm{d}y|x^i)$. Note that $\tilde{X}_{N,\epsilon}^i$ will be used where necessary to avoid ambiguity when comparing to other resampling schemes.

Recall also for $p > 0$:

$$\mathcal{W}_p^p(\alpha,\beta) = \min_{\mathcal{P}\in\mathcal{U}(\alpha,\beta)} \mathbb{E}_{(U,V)\sim\mathcal{P}}\left[||U-V||^p\right] \tag{29}$$

where $\mathcal{U}(\alpha,\beta)$ is the collection of couplings with marginals α and β.

B.3. Assumptions

Our results will rely on the following four assumptions.

Assumption B.1. $\mathcal{X} \subset \mathbb{R}^d$ is a compact subset with diameter

$$\mathfrak{d} := \sup_{x,y\in\mathcal{X}} |x-y|.$$

Assumption B.2. There exists $\kappa \in (0,1)$ such that for any two probability measures π, ρ on $\mathcal{X}$

$$\mathcal{W}_k(\pi f, \rho f) \leq \kappa \mathcal{W}_k(\pi,\rho), \qquad k = 1,2.$$

Assumption B.3. The weight function $\omega^{(t)} : \mathcal{X} \to [\Delta, \Delta^{-1}]$ is 1-Lipschitz for all t.

Assumption B.4. There exists a $\lambda > 0$, such that for all $t \geq 0$ the unique optimal transport plan between $\alpha^{(t)}$ and $\beta^{(t)}$ is given by a deterministic, λ-Lipschitz map $\mathbf{T}^{(t)}$.

C. Auxiliary Results and Proof of Proposition 4.2

We start by establishing a couple of key auxiliary results which will be then used subsequently to establish Proposition 4.2.

C.1. Auxiliary Results

As per section 2.1, let $\mathcal{S}(\alpha_N, \beta_N)$ denote the collection of coupling matrices between $\alpha_N = \sum_{i=1}^N a_i \delta_{Y^i}$ with $a_i > 0$ and $\beta_N = \sum_{i=1}^N b_i \delta_{X^i}$. We also denote entropy by H where $H(\mathbf{P}) = \sum_{i,j} p_{i,j}\log(1/p_{i,j})$ for $\mathbf{P} = (p_{i,j})_{i,j} \in \mathcal{S}(\alpha_N,\beta_N)$.

Lemma C.1. *The entropic radius, R_H, of simplex $\mathcal{U}(\alpha_N,\beta_N)$ may be bounded above as follows*

$$R_H := \max_{\mathbf{P}_1,\mathbf{P}_2\in\mathcal{S}(\alpha_N,\beta_N)} H(\mathbf{P}_1) - H(\mathbf{P}_2) \leq 2\log(N)$$

Proof. Notice that $-H(\mathbf{P})$ is convex, so that $H(\mathbf{P})$ is concave.

$$\begin{aligned}\sum_{i,j} p_{i,j}\log\left(\frac{1}{p_{i,j}}\right) &= N^2 \sum_{i,j} \frac{1}{N^2} p_{i,j}\log\left(\frac{1}{p_{i,j}}\right) \\ &\leq N^2 H\left(\frac{1}{N^2}\sum_{i,j} p_{i,j}\right) = N^2 H(1/N^2) = N^2\frac{1}{N^2}\log\big(N^2\big) = 2\log(N).\end{aligned}$$

In addition since $p_{i,j} \leq 1$ for all i, j, we have that $H(\mathbf{P}) \geq 0$ and therefore we can bound

$$R_H = \max_{P_1, P_2 \in \mathcal{P}} H(\mathbf{P}_1) - H(\mathbf{P}_2) \leq \max_{\mathbf{P}_1 \in \mathcal{S}(\alpha_N, \beta_N)} H(\mathbf{P}_1) \leq 2\log(N).$$

□

Lemma C.2. *Let $\mathcal{X} \subset \mathbb{R}^d$ be compact with diameter $\mathfrak{d} > 0$. Suppose we are given two probability measures α, β on $\mathcal{X}$ with a unique deterministic, λ-Lipschitz optimal transport map $\mathbf{T}$ while $\alpha_N = \sum_{i=1}^N a_i \delta_{Y^i}$ with $a_i > 0$ and $\beta_N = \sum_{i=1}^N b_i \delta_{X^i}$. We write $\mathcal{P}^{\mathrm{OT},N}$, resp. $\mathcal{P}_\epsilon^{\mathrm{OT},N}$, for an optimal coupling between α_N and β_N, resp. the ϵ-regularized optimal transport plan, between α_N and β_N. Then*

$$\left[\int ||y - \mathbf{T}(x)||^2 \mathcal{P}_\epsilon^{\mathrm{OT},N}(\mathrm{d}x, \mathrm{d}y)\right]^{\frac{1}{2}} \leq 2\lambda^{1/2}\mathcal{E}^{1/2}\left[\mathfrak{d}^{1/2} + \mathcal{E}\right]^{1/2} + \max\{\lambda, 1\}\left[\mathcal{W}_2(\alpha_N, \alpha) + \mathcal{W}_2(\beta_N, \beta)\right],$$

where

$$\mathcal{E} \coloneqq \mathcal{E}(N, \epsilon, \alpha, \beta) \coloneqq \mathcal{W}_2(\alpha_N, \alpha) + \mathcal{W}_2(\beta_N, \beta) + \sqrt{2\epsilon\log(N)}.$$

Proof. From Corollary 3.8 from (Li & Nochetto, 2021)

$$\left[\int \|\mathbf{T}(x) - y\|^2 \mathcal{P}_\epsilon^{\mathrm{OT},N}(\mathrm{d}x, \mathrm{d}y)\right]^{1/2} \leq 2\lambda^{1/2}\sqrt{\tilde{e}_{N,\epsilon}}\left[\mathcal{W}_2(\alpha, \beta) + \tilde{e}_{N,\epsilon}\right]^{1/2} + \lambda\mathcal{W}_2(\alpha_N, \alpha) + \mathcal{W}_2(\beta_N, \beta),$$

where λ is the Lipschitz constant of the optimal transport map $\mathbf{T}$ sending α to β, and

$$\tilde{e}_{N,\epsilon} \coloneqq \mathcal{W}_2(\alpha_N, \alpha) + \mathcal{W}_2(\beta_N, \beta) + \left[\int \|x - y\|^2 \mathcal{P}_\epsilon^{\mathrm{OT},N}(\mathrm{d}x, \mathrm{d}y)\right]^{1/2} - \mathcal{W}_2(\alpha_N, \beta_N). \tag{30}$$

From Proposition 4 of (Weed, 2018),

$$\sum_{i,j=1,\ldots,N} p_{\epsilon,i,j}^{\mathrm{OT}} |Y_i - X_j|^2 - \mathcal{W}_2^2(\alpha_N, \beta_N) \leq \epsilon R_H,$$

where R_H is the entropic radius as defined in Lemma C.1.

By Lemma C.1 we therefore have that

$$\int \|x - y\|^2 \mathcal{P}_\epsilon^{\mathrm{OT},N}(\mathrm{d}x, \mathrm{d}y) - \mathcal{W}_2^2(\alpha_N, \beta_N) \leq 2\epsilon\log(N).$$

Since $x \mapsto \sqrt{x}$ is sub-additive, for $r, s > 0$ we have that $\sqrt{r} - \sqrt{s} \leq \sqrt{r - s}$, whence

$$\left[\int \|x - y\|^2 \mathcal{P}_\epsilon^{\mathrm{OT},N}(\mathrm{d}x, \mathrm{d}y)\right]^{1/2} - \mathcal{W}_2(\alpha_N, \beta_N) \leq \sqrt{2\epsilon\log N}.$$

We thus have

$$\tilde{e}_{N,\epsilon} \leq \mathcal{W}_2(\alpha_N, \alpha) + \mathcal{W}_2(\beta_N, \beta) + \sqrt{2\epsilon\log(N)}.$$

In addition, by Assumption B.1 we have that $\mathcal{W}_2(\alpha, \beta) \leq \mathfrak{d}^{1/2}$ and the result follows. □

C.2. Proof of Proposition 4.2

Proof of Proposition 4.2. By definition, we have $\tilde{\beta}_N(\mathrm{d}\tilde{x}) = \int \alpha_N(\mathrm{d}x)\delta_{\mathbf{T}_{N,\epsilon}(x)}(\mathrm{d}\tilde{x})$ with $\mathbf{T}_{N,\epsilon}(x) \coloneqq \int \tilde{x}\mathcal{P}_\epsilon^{\mathrm{OT,N}}(\mathrm{d}\tilde{x}|x)$ while, as $\mathcal{P}_\epsilon^{\mathrm{OT},N}$ belongs to $\mathcal{U}(\alpha_N, \beta_N)$, we also have $\beta_N(\mathrm{d}\tilde{x}) = \int \alpha_N(\mathrm{d}x)\mathcal{P}_\epsilon^{\mathrm{OT,N}}(\mathrm{d}\tilde{x}|x)$. We then have for any 1-Lipschitz

function

$$\begin{aligned}\left|\beta_N(\psi) - \tilde{\beta}_N(\psi)\right| &= \left|\int \left[\int (\psi(\tilde{x}) - \psi(\mathbf{T}_{N,\epsilon}(x)))\mathcal{P}_\epsilon^{\mathrm{OT,N}}(\mathrm{d}\tilde{x}|x)\right] \alpha_N(\mathrm{d}x)\right| \\ &\leq \iint |\psi(\tilde{x}) - \psi(\mathbf{T}_{N,\epsilon}(x))| \, \alpha_N(\mathrm{d}x)\mathcal{P}_\epsilon^{\mathrm{OT,N}}(\mathrm{d}\tilde{x}|x) \\ &\leq \iint ||\tilde{x} - \mathbf{T}_{N,\epsilon}(x)||\mathcal{P}_\epsilon^{\mathrm{OT,N}}(\mathrm{d}x, \mathrm{d}\tilde{x}) \\ &\leq \left(\iint ||\tilde{x} - \mathbf{T}_{N,\epsilon}(x)||^2\mathcal{P}_\epsilon^{\mathrm{OT,N}}(\mathrm{d}x, \mathrm{d}\tilde{x})\right)^{\frac{1}{2}} \\ &\leq \left(\iint ||\tilde{x} - \mathbf{T}(x)||^2\mathcal{P}_\epsilon^{\mathrm{OT,N}}(\mathrm{d}x, \mathrm{d}\tilde{x})\right)^{\frac{1}{2}},\end{aligned}$$

where the final inequality follows from the fact that for any random vector V the mapping $v \mapsto \mathbb{E}[\|V - v\|^2]$ is minimized at $v = \mathbb{E}[V]$. The stated result is then obtained using Lemma C.2. □

D. Proof of Proposition 4.3

For technical reasons, we analyse here a slightly modified PF algorithm where

$$\alpha_N^{(t)} = \frac{1}{N}\sum_{j=1}^N \delta_{X_t^j}, \qquad X_t^j \overset{\text{i.i.d.}}{\sim} \tilde{\beta}_N^{(t-1)} f = \frac{1}{N}\sum_{j=1}^N f\left(\cdot \middle| \tilde{X}_{t-1}^j\right). \tag{31}$$

instead of the standard version where one has

$$\alpha_N^{(t)} = \frac{1}{N}\sum_{j=1}^N \delta_{X_t^j}, \qquad X_t^j \sim f\left(\cdot \middle| \tilde{X}_{t-1}^j\right).$$

This slightly modified version of the bootstrap PF was analyzed for example in (Del Moral & Guionnet, 2001). The analysis does capture the additional error arising from the use of DET instead of resampling. Similar results should hold for the standard PF algorithm. The main technical reason for analysing this modified algorithm is our reliance on Theorem 2 of (Fournier & Guillin, 2015); analysing the standard PF algorithm requires a version of (Fournier & Guillin, 2015) for stratified sampling and will be done in future work.

Proposition D.1. *Suppose that Assumptions B.1, B.2 and B.3 hold. Suppose also that given $\tilde{\beta}_N^{(t-1)}$, $\alpha_N^{(t)}$ is defined through (31). Define the functions*

$$\begin{aligned}\mathcal{F}(x) &\coloneqq x + \sqrt{\mathfrak{d}K_1(\Delta, \mathfrak{d})x} \\ f_d(x) &\coloneqq \begin{cases} x, & d < 4 \\ \frac{x}{\log(2+1/x)}, & d = 4 \\ x^{d/2}, & d > 4. \end{cases} \\ \mathcal{F}_{N,\epsilon,\delta,d}(x) &\coloneqq \mathcal{F}\left(\kappa x + \sqrt{f_d^{-1}\left(\frac{\log(C/\delta)}{cN}\right)}\right), \\ \frac{1}{\mathfrak{d}}\mathfrak{G}_{\epsilon,\delta,N,d}^2(x) &\coloneqq 2\lambda^{1/2}\left[\mathcal{F}_{N,\epsilon,\delta,d}(x) + \sqrt{2\epsilon\log N}\right]^{1/2}\left[\mathfrak{d}^{1/2} + \mathcal{F}_{N,\epsilon,\delta,d}(x) + \sqrt{2\epsilon\log N}\right]^{1/2} \\ &\quad + \lambda\kappa\mathcal{F}_{N,\epsilon,\delta,d}(x) + \max\{\lambda, 1\}\mathcal{F}_{N,\epsilon,\delta,d}(x).\end{aligned} \tag{32}$$

Then for any $\epsilon, \delta > 0$ we have with probability at least $1 - \delta$, over the sampling step in (31), that

$$\mathcal{W}_2\left(\tilde{\beta}_N^{(t)}, \beta^{(t)}\right) \leq \mathfrak{G}_{\epsilon,\delta,N,d}\left[\mathcal{W}_2\left(\tilde{\beta}_N^{(t-1)}, \beta^{(t-1)}\right)\right] \tag{33}$$

In particular if $\mathcal{W}_2(\tilde{\beta}_N^{(t-1)}, \beta^{(t-1)}) \to 0$ and $\epsilon_N = o(1/\log(N))$ as $N \to \infty$ we have that

$$\mathcal{W}_2\big(\tilde{\beta}_N^{(t)}, \beta^{(t)}\big) \to 0,$$

in probability.

Proof of Proposition D.1. To keep notation concise we write for $N \geq 1$

$$\alpha_N \coloneqq \alpha_N^{(t)}, \quad \alpha'_N \coloneqq \tilde{\beta}_N^{(t-1)} f, \quad \beta_N \coloneqq \beta_N^{(t)}, \quad \tilde{\beta}_N \coloneqq \tilde{\beta}_N^{(t-1)}.$$

Controlling $\mathcal{W}_1(\beta_N, \beta)$. Let ψ be 1-Lipschitz. Without loss of generality we may assume that $\psi(0) = 0$ since otherwise we can remove a constant.

$$\begin{aligned}
|\beta_N(\psi) - \beta(\psi)| &= \left| \frac{\alpha_N(\omega\psi)}{\alpha_N(\omega)} - \frac{\alpha(\omega\psi)}{\alpha(\omega)} \right| \\
&\leq \left| \frac{\alpha_N(\omega\psi)}{\alpha_N(\omega)} - \frac{\alpha(\omega\psi)}{\alpha_N(\omega)} \right| + \left| \frac{\alpha(\omega\psi)}{\alpha_N(\omega)} - \frac{\alpha(\omega\psi)}{\alpha(\omega)} \right| \\
&\leq \Delta^{-1} \left| \alpha_N(\omega\psi) - \alpha(\omega\psi) \right| + \Delta^{-2} \alpha(\omega\psi) |\alpha_N(\omega) - \alpha(\omega)|.
\end{aligned}$$

At this stage notice that

$$|(\omega\psi)'| \leq |\omega'\psi| + |\omega\psi'| \leq \|\psi\|_\infty + \|\omega\|_\infty.$$

Notice that

$$|\psi(x)| = |\psi(x) - \psi(0)| \leq |x - 0| \leq \mathfrak{d}.$$

Therefore we have that

$$|(\omega\psi)'| \leq \mathfrak{d} + \Delta^{-1},$$

and thus $\omega\psi$ is $(\mathfrak{d} + \Delta^{-1})$-Lipschitz. It follows that

$$\begin{aligned}
|\beta_N(\psi) - \beta(\psi)| &\leq \Delta^{-1} \left| \alpha_N(\omega\psi) - \alpha(\omega\psi) \right| + \Delta^{-2} \alpha(\omega\psi) |\alpha_N(\omega) - \alpha(\omega)| \\
&\leq \Delta^{-1} (\mathfrak{d} + \Delta^{-1}) \mathcal{W}_1(\alpha_N, \alpha) + \Delta^{-3} \mathfrak{d} \mathcal{W}_1(\alpha_N, \alpha) \\
&=: K_1(\Delta, \mathfrak{d}) \mathcal{W}_1(\alpha_N, \alpha).
\end{aligned}$$

Therefore we have that

$$\mathcal{W}_1(\beta_N, \beta) \leq K_1(\Delta, \mathfrak{d}) \mathcal{W}_1(\alpha_N, \alpha). \tag{34}$$

Notice that using the compactness of the state space we easily get also that

$$\mathcal{W}_2(\beta_N, \beta) \leq \sqrt{\mathfrak{d} \mathcal{W}_1(\beta_N, \beta)} \leq \sqrt{\mathfrak{d} K_1(\Delta, \mathfrak{d}) \mathcal{W}_1(\alpha_N, \alpha)} \leq \sqrt{\mathfrak{d} K_1(\Delta, \mathfrak{d}) \mathcal{W}_2(\alpha_N, \alpha)}, \tag{35}$$

since clearly $\mathcal{W}_1(\rho, \sigma) \leq \mathcal{W}_2(\rho, \sigma)$ for any two probability measures ρ, σ.

Controlling $\mathcal{W}_1(\tilde{\beta}_{N,\epsilon}, \beta)$. Again supposing ψ is 1-Lipschitz, and $\psi(0) = 0$, consider

$$\begin{aligned}
\left| \tilde{\beta}_N(\psi) - \tilde{\beta}(\psi) \right| &= \left| \int \psi(\mathbf{T}_{N,\epsilon}(x)) \alpha_N(\mathrm{d}x) - \int \psi(\mathbf{T}(x)) \alpha(\mathrm{d}x) \right| \\
&\leq \left| \int \psi(\mathbf{T}_{N,\epsilon}(x)) \alpha_N(\mathrm{d}x) - \int \psi(\mathbf{T}(x)) \alpha_N(\mathrm{d}x) \right| \\
&\quad + \left| \int \psi(\mathbf{T}(x)) \alpha_N(\mathrm{d}x) - \int \psi(\mathbf{T}(x)) \alpha(\mathrm{d}x) \right|
\end{aligned}$$

For the second term, using the fact that $\mathbf{T}$ and ψ are λ- and 1-Lipschitz respectively, we have that $\psi \circ \mathbf{T}$ is λ-Lipschitz and therefore

$$\left| \int \psi(\mathbf{T}(x)) \alpha_N(\mathrm{d}x) - \int \psi(\mathbf{T}(x)) \alpha(\mathrm{d}x) \right| \leq \lambda \mathcal{W}_1(\alpha_N, \alpha) \leq \lambda \mathcal{W}_2(\alpha_N, \alpha),$$

where we used Assumption B.2 for that last inequality For the first term recall that using Cauchy-Schwarz and Jensen we get

$$\begin{aligned}
&\left|\int \psi(\mathbf{T}_{N,\epsilon}(x))\alpha_N(\mathrm{d}x) - \int \psi(\mathbf{T}(x))\alpha_N(\mathrm{d}x)\right| \\
&\leq \int \left|\mathbf{T}_{N,\epsilon}(x) - \mathbf{T}(x)\right| \alpha_N(\mathrm{d}x) \\
&\leq \int \left|\int y\mathcal{P}_{N,\epsilon}(x,\mathrm{d}y) - \mathbf{T}(x)\right| \alpha_N(\mathrm{d}x) \\
&\leq \iint \left|y - \mathbf{T}(x)\right| \alpha_N(\mathrm{d}x)\mathcal{P}_{N,\epsilon}(x,\mathrm{d}y) \\
&\leq \left[\iint \left|y - \mathbf{T}(x)\right|^2 \alpha_N(\mathrm{d}x)\mathcal{P}_{N,\epsilon}(x,\mathrm{d}y)\right]^{1/2}.
\end{aligned}$$

Here we can directly apply Lemma C.2 to obtain

$$\begin{aligned}
&\left[\iint \left|y - \mathbf{T}(x)\right|^2 \alpha_N(\mathrm{d}x)\mathcal{P}_{N,\epsilon}(x,\mathrm{d}y)\right]^{1/2} \\
&\leq 2\lambda^{1/2}\mathcal{E}^{1/2}\left[\mathfrak{d}^{1/2} + \mathcal{E}\right]^{1/2} + \max\{\lambda, 1\}\left[\mathcal{W}_2(\alpha_N, \alpha) + \mathcal{W}_2(\beta_N, \beta)\right],
\end{aligned}$$

where

$$\mathcal{E} \coloneqq \mathcal{E}(n, \epsilon, \alpha, \beta) \coloneqq \mathcal{W}_2(\alpha_N, \alpha) + \mathcal{W}_2(\beta_N, \beta) + \sqrt{2\epsilon\log(N)}.$$

From (35) we have that

$$\mathcal{W}_2(\alpha_N, \alpha) + \mathcal{W}_2(\beta_N, \beta) \leq \mathcal{W}_2(\alpha_N, \alpha) + \sqrt{\mathfrak{d}K_1(\Delta, \mathfrak{d})\mathcal{W}_2(\alpha_N, \alpha)}.$$

Next we want to bound $\mathcal{W}_2(\alpha_N, \alpha)$. Notice first that

$$\mathcal{W}_2(\alpha_N, \alpha) \leq \mathcal{W}_2(\alpha_N, \alpha'_N) + \mathcal{W}_2(\alpha'_N, \alpha) \leq \mathcal{W}_2(\alpha_N, \alpha'_N) + \kappa\mathcal{W}_2\left(\tilde{\beta}_N^{(t-1)}, \beta^{(t-1)}\right),$$

by Assumption B.2.

To control the other term we use (Fournier & Guillin, 2015) to obtain a high probability bound on $\mathcal{W}_2(\alpha_N, \alpha'_N)$. In particular, using Theorem 2 from (Fournier & Guillin, 2015), with $\alpha = \infty$ since we are in a compact domain, that for some positive constants C, c we have

$$\mathbb{P}\left[\mathcal{W}_2^2(\alpha_N, \alpha'_N) \geq x\right] \leq C\exp\left[-cNf_d^2(x)\right], \tag{36}$$

where

$$f_d(x) \coloneqq \begin{cases} x, & d < 4 \\ \frac{x}{\log(2+1/x)}, & d = 4 \\ x^{d/2}, & d > 4. \end{cases} \tag{37}$$

In particular, for any $\delta > 0$, with probability at least $1 - \delta$ over the sampling step in F_N we have that

$$\mathcal{W}_2(\alpha_N, \alpha'_N) \leq \sqrt{f_d^{-1}\left(\frac{\log(C/\delta)}{cN}\right)}. \tag{38}$$

Assuming that $d \geq 4$ the rate then is of order $N^{-1/d}$ as expected.

Therefore with probability at least $1 - \delta$ over the sampling step we have that

$$\mathcal{W}_2(\alpha_N, \alpha) + \mathcal{W}_2(\beta_N, \beta) \leq \mathcal{F}_{N,\epsilon,\delta,d}\left(\mathcal{W}_2\left(\tilde{\beta}_N^{(t-1)}, \beta^{(t-1)}\right)\right),$$

where

$$\mathcal{F}_{N,\epsilon,\delta,d}(x) = \mathcal{F}\left(\kappa x + \sqrt{f_d^{-1}\left(\frac{\log(C/\delta)}{cN}\right)}\right), \qquad \mathcal{F}(x) \coloneqq x + \sqrt{\mathfrak{d}K_1(\Delta, \mathfrak{d})x} \tag{39}$$

Thus overall we have with probability at least $1-\delta$ over the sample

$$\mathcal{W}_2(\tilde{\beta}_{N,\epsilon},\tilde{\beta}) \leq \sqrt{\mathfrak{d}\mathcal{W}_1(\tilde{\beta}_{N,\epsilon},\tilde{\beta})} \leq \mathfrak{G}_{\epsilon,\delta,N,d}\left(\mathcal{W}_2\left(\tilde{\beta}_N^{(t-1)},\beta^{(t-1)}\right)\right),$$

where

$$\begin{aligned}\frac{1}{\mathfrak{d}}\mathfrak{G}^2_{\epsilon,\delta,N,d}(x) \coloneqq 2\lambda^{1/2}\left[\mathcal{F}_{N,\epsilon,\delta,d}(x)+\sqrt{2\epsilon\log N}\right]^{1/2}\left[\mathfrak{d}^{1/2}+\mathcal{F}_{N,\epsilon,\delta,d}(x)+\sqrt{2\epsilon\log N}\right]^{1/2}\\ +\lambda\kappa\mathcal{F}_{N,\epsilon,\delta,d}(x)+\max\{\lambda,1\}\mathcal{F}_{N,\epsilon,\delta,d}(x).\end{aligned}$$

In particular notice that if we set $\epsilon_N = o(1/\log N)$ and $x_N = o(1)$ we have

$$\mathfrak{G}_{\epsilon_N,\delta,N,d}(x_N) \to 0.$$

Therefore, notice that if $\epsilon_N = o(1/\log N)$ and $\mathcal{W}_2(\mu_N,\mu)\to 0$, then for any $x>0$ we have that

$$\mathbb{P}\left[\mathcal{W}_2\big(\tilde{\beta}_{N,\epsilon},\tilde{\beta}\big)\geq x\right] \leq \mathbb{P}[\mathcal{W}_2(\alpha'_N,\alpha_N)\geq x'],$$

for some x' that does not depend on N, where the probability is over the sampling step. The convergence in probability follows. □

Proposition D.2. *Let $\mu_N = \frac{1}{N}\sum_{i=1}^N \delta_{X_1^i}$ where $X_1^i \overset{\text{i.i.d.}}{\sim} \mu \coloneqq q(\cdot|y_1)$ for $i\in[N]$ and suppose that for $t\geq 1$, $\alpha_N^{(t)}$ is defined through (31). Under Assumptions B.1, B.2, B.3 and B.4, for any $\delta>0$, with probability at least $1-2\delta$ over the sampling steps, for any bounded 1-Lipschitz ψ, for any $t\in[1:T]$, the approximations of the filtering distributions and log-likelihood computed by DPF satisfy*

$$|\tilde{\beta}_N^{(t)}(\psi)-\beta^{(t)}(\psi)| \leq \mathfrak{G}^{(t)}_{\epsilon,\delta/T,N,d}\left(\sqrt{f_d^{-1}\left(\frac{\log(CT/\delta)}{cN}\right)}\right) \tag{40}$$

$$\left|\log\frac{\hat{p}_N(y_{1:T})}{p(y_{1:T})}\right| \leq \frac{\kappa}{\Delta}\max_{t\in[1:T]}\mathrm{Lip}\left[g(y_t\mid\cdot)\right]\sum_{t=1}^T \mathfrak{G}^{(t)}_{\epsilon,\delta/T,N,d}\left(\sqrt{f_d^{-1}\left(\frac{\log(CT/\delta)}{cN}\right)}\right) \tag{41}$$

where C is a finite constant independent of T, $\mathfrak{G}_{\epsilon,\delta/T,N,d}$, f_d are defined in (32), and $\mathrm{Lip}[f]$ is the Lipschitz constant of the function f. $\mathfrak{G}^{(t)}_{\epsilon,\delta/T,N,d}$ denotes the t-repeated composition of function $\mathfrak{G}_{\epsilon,\delta/T,N,d}$. In particular, if we set $\epsilon_N = o(1/\log N)$

$$\left|\log\frac{\hat{p}_N(y_{1:T})}{p(y_{1:T})}\right| \to 0,$$

in probability.

Proof of Proposition D.2. Following the proof of Proposition D.1, we define $\alpha_N^{(t)'} = \tilde{\beta}_N^{(t-1)}f$ and for $t\in[1:T]$, the events

$$A_t \coloneqq \mathcal{W}_2\left(\alpha_N^{(t)},\alpha_N^{(t)'}\right) \leq \sqrt{f_d^{-1}\left(\frac{\log(CT/\delta)}{cN}\right)}.$$

We know from Theorem 2 in (Fournier & Guillin, 2015) that $\mathbb{P}(A_t)\geq 1-\delta/T$, where the probability is over the sampling step. In particular we have that

$$\mathbb{P}\left[\bigcap_{t=1}^T A_t\right] = 1-\mathbb{P}\left[\bigcup_{t=1}^T A_t^{\mathsf{c}}\right] \geq 1-\sum_{t=1}^N \mathbb{P}\left[A_t^{\mathsf{c}}\right] \geq 1-T\frac{\delta}{T} = 1-\delta.$$

Notice that on the event $\cap_{t=1}^T A_t$, iterating the bound (33) we have

$$\mathcal{W}_2\left(\tilde{\beta}_N^{(t)},\beta^{(t)}\right) \leq \mathfrak{G}^{(t)}_{\epsilon,\delta/T,N,d}\left(\mathcal{W}_2\left(\mu_N,\mu\right)\right),$$

with probability at least $1-\delta$. Again by Theorem 2 in (Fournier & Guillin, 2015) we have that with probability at least $1-\delta$

$$\mathcal{W}_2(\mu_N, \mu) \leq \sqrt{f_d^{-1}\left(\frac{\log(CT/\delta)}{cN}\right)}.$$

Therefore with probability at least $1-2\delta$ we have

$$\mathfrak{G}^{(t)}_{\epsilon,\delta/T,N,d}\left(\sqrt{f_d^{-1}\left(\frac{\log(CT/\delta)}{cN}\right)}\right).$$

It remains to prove (41). Note that $|\log(x) - \log(y)| \leq \frac{|x-y|}{\min\{x,y\}}$ for any $x, y > 0$ so

$$\begin{aligned}
\left|\log \hat{p}(y_{1:T}) - \log p(y_{1:T})\right| &\leq \sum_{t=1}^{T} \left|\log \hat{p}(y_t|y_{1:t-1}) - \log p(y_t|y_{1:t-1})\right| \\
&\leq \sum_{t=1}^{T} \left|\frac{\hat{p}(y_t|y_{1:t-1}) - p(y_t|y_{1:t-1})}{\min(\hat{p}(y_t|y_{1:t-1}), p(y_t|y_{1:t-1}))}\right| \\
&\leq \Delta^{-1} \sum_{t=1}^{T} \left|\hat{p}(y_t|y_{1:t-1}) - p(y_t|y_{1:t-1})\right| \qquad (42)
\end{aligned}$$

where Δ is defined in Assumption B.3.

The term in line (42) may be written as follows

$$\begin{aligned}
&\hat{p}(y_t|y_{1:t-1}) - p(y_t|y_{1:t-1}) \\
=&\iint g(y_t|x_t) f(\mathrm{d}x_t|\tilde{x}_{t-1}) \tilde{\beta}_N^{(t-1)}(\mathrm{d}\tilde{x}_{t-1}) - \iint g(y_t|x_t) f(\mathrm{d}x_t|\tilde{x}_{t-1}) \tilde{\beta}^{(t-1)}(\mathrm{d}\tilde{x}_{t-1}) \\
=&\tilde{\beta}_N^{(t-1)}(h) - \beta^{(t-1)}(h)
\end{aligned}$$

for $\Delta^2 \leq h(x) := \int g(y_t|x') f(x'|x) \mathrm{d}x' \leq \Delta^{-2}$. At this point notice also that

$$\begin{aligned}
h(x) - h(x') &= \int f(\mathrm{d}w|x) g(y_t \mid w) - \int f(\mathrm{d}w|x') g(y_t \mid w) \\
&= \int \delta_x(\mathrm{d}z) \int f(\mathrm{d}w|z) g(y_t \mid w) - \int \delta_{x'}(\mathrm{d}z) \int f(\mathrm{d}w|z) g(y_t \mid w) \\
&= [\delta_x f][g(y_t \mid \cdot)] - [\delta_{x'} f][g(y_t \mid \cdot)] \\
&\leq \mathrm{Lip}\left[g(y_t \mid \cdot)\right] \mathcal{W}_1\left(\delta_x f, \delta_{x'} f\right) \leq \kappa \mathrm{Lip}\left[g(y_t \mid \cdot)\right] \mathcal{W}_1\left(\delta_x, \delta_{x'}\right) = \kappa \mathrm{Lip}\left[g(y_t \mid \cdot)\right] |x - x'|,
\end{aligned}$$

by Assumption B.2. It follows therefore that h is Lipschitz and therefore that

$$\hat{p}(y_t|y_{1:t-1}) - p(y_t|y_{1:t-1}) = \tilde{\beta}_N^{(t-1)}(h) - \beta^{(t-1)}(h) \leq \kappa \mathrm{Lip}\left[g(y_t \mid \cdot)\right] \mathcal{W}_1\left(\beta_N^{(t-1)}, \beta^{(t-1)}\right).$$

Combining (40) and (42), and using the fact that $\mathcal{W}_1 \leq \mathcal{W}_2$, we thus get

$$\begin{aligned}
\left|\log \hat{p}(y_{1:T}) - \log p(y_{1:T})\right| &\leq \Delta^{-1} \kappa \sum_{t=1}^{T} \mathrm{Lip}\left[g(y_t \mid \cdot)\right] \mathcal{W}_1\left(\tilde{\beta}_N^{(t-1)}, \beta^{(t)}\right) \\
&\leq \Delta^{-1} \kappa \max_{t \in [1:T]} \mathrm{Lip}\left[g(y_t \mid \cdot)\right] \sum_{t=1}^{T} \mathfrak{G}^{(t)}_{\epsilon,\delta/T,N,d}\left(\sqrt{f_d^{-1}\left(\frac{\log(CT/\delta)}{cN}\right)}\right),
\end{aligned}$$

where the last inequality holds with probability at least $1-\delta$ over the sampling steps.

The convergence in probability follows from the corresponding statement of Proposition D.1. □

E. Additional Experiments and Details

E.1. Linear Gaussian model

We first consider the following 2-dimensional linear Gaussian SSM for which exact inference can be carried out using Kalman techniques:

$$X_t|\{X_{t-1}=x\} \sim \mathcal{N}\left(\text{diag}(\theta_1\ \theta_2)x, 0.5\mathbf{I}_2\right), \quad Y_t|\{X_t=x\} \sim \mathcal{N}(x, 0.1\cdot\mathbf{I}_2). \tag{43}$$

We simulate $T=150$ observations using $\theta=(\theta_1,\theta_2)=(0.5,0.5)$. As a result, we expect in these scenarios that the filtering distribution $p_\theta(x_t|y_{1:t})$ is not too distinct from the smoothing distribution $p_\theta(x_t|y_{1:T})$ as the latent process is mixing quickly. From Proposition 4.1, this is thus a favourable scenario for methods ignoring resampling terms in the gradient as the bias should not be very large. Figure 1, displayed earlier, shows $\ell(\theta)$ obtained by Kalman and $\hat{\ell}(\theta;\mathbf{u})$ computed regular PF and DPF for the same number $N=25$ of particles using $q_\phi(x_t|x_{t-1},y_t)=f_\theta(x_t|x_{t-1})$. The corresponding gradient vector fields are given in Figure 1, where the gradient is computed using the biased gradient from (Maddison et al., 2017; Naesseth et al., 2018; Le et al., 2018) for regular PF.

We now compare the performance of the estimators $\hat{\theta}_{\text{SMLE}}$ (for DPF) and $\hat{\theta}_{\text{ELBO}}$ (for both regular PF and DPF) learned using gradient with learning rate 10^{-4} on 100 steps, using $N=25$ for DPF and $N=500$ for regular PF, to $\hat{\theta}_{\text{MLE}}$ computed using Kalman derivatives. We simulate $M=50$ realizations of $T=150$ observations using $\theta=(\theta_1,\theta_2)=(0.5,0.5)$. The ELBO stochastic gradient estimates are computed using biased gradient estimates of $\ell_{\text{ELBO}}(\theta)$ ignoring the contributions of resampling steps as in (Maddison et al., 2017; Naesseth et al., 2018; Le et al., 2018) (we recall that unbiased estimates suffer from very high variance) and unbiased gradients of $\ell^{\text{ELBO}}(\theta)$ using DPF. We average B parallel PFs to reduce the variance of these gradients of the ELBO and also B PFs (with fixed random seeds) to compute the gradient of $\hat{\ell}_{\text{SMLE}(\theta;\mathbf{u}_{1:B})} := \frac{1}{B}\sum_{b=1}^{B}\hat{\ell}(\theta;\mathbf{u}_b)$. The results are given in Table 4. For this example, $\hat{\theta}^{\text{DPF}}_{\text{ELBO}}$ maximizing $\ell^{\text{ELBO}}_{\text{DPF}}(\theta)$ outperforms $\hat{\theta}^{\text{PF}}_{\text{ELBO}}$ and $\hat{\theta}_{\text{SMLE}}$. However, as B increases, $\hat{\theta}_{\text{SMLE}}$ gets closer to $\hat{\theta}^{\text{DPF}}_{\text{ELBO}}$ which is to be expected as $\hat{\ell}_{\text{SMLE}}(\theta;\mathbf{u}_{1:B}) \longrightarrow \ell^{\text{ELBO}}(\theta)$. In Table 4, the Root Mean Square Error (RMSE) is defined as $\sqrt{\frac{1}{M}\sum_{i=1}^{2}\sum_{k=1}^{M}(\hat{\theta}_i^k-\hat{\theta}^k_{\text{MLE},i})^2}$.

Table 4. $10^3\times$ RMSE[4] over 50 datasets - lower is better

B	$\hat{\theta}^{\text{PF}}_{\text{ELBO}}$	$\hat{\theta}^{\text{DPF}}_{\text{ELBO}}$	$\hat{\theta}_{\text{SMLE}}$
1	1.94	1.30	7.94
4	2.40	1.35	3.28
10	2.80	1.37	2.18

E.2. Variational Recurrent Neural Network

$N=32$ particles were used for training, with a regularization parameter of $\epsilon=0.5$. The ELBO (scaled by sequence length) was used as the training objective to maximise for each resampling/ DET procedure. The ELBO evaluated on test data using $N=500$ particles and multinomial resampling. Resampling / DET operations were carried out when effective sample (ESS) size fell below $N/2$. Learning rate 0.001 was used with the Adam optimizer.

Recall the state-space model is given by

$$\begin{aligned}(R_t,O_t) &= \text{RNN}_\theta(R_{t-1},Y_{1:t-1},E_\theta(Z_{t-1})),\\ Z_t &\sim \mathcal{N}(\mu_\theta(O_t),\sigma_\theta(O_t)),\\ \hat{p}_t &= h_\theta(E_\theta(Z_t),O_t),\\ Y_t|X_t &\sim \text{Ber}(\hat{p}_t).\end{aligned}$$

Network architectures and data preprocessing steps were based loosely on (Maddison et al., 2017). Given the low volume of data and sparsity of the observations, relatively small neural networks were considered to prevent overfitting, larger neural

[4]The Root Mean Square Error (RMSE) is defined as $\sqrt{\frac{1}{M}\sum_{i=1}^{2}\sum_{k=1}^{M}(\hat{\theta}_i^k-\hat{\theta}^k_{\text{MLE},i})^2}$.

networks are considered in the more complex robotics experiments. R_t is of dimension $d_r = 16$, Z_t is of dimension $d_z = 8$. E_θ is a single layer fully connected network with hidden layer of width 16, output of dimension 16 and RELU activation.

μ_θ and σ_θ are both fully connected neural networks with two hidden layers, each of 16 units and RELU activation, the activation function is not applied to the final output of μ_θ but the softplus is applied to the output of σ_θ, which is the diagonal entries of the covariance matrix of the normal distribution that is used to sample Z_t.

h_θ is a single layer fully connected network with two hidden layers, each of width 16 and RELU activation. The final output is not put through the RELU and is instead used as the logits for the Bernoulli distribution of observations.

E.3. Robot Localization

Similar to the VRNN example, $N = 32$ particles were used for training, with a regularization parameter of $\epsilon = 0.5$ and resampling / DET operations were carried out when ESS size fell below $N/2$. Learning rate 0.001 was used with the Adam optimizer.

Network architectures and data preprocessing were based loosely on (Jonschkowski et al., 2018). There are 3 neural networks being considered:

- Encoder E_θ maps RBG 24×24 pixel images, hence dimension $3 \times 24 \times 24$, to encoding of size $d_E = 128$. This network consists of a convolutional network (CNN) of kernel size 3 and a single layer fully connected network of hidden width 128 and RELU activation.
- Decoder D_θ maps encoding back to original image. This consists of a fully connected neural network with three hidden layers of width 128 and RELU activation function. This is followed by a transposed convolution network with matching specification to the CNN in the encoder, to return an output with the same dimension as observation images, $3 \times 24 \times 24$.
- Network G_θ maps the state $S_t = (X_t^{(1)}, X_t^{(2)}, \gamma_t)$ to encoding of dimension 128. First angle γ_t was converted to $\sin(\gamma_t), \cos(\gamma_t)$. Then the augmented state $(X_t^{(1)}, X_t^{(2)}, \sin(\gamma_t), \cos(\gamma_t))$ was passed to a 3 layer fully connected network with hidden layers of dimensions $16, 32, 64$ and RELU activation function, with final output of dimension 128.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Differentiable Particle Filtering via Regularized Optimal Transport
Publication Status	□ Accepted for Publication
Publication Details	Adrien Corenflos*, James Thornton*, George Deligiannidis, Arnaud Doucet, published at ICML 2021 * Equal contribution.

Student Confirmation

Student Name:	James Thornton		
Contribution to the Paper	- Adrien and I jointly wrote the code for the primary method. - I coded and executed neural network experiments, in particular the larger scale experiments involving neural networks, such as the robotic localization and variational RNN experiments. Adrien carried out the experiments involving learning the proposal model. - I devised initial asymptotic convergence results, later improved by George. - I raised $1k compute credits to fund the project experiments on cloud compute - Initial idea came from Arnaud Doucet. - All jointly contributed to writing the paper		
Signature 		Date	23/03/2023

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Arnaud Doucet, Professor of Statistics		
Supervisor comments I agree with the statement of contributions.		
Signature 	Date	23/03/2023

This completed form should be included in the thesis, at the end of the relevant chapter.

Chapter 4

Rethinking Initialization of the Sinkhorn Algorithm

Rethinking Initialization of the Sinkhorn Algorithm

James Thornton
University of Oxford

Marco Cuturi
Apple

Abstract

While the optimal transport (OT) problem was originally formulated as a linear program, the addition of entropic regularization has proven beneficial both computationally and statistically, for many applications. The Sinkhorn fixed-point algorithm is the most popular approach to solve this regularized problem, and, as a result, multiple attempts have been made to reduce its runtime using, e.g., annealing in the regularization parameter, momentum or acceleration. The premise of this work is that *initialization* of the Sinkhorn algorithm has received comparatively little attention, possibly due to two preconceptions: since the regularized OT problem is convex, it may not be worth crafting a good initialization, since *any* is guaranteed to work; secondly, because the outputs of the Sinkhorn algorithm are often unrolled in end-to-end pipelines, a data-dependent initialization would bias Jacobian computations. We challenge this conventional wisdom, and show that data-dependent initializers result in dramatic speed-ups, with no effect on differentiability as long as implicit differentiation is used. Our initializations rely on closed-forms for exact or approximate OT solutions that are known in the 1D, Gaussian or GMM settings. They can be used with minimal tuning, and result in consistent speed-ups for a wide variety of OT problems.

1 Introduction

The optimal assignment problem and its generalization, the optimal transport (OT) problem, play an increasingly important role in modern machine learning. These problems define the Wasserstein geometry (Santambrogio, 2015; Peyré et al., 2019), which is routinely used as a loss function in imaging (Schmitz et al., 2018; Janati et al., 2020), but also used to reconstruct correspondences between datasets, as for instance in domain adaptation (Courty et al., 2014, 2017) or single-cell genomics (Schiebinger et al., 2019). Several recent applications use OT to obtain an intermediate representation, as in self-supervised learning (Caron et al., 2020), balanced attention (Sander et al., 2022), parameterized matching (Sarlin et al., 2020), differentiable sorting and ranking (Adams and Zemel, 2011; Cuturi et al., 2019, 2020; Xie et al., 2020a), differentiable resampling (Corenflos et al., 2021) and clustering (Genevay et al., 2019).

Sinkhorn as a subroutine for OT. A striking feature of all of the approaches outlined above is that they do not rely on the linear programming formulation of OT (Ahuja et al., 1988, §9-11), but use instead an entropy regularized formulation (Cuturi, 2013). This formulation is typically solved with the Sinkhorn algorithm (1967), which has gained popularity for its versatility, efficiency and differentiability.

Ever Faster Sinkhorn. Given two discrete measures, the Sinkhorn algorithm runs a fixed-point iteration that outputs two optimal dual vectors, along with their objective–a proxy for their Wasserstein distance. Because Sinkhorn is often used as an inner routine within more complex architectures, its contribution to the total runtime may result in a substantial share of the entire computational burden. As a result, accelerating the Sinkhorn algorithm is crucial, and has been explored along two lines of works: through faster kernel matrix-vector multiplications, using geometric properties (Solomon et al., 2015; Altschuler et al., 2019; Scetbon and Cuturi, 2020), or by reducing the total number of iterations needed to converge, using e.g. an annealing regularization parameter (Kosowsky and Yuille, 1994; Schmitzer, 2019; Xie et al., 2020b), momentum (Thibault et al., 2021; Lehmann et al., 2021), or Anderson acceleration (1965), as considered in (Chizat et al., 2020).

Initialization as a Blind Spot. All methods above are, however, implemented by default by setting initial dual vectors naively at $\mathbf{0}$.

To our knowledge, initialization schemes have only been explored in a few restricted setups, such as semi-discrete settings in 2/3D (Meyron, 2019), or for discrete Wasserstein barycenter problems (Cuturi and Peyré, 2015). We

Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS) 2023, Valencia, Spain. PMLR: Volume 206.

argue that careful initialization of dual potentials presents an overlooked opportunity for efficiency.

Contributions. We propose multiple methods to initialize dual vectors. Contrary to concurrent and complementary work by Amos et al. (2022), our initializers are not trained, and not limited to fixed support setups. They require minimal hyperparameter tuning and result in small to negligible overheads. To do so, we leverage closed-form formulae and approximate solutions for simpler OT problems, resulting in the following procedures:

- We introduce a method to recover dual vectors when the primal problem solution is known in closed-form, and apply this to the non-regularized 1D problem. We show that initializing Sinkhorn with these vectors results in orders of magnitude speedups that can be readily applied to differentiable sorting and ranking.
- When the ground cost is the squared L2 distance in $\mathbb{R}^d$, $d > 1$, we leverage closed-form dual potential functions from the Gaussian approximation of source/target measures, and evaluate them on source points to initialize the Sinkhorn algorithm. We extend this by introducing an approximation of OT potentials for Gaussian *mixtures*.
- Finally we reformulate the multiscale approach of (Feydy, 2020, Alg. 3.6) as a subsample initializer.

We provide extensive empirical evaluation, and compare our approaches to other acceleration methods. We show that our initializations are robust and effective, outperforming existing alternatives, yet can also work in combination with them to achieve even better results.

2 Background material on OT

2.1 Entropic Regularization and Sinkhorn

Given two discrete probability measures $\mu = \sum_{i=1}^n a_i \delta_{x_i}$ and $\nu = \sum_{j=1}^m b_j \delta_{y_j}$ in $\mathcal{P}(\mathbb{R}^d)$, where $\mathbf{a} = (a_1, \dots, a_n)$, $\mathbf{b} = (b_1, \dots, b_m)$ are probability weights and $(\mathbf{x}_1, \dots, \mathbf{x}_n) \in \mathbb{R}^{d\times n}$, $(\mathbf{y}_1, \dots, \mathbf{y}_m) \in \mathbb{R}^{d\times m}$, the entropy regularized OT problem between μ and ν parameterized by $\varepsilon \geq 0$ and a cost function c has two equivalent formulations,

$$\min_{\mathbf{P}\in\mathbb{R}_+^{n\times m}, \mathbf{P}\mathbf{1}_m = \mathbf{a}, \mathbf{P}^T\mathbf{1}_n = \mathbf{b}} \langle \mathbf{P}, \mathbf{C}\rangle - \varepsilon\langle \mathbf{P}, \log(\mathbf{P}) - 1\rangle, \quad (1)$$

$$\max_{\mathbf{f}\in\mathbb{R}^n, \mathbf{g}\in\mathbb{R}^m} \mathcal{E}_{\mu,\nu,c,\varepsilon}(\mathbf{f},\mathbf{g}) := \langle \mathbf{f}, \mathbf{a}\rangle + \langle \mathbf{g}, \mathbf{b}\rangle - \varepsilon\langle e^{\frac{\mathbf{f}}{\varepsilon}}, \mathbf{K} e^{\frac{\mathbf{g}}{\varepsilon}}\rangle. \quad (2)$$

where $\mathbf{C} := [c(\mathbf{x}_i, \mathbf{y}_j)]_{i,j}$, with corresponding kernel $\mathbf{K} := e^{-\mathbf{C}/\varepsilon}$. While $(\mathbf{f}, \mathbf{g})$ are unconstrained for $\varepsilon > 0$, the regularization term converges as $\varepsilon \to 0$ to an indicator function that requires $\mathbf{f}_i + \mathbf{g}_j \leq c(\mathbf{x}_i, \mathbf{y}_j)$.

The Sinkhorn Algorithm. Algorithm 1 describes a sequence of updates to optimize $\mathbf{f}, \mathbf{g}$ in (2). When $\omega = 1$, these updates correspond to cancelling alternatively the gradients $\nabla_1 \mathcal{E}_{\mu,\nu,c,\varepsilon}(\mathbf{f},\mathbf{g})$ (line 4) and $\nabla_2 \mathcal{E}_{\mu,\nu,c,\varepsilon}(\mathbf{f},\mathbf{g})$ (line 5) of the objective in (2). These updates use the row-wise soft-min operator $\min_\varepsilon$, defined as:

Algorithm 1: Sinkhorn's Algorithm

1: **Input:** $\mathbf{a}, \mathbf{b}, \mathbf{C}, \varepsilon > 0, \omega > 0, \mathbf{f}^{(0)}, \mathbf{g}^{(0)}$.
2: **Initialize:** $\mathbf{f} \leftarrow \mathbf{f}^{(0)}, \mathbf{g} \leftarrow \mathbf{g}^{(0)}$
3: **while** not converged **do**
4: $\quad \mathbf{f} \leftarrow \omega(\varepsilon \log \mathbf{a} - \min_\varepsilon(\mathbf{C} - \mathbf{f} \oplus \mathbf{g})) + \mathbf{f}$
5: $\quad \mathbf{g} \leftarrow \omega(\varepsilon \log \mathbf{b} - \min_\varepsilon(\mathbf{C}^T - \mathbf{g} \oplus \mathbf{f})) + \mathbf{g}$
6: **end while**
7: **Return** $\mathbf{f}, \mathbf{g}$

$$\text{Given } \mathbf{S} = [\mathbf{S}_{i,j}],\ \min_\varepsilon(\mathbf{S}) := \left[-\varepsilon \log\left(\mathbf{1}^T e^{-\mathbf{S}_{i,:}/\varepsilon}\right)\right]_i,$$

and the tensor addition notation $\mathbf{f} \oplus \mathbf{g} = [\mathbf{f}_i + \mathbf{g}_j]_{i,j}$. The runtime of the Sinkhorn algorithm hinges on several factors, notably the choice of ε. Several works report that hundreds of iterations are typically required when using fairly small regularization ε (e.g. 500 in Salimans et al. 2018, App.B). These scalability issues are compounded in advanced applications whereby multiple Sinkhorn layers are embedded in a single computation or batched across examples (Cuturi et al., 2019; Xie et al., 2020a; Cuturi et al., 2020). To mitigate runtime issues, popular acceleration techniques such as fixed (Thibault et al., 2021) or adaptive (Lehmann et al., 2021) momentum approaches, as well as Anderson acceleration (Chizat et al., 2020) have been considered. While acceleration methods are known to work well when initialized not too far away from optima (d'Aspremont et al., 2021), all common implementations (Flamary et al., 2021; Cuturi et al., 2022) initialize these vectors to $(\mathbf{0}_n, \mathbf{0}_m)$.

2.2 Dual Variables in the Sinkhorn Algorithm

On starting closer to the solution. While the Sinkhorn algorithm will converge with any initialization, the speed of convergence is bounded by (Peyré et al., 2019, Rem. 4.14):

$$\|\mathbf{f}^{(\ell)} - \mathbf{f}^\star\|_{\text{var}} \leq \|\mathbf{f}^{(0)} - \mathbf{f}^\star\|_{\text{var}} \lambda(\mathbf{K})^{2\ell}, \quad (3)$$

where $\mathbf{f}^{(\ell)}$ denotes the potential vector $\mathbf{f}$ obtained after running Algorithm 1 for ℓ iterations, $\mathbf{f}^\star$ the optimal potential, and deviation is measured using the variation norm. $\lambda(\mathbf{K})$ reflects conditioning in $\mathbf{K}$ (Peyré et al., 2019, Theorem 4.1), determined by the range and magnitude of costs evaluated on $(\mathbf{x}_i, \mathbf{y}_j)$ pairs relative to ε. Since $0 < \lambda(\mathbf{K}) < 1$, the Sinkhorn algorithm converges more slowly as $\lambda(\mathbf{K})$ approaches 1. The motivation to obtain a better initialization relies on targeting the initial gap in $\|\mathbf{f}^{(0)} - \mathbf{f}^\star\|_{\text{var}}$.

Two or One Dual Initializations? While Algorithm 1 lists two initial vectors $(\mathbf{f}^{(0)}, \mathbf{g}^{(0)})$, a closer inspection of the updates shows that only a single dual variable is needed: when starting with an iteration updating $\mathbf{g}$, only $\mathbf{f}^{(0)}$ is required (the reference to $\mathbf{g}$ is only there for numerical

stability). Conversely, only $\mathbf{g}^{(0)}$ is required when updating $\mathbf{f}$. Since only one is needed, we supply by default the smallest vector when $n \neq m$, and set the other to $\mathbf{0}$.

Differentiability and Dual initialization. Any output of the Sinkhorn fixed-point algorithm can be differentiated using unrolling (Adams and Zemel, 2011; Hashimoto et al., 2016; Genevay et al., 2018, 2019; Cuturi et al., 2019; Caron et al., 2020). This approach has, however, two drawbacks: its memory footprint grows as $L(n+m)$, where L is the number of iterations needed to converge, and, more fundamentally, it prevents us from using more efficient steps, such as adaptive momentum and acceleration, because they typically involve non-differentiable operations. These issues can be avoided by relying instead on implicit differentiation (Luise et al., 2018; Cuturi et al., 2020; Xie et al., 2020b; Cuturi et al., 2022), which only requires access to solutions $\mathbf{f}^\star, \mathbf{g}^\star$ to work. We recall how this can be implemented for completeness. Introducing the following notations:

$$F : \mu, \nu, c, \varepsilon \mapsto \mathbf{f}^\star, \mathbf{g}^\star, \text{optimal solutions to (2)},$$
$$H : \mu, \nu, c, \varepsilon, \mathbf{f}, \mathbf{g} \mapsto \begin{bmatrix} \nabla_1 \mathcal{E}_{\mu,\nu,c,\varepsilon}(\mathbf{f}, \mathbf{g}) \\ \nabla_2 \mathcal{E}_{\mu,\nu,c,\varepsilon}(\mathbf{f}, \mathbf{g}) \end{bmatrix},$$

one has that $H(\mu, \nu, c, \varepsilon, F(\mu, \nu, c, \varepsilon)) = \mathbf{0}_{n+m}$, which is the root equation that can be used to instantiate the implicit function theorem, to recover the Jacobian of the outputs of F (i.e. $\mathbf{f}^\star, \mathbf{g}^\star$) w.r.t. *any* variable "■" within inputs. As a result, the transpose-Jacobian of F applied to any perturbation of the size of ■ (the only operation needed to implement reverse-mode differentiation) is recovered as (where ... is a shorthand notation for ■, $(\mathbf{f}^\star, \mathbf{g}^\star)$):

$$J_{F,\blacksquare}(\dots)^T z = -J_{H,\blacksquare}(\dots)^T \left(J_{H,(\mathbf{f},\mathbf{g})}(\dots)^T\right)^{-1} \mathbf{z}$$

All of these operations can be instantiated easily using `vjp` Jacobian operators (Bradbury et al., 2018) and linear systems that rely on linear functions (rather than matrices) as detailed in (Cuturi et al., 2022). These computations only require access to optimal values $\mathbf{f}^\star, \mathbf{g}^\star$, not the computational graph that was needed to reach them.

2.3 Closed-Form Expressions in Optimal Transport

A few closed-forms for *unregularized* ($\varepsilon = 0$) OT are known. Some of these closed forms rely on the Monge formulation of OT, recalled for completeness for two measures $\mu, \nu \in \mathcal{P}(\mathbb{R}^d)$ in (4), using the push-forward $\sharp$ notation, as well as the dual formulation of OT in (5), using the convention $f^c(\mathbf{y}) := \min_{\mathbf{x}} c(\mathbf{x}, \mathbf{y}) - f(\mathbf{x})$, the c-transform of f.

$$\min_{\substack{T:\mathbb{R}^d \to \mathbb{R}^d \\ T_\sharp \mu = \nu}} \int c(\mathbf{x}, T(\mathbf{x})) \mathrm{d}\mu(\mathbf{x}). \tag{4}$$

$$\max_{f:\mathbb{R}^d \to \mathbb{R}} \int f \mathrm{d}\mu + \int f^c \mathrm{d}\nu\,. \tag{5}$$

We review two relevant cases, where either an optimal coupling $\mathbf{P}^\star$ (for $\varepsilon = 0$) in the primal formulation of (1), or an optimal map $T^\star$ to (4) can be obtained in closed form. We show in §3 how these solutions can be leveraged to recover initialization vectors $\mathbf{f}^{(0)}$ and $\mathbf{g}^{(0)}$ for Alg. 1.

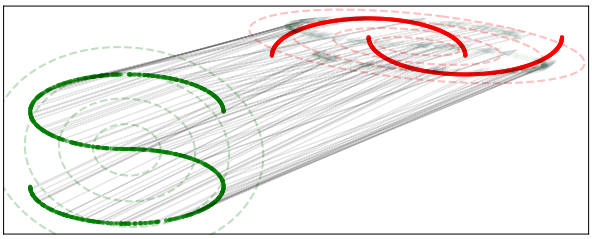


Figure 1: Transport map (black) from Gaussian approximations (dashed) of S-curve (green) and two-moons (red)

OT in 1D. For univariate data ($d = 1$), and when the cost function c is such that $-c$ is supermodular ($\partial c / \partial x \partial y < 0$), a solution $\mathbf{P}^\star$ to (1) can be recovered in closed form (Chiappori et al., 2017; Santambrogio, 2015, §3). Writing σ, ρ for sorting permutations of the supports of μ and ν, $x_{\sigma_1} \leq \cdots \leq x_{\sigma_n}$ and $y_{\rho_1} \leq \dots\ y_{\rho_m}$, a solution $\mathbf{P}^\star$ is given by the *north-west corner* solution $\mathbf{NW}(\mathbf{a}_\sigma, \mathbf{b}_\rho)$, where $\mathbf{a}_\sigma$ and $\mathbf{b}_\rho$ are the weight vectors $\mathbf{a}, \mathbf{b}$ permuted using σ and ρ respectively (Peyré et al., 2019, §3.4.2).

Gaussian. The Monge formulation of the OT problem (4) from a Gaussian measure $\mathfrak{N}_1 = \mathcal{N}(\mathbf{m}_1, \mathbf{\Sigma}_1), \mathbf{\Sigma}_1 > 0$, to another $\mathfrak{N}_2 = \mathcal{N}(\mathbf{m}_2, \mathbf{\Sigma}_2)$, is solved by (see also Fig. 1):

$$T^\star(\mathbf{x}) := \mathbf{A}(\mathbf{x} - \mathbf{m}_1) + \mathbf{m}_2, \mathbf{A} = \mathbf{\Sigma}_1^{-\frac{1}{2}} (\mathbf{\Sigma}_1^{\frac{1}{2}} \mathbf{\Sigma}_2 \mathbf{\Sigma}_1^{\frac{1}{2}})^{\frac{1}{2}} \mathbf{\Sigma}_1^{-\frac{1}{2}}.$$

The optimal dual *potential* $f^\star$ is a quadratic form given by

$$f^\star(\mathbf{x}) = \tfrac{1}{2} \mathbf{x}^T (\mathbf{I} - \mathbf{A}) \mathbf{x} + (\mathbf{m}_2 - \mathbf{A}\mathbf{m}_1)^T \mathbf{x}, \tag{6}$$

which recovers $T^\star = \mathrm{Id} - \nabla f^\star$. The OT cost between $\mathfrak{N}_1$ and $\mathfrak{N}_2$ is known as the Bures-Wasserstein distance:

$$W_2^2(\mathfrak{N}_1, \mathfrak{N}_2) = \|\mathbf{m}_1 - \mathbf{m}_2\|^2 + \mathcal{B}_2^2(\Sigma_1, \Sigma_2)\,,$$
$$\mathcal{B}_2^2(\Sigma_1, \Sigma_2) := \mathrm{tr}(\mathbf{\Sigma}_1 + \mathbf{\Sigma}_2 - 2(\mathbf{\Sigma}_1^{\frac{1}{2}} \mathbf{\Sigma}_2 \mathbf{\Sigma}_1^{\frac{1}{2}})^{\frac{1}{2}})\,. \tag{7}$$

3 Crafting Sinkhorn Initializations

We present important scenarios where careful initialization can dramatically speed up the Sinkhorn algorithm. We start with the 1D case (§3.1), where entropic transport has been used recently as a possible approach to obtain differentiable rank and sorting operators. We follow with the generic and by now standard multivariate OT problem in $\mathbb{R}^d$ with squared-L2 ground cost, using Gaussian approximations (§3.2) and an extension to mixtures (§3.3).

3.1 Initialization for 1D Regularized OT

Ranking as an OT problem. Using a cost c on $\mathbb{R} \times \mathbb{R}$ such that $\partial c / \partial x \partial y < 0$, sorting the entries of a vector $\mathbf{x} =$

$(x_1, \dots, x_n) \in \mathbb{R}^n$ can be recovered using a solution $\mathbf{P}^\star$ to (1), setting $\varepsilon = 0$, $\mathbf{a} = \mathbf{1}_n/n$, and ν to a uniform measure on n increasing numbers, e.g. $\mathbf{y} = (1, 2, \dots, n)$. The ranks of the entries of $\mathbf{x}$ are then $n\mathbf{P}^\star\mathbf{z}$, where $\mathbf{z} = (1, 2, ..., n)$, and its sorted entries as $n\mathbf{P}^{\star T}\mathbf{x}$ (Cuturi et al., 2019).

Differentiable Ranking. A differentiable and fractional soft sorting/ranking operator can be derived from entropy regularized couplings, using instead a solution $\mathbf{P}_\varepsilon$ to (1, $\varepsilon > 0$) to form $n\mathbf{P}_\varepsilon\mathbf{z}$ and $n\mathbf{P}_\varepsilon^T\mathbf{x}$ (Cuturi et al., 2019), with the possibility to use a different target size m or non-uniform weights $\mathbf{a}, \mathbf{b}$. A practical challenge of that approach is that the number of Sinkhorn iterations needed for the coupling to converge can be typically quite large, see Figure 3 and further results in Appendix B.1.

Dual 1D Initializers. Regularized 1D OT problems often require a small regularization ε to be meaningful, in order to recover rank approximations that are not too smoothed, which then requires many Sinkhorn iterations to converge. To address this, we introduce an initializer using potentials for the non-regularized problem ($\varepsilon = 0$). Our strategy to pick initialization vectors for Algorithm 1 is upon first glance deceptively simple: sort $\mathbf{x}$, recover a primal solution $\mathbf{P}^\star$ (the North-West corner solution) that is guaranteed to solve (1); turn it into a pair of optimal dual vectors $\mathbf{f}_0^\star, \mathbf{g}_0^\star$ for the same unregularized problem, and seed them to Alg. 1 to solve 2 with $\varepsilon > 0$. While obtaining $\mathbf{P}^\star$ only requires a sort, efficiently recovering a corresponding dual pair $(\mathbf{f}_0^\star, \mathbf{g}_0^\star)$ is less straightforward. In principle, duals may be obtained by solving an elementary cascading linear system using primal-dual conditions (Peyré et al., 2019, §3.5.1). That approach does not always work, however, when the size of the support of $\mathbf{P}^\star$ is strictly smaller than $n + m - 1$ (it results in a system that has less equalities than variables), which is the case in the original ranking problem, where $n = m$. Sejourne et al. (2022, Alg.1) propose an algorithm to construct $\mathbf{f}_0^\star, \mathbf{g}_0^\star$ in $n+m$ sequential operations, interlaced with conditional statements. We consider a more generic algorithm that works in higher dimensions, but which, when particularized to the 1D case, results in the DUALSORT Algorithm 2 (see also Appendix E), a parallel approach with larger $\mathcal{O}(nm)$ complexity, but simpler to deploy on GPU, since it only requires a handful of iterations to converge, each directly comparable to that of the Sinkhorn algorithm. See application to experiments in §4.1 and §4.2.

Algorithm 2: DUALSORT Initializer

1: **Input:** Cost matrix $\mathbf{C} = [c(x_{\sigma_i}, y_{\rho_j})]$ for the sorted entries of input vectors $\mathbf{x}, \mathbf{y}$ entries, see §2.3.
2: **Initialize:** $\mathbf{f} = 0$
3: **while** not converged **do**
4: $\quad \mathbf{f} \leftarrow \min_{\text{axis=1}} \left(\mathbf{C} - \text{diag}(\mathbf{C})\mathbf{1}^T + \mathbf{f}\mathbf{1}^T\right)$
5: **end while**
6: **Return** $\mathbf{f}$

3.2 Computing Dual Initializers from Gaussian OT

From optimal potentials to dual initializers. We leverage Gaussian approximations to obtain an efficient initializer, coined GAUS, for the Sinkhorn problem, when $c(x, y) = \|x - y\|_2^2$, notably when $n \gg d$.

To do so, and given two discrete empirical measures μ and ν, compute their empirical means and covariance matrices $(\mathbf{m}_\mu, \mathbf{\Sigma}_\mu)$ and $(\mathbf{m}_\nu, \mathbf{\Sigma}_\nu)$, to recover a dual potential *function* $f^\star$ from (6) that solves the Gaussian dual OT problem, where $\mathbf{A}$ in that equation can be obtained by replacing $\mathbf{\Sigma}_1$ with $\mathbf{\Sigma}_\mu$ and $\mathbf{\Sigma}_2$ with $\mathbf{\Sigma}_\nu$. Next, evaluate that quadratic potential on all observed points of the first measure $[\mathbf{f}^{(0)}]_i \leftarrow f^\star(\mathbf{x}_i)$ (or alternatively the second measure if $m < n$) to seed the Sinkhorn algorithm.

Table 1: Toy examples, $n = m = 1024$, $d = 2$, 200 runs.

	# Iterations (mean ± std)	
Dataset	Init $\mathbf{0}$	Init Gaus
2-moons	120.0 ± 0.0	$\mathbf{11.0 \pm 0.0}$
S curve / 2-moons	137.2 ± 16.7	$\mathbf{49.6 \pm 14.8}$
3 Gaussian blobs	236.0 ± 24.3	$\mathbf{45.4 \pm 9.7}$

Complexity. Solving OT on the Gaussian approximations of μ, ν, requires computing means and covariance matrices $\mathcal{O}((n + m)d^2)$, as well as matrix square-roots and their inverse, using the Newton-Schulz iterations (Higham, 2008) at cost $\mathcal{O}(d^3)$. The GAUS initializer is therefore particularly relevant in settings where $d \ll n$, which is typically the regime where OT has found practical relevance.

Implementation. Our experiments show that GAUS often works significantly better, than the default null initialization, notably with toy datasets (see Table 1), but also when computing OT on latent space embeddings as shown in §4.3 and §4.4, or to word-embeddings as demonstrated in §4.5. The overhead induced by the computations of dual solutions is naturally dictated by the tradeoff between n (the number of points) and d (their dimension). In all cases considered here that overhead is negligible, but explored with more care in Appendix C. Note that many of the matrix-squared-roots computations can be pre-stored for efficiency, if the same measure μ is to be compared repeatedly to other measures.

3.3 Gaussian Mixture Approximations

The Gaussian initialization approach can be extended to Gaussian mixture models (GMMs), resulting in greater flexibility, yet pending further approximations. This requires the additional cost of pre-estimating GMMs for each input measure. By *further* approximations above, we refer more explicitly to the fact that, unlike for single Gaussians, we do not have access to closed-form OT solutions between GMMs, but instead only "efficient" couplings that return a

cost that is an upper-bound on the true Wasserstein distance between two GMMs, as introduced next.

OT in the space of Gaussian measures. Given two Gaussian mixtures $\rho = \sum_{k=1}^{K} \alpha_k \rho_k$ and $\tau = \sum_{k=1}^{K} \beta_k \tau_k$, assuming each ρ_k and τ_k is itself a Gaussian measure, and that weights α_k and β_k sum to 1. It was proposed in (Chen et al., 2018) to approximate the continuous OT problem between ρ and τ in the space $\mathbb{R}^d$ as a discrete OT problem in the space of mixtures of Gaussians, where each mixture is a discrete measure on K atoms (each atom being a Gaussian), and the ground cost between them is set to the pairwise Bures-Wasserstein distance, forming a cost matrix for (1) as $\mathbf{C} = [W_2^2(\rho_i, \tau_j)]_{ij}$ using (7). That optimization results in two potentials $\tilde{\mathbf{f}}$ and $\tilde{\mathbf{g}} \in \mathbb{R}^K$ that solve the corresponding regularized $K \times K$ OT problem.

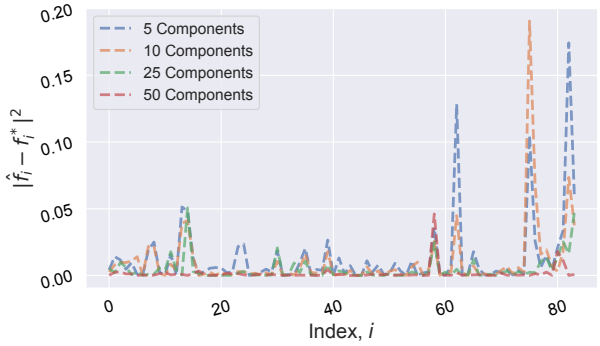


Figure 2: Gap between the true dual $\mathbf{f}^\star$ and the GMM approximate dual, for a pair of measures of word embeddings, as a function of K, the number of mixture components.

Approximating Dual Potentials with GMMs. Our proposed initializer, GMM, is computed as follows. Given two empirical measures μ, ν, we fit first two K-component GMMs τ and ρ, then obtain two potential vectors $\tilde{\mathbf{f}}, \tilde{\mathbf{g}} \in \mathbf{R}^K$ using the Sinkhorn algorithm on a $K \times K$ problem, as described above. From those potentials, we propose to compute an approximate $\hat{f}$ dual potential function:

$$\hat{f}(\mathbf{x}) = \tilde{\mathbf{f}}^T p(\mathbf{x}),\ [\,p(\mathbf{x})]_k = \frac{\alpha_k d\rho_k(\mathbf{x})}{\sum_{l=1}^{K} \alpha_l d\rho_l(\mathbf{x})}\,. \quad (8)$$

that is then evaluated on all n points of μ. Intuitively this approximation interpolates continuously the K potentials depending on probability within mixture. This recovers, in the limit where $K \to n$, n components with means $(x_i)_i$ and zero covariance, resulting in the original potential $\mathbf{f}^\star$.

Complexity. Fitting GMMs cost $\mathcal{O}(nKd^2)$. Computing the Bures-Wasserstein distances between two Gaussian measures would have complexity $\mathcal{O}(d^3)$ for full covariance matrices and $\mathcal{O}(d)$ for diagonal. Computing the cost matrix for the GMM OT problem would then amount to $\mathcal{O}(K^2 d^3)$ or $\mathcal{O}(K^2 d)$. Since naive Sinkhorn requires $\mathcal{O}(Ln^2)$ to run between pointclouds of size n for L iterations, and so the proposed GMM initialization may provide, very roughly and not taking into account pre-storage, efficiency gains when $K^2 d \ll n^2$.

3.4 Initialization via Subsampling

We next bring attention to a multi-scale approach described in detail in (Feydy, 2020, Alg. 3.6), which is a competitive baseline for comparison. Although not how originally described, this approach may be framed as a Sinkhorn initializer which we call the SUBSAMPLE initializer. The SUBSAMPLE initializer builds on the idea of the out-of-sample extrapolated entropic potentials (Pooladian and Niles-Weed, 2021) that are derived readily from a first resolution of the OT problem on a subset of points. Let $\breve{\mu}$, $\breve{\nu}$ denote uniformly subsampled measures of μ and ν of size $\breve{n} \ll n$ and $\breve{m} \ll m$. Write $\breve{\mu} = \frac{1}{\breve{n}} \sum_i \delta_{w_i}$, $\breve{\nu} = \frac{1}{\breve{m}} \sum_i \delta_{z_i}$ and write $\breve{\mathbf{f}}, \breve{\mathbf{g}}$ the optimal vector dual potentials obtained for (2) for the same regularization ε and cost, but using $\breve{\mu}$ and $\breve{\nu}$ instead. An initializer for $\mathbf{f}^{(0)}$, can then be defined by using the entropic potential function derived from $\breve{\mathbf{g}}$ (or, alternatively from $\breve{\mathbf{f}}$ if $n \ll m$):

$$[\mathbf{f}^{(0)}]_i = \breve{f}(\mathbf{x}_i), \text{with } \breve{f} : \mathbf{x} \mapsto -\varepsilon \log \frac{1}{\breve{m}} \sum_{j=1}^{\breve{m}} e^{\frac{\breve{\mathbf{g}}_j - c(\mathbf{x}, \mathbf{z}_j)}{\varepsilon}}. \quad (9)$$

Although more general than the GMM initializer, the SUBSAMPLE initializer requires running Sinkhorn on a subsample of points $\breve{n}, \breve{m}$ that is typically larger than the $K \times K$ problem induced by K-components GMMs. While this may show in runtime costs, as in Figure 7, the Sinkhorn initializer, on the other hand, not affected by large dimensions.

4 Experiments

In this section we illustrate the benefits of our proposed initialization strategies. In particular, we apply DUALSORT for differentiable sorting and soft-0/1 loss from (Cuturi et al., 2019). We investigate Gaussian (GAUS) initializers for deep differentiable clustering from (Genevay et al., 2019) and differentiable particle filtering from (Corenflos et al., 2021). Finally, we showcase GMM initializers with a document similarity task. The purpose of these experiments is to show the benefit of the initializer and not the performance in the particular task, or in claiming these tasks are original. With that in mind, we have not performed extensive network parameter tuning, though we do include some performance metrics to illustrate that the setups are reasonable. Further experimental details are given in Appendix B. Experiments were carried out using `OTT-JAX` (Cuturi et al., 2022), in particular acceleration methods for comparison, but also, when relevant, implicit differentiation of Sinkhorn's outputs.

We compare our proposed approaches to the default **0** initialization typical in most Sinkhorn implementations, in addition to fixed (Thibault et al., 2021) and adaptive (Lehmann et al., 2021) momentum, $\varepsilon - decay$, as well as Anderson acceleration (Chizat et al., 2020).

4.1 Differentiable Sorting

Arrays of size $n \in \{16, 32, 64, 128, 256, 512, 1024\}$ were sampled in this experiment from the Gaussian blob dataset (Pedregosa et al., 2011) for 200 different seeds. For each seed, 1-dimensional Gaussian data was generated from 5 random centers with centers uniformly distributed in $(-10, 10)$ with standard deviation 3. The Sinkhorn algorithm was then ran with the proposed initialization, DUALSORT, and with the default zero initializer, labelled **0**. Other Sinkhorn acceleration methods were also investigated including Anderson acceleration (And= 5), momentum (mom. = 1.05), regularization decay (ε decay = 0.8) and adaptive momentum (adapt= 10, meaning adaptation is recomputed after 10 iterations). The parameter values for these competing methods were pre-tuned following an initial hyper-parameter sweep.

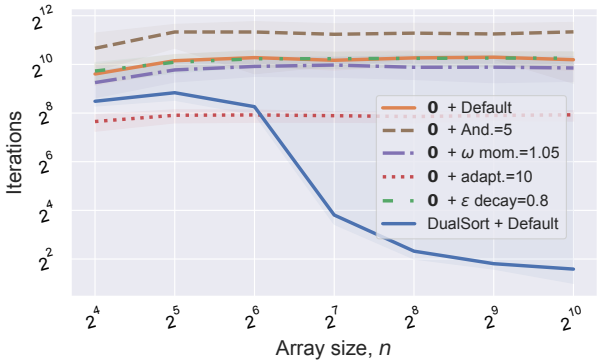


Figure 3: DUALSORT with a default Sinkhorn setup dominates all existing acceleration methods implemented when run with a default **0** initialization. We plot median, upper and lower quartiles of iterations needed to converge over 200 seeds for various array sizes (iterations for DUALSORT include steps for the primal-dual procedure).

Figures 3 and 4 illustrate the dramatic speed-up effect from using the DUALSORT procedure, with just 3 vectorized iterations. Figure 3 compares Sinkhorn algorithm with initialization to Sinkhorn enhanced through other acceleration method. Figure 4 illustrates the relative-speed up from including initialization along with other enhancements where speed-up is defined as the ratio of iterations using the zero initializer and the DUALSORT initializer, hence > 1 indicates an improvement using DUALSORT. DUALSORT complements existing acceleration methods. When the DUALSORT initializer is paired with other acceleration methods, we still observe, no matter which one is used, very large speedups.

Runtime cost. The DUALSORT initializer's runtime cost is negligible and took just 0.0012 seconds (s) to run for all experiments. The resulting absolute speed-up was $0.06s$ to 0.13s per OT problem. See further timing details in Appendices B.1. Note that this speed-up is compounded when running many thousands of OT problems.

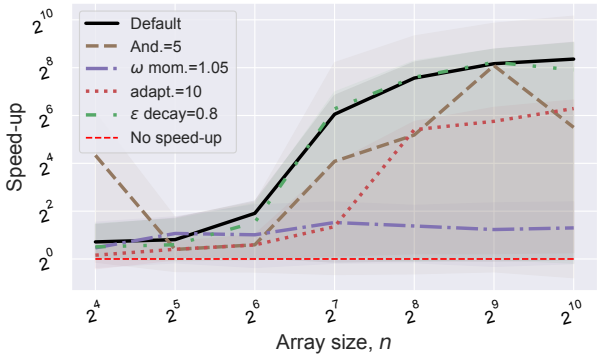


Figure 4: Relative speed-up from DUALSORT initializer (higher is better). Median, upper and lower quartiles of iterations needed to converge over 200 seeds for various array sizes.

Table 2: Average time in seconds for DualSort with 3 iterations and Sinkhorn iterations to convergence over 200 soft sorting problems for Gaussian blob data of dimension n

n	**Initializer**	**Initialization**	**Iterations**
32	**0**	-	0.28
	DualSort	0.0012	0.22
64	**0**	-	0.22
	DualSort	0.0012	0.088
128	**0**	-	0.17
	DualSort	0.0012	0.066
256	**0**	-	0.17
	DualSort	0.0012	0.049
512	**0**	-	0.13
	DualSort	0.0012	0.050
1024	**0**	-	0.14
	DualSort	0.0012	0.058

4.2 Soft Error Classification

The following experiment demonstrates the differentiability of the soft-sorting and ranking operations as well as how the DUALSORT initializer improves computational performance for real tasks. Let $h_\theta : \mathcal{X} \to \mathbb{R}^K$ be a parameterized K-label classifier and R the differentiable ranking operator described in §3.1. For input $x \in \mathcal{X}$, the soft-0/1 loss (or soft-error) evaluated at labeled (x, y), $y \leq K$, is therefore $\max(0, K - R(h_\theta(x))_y)$, see (Cuturi et al., 2019) for details.

We follow the experimental setup from (Cuturi et al., 2019). The classifier network from (Cuturi et al., 2022) is used for CIFAR-100, consisting of four CNN layers, and a fully connected hidden layer, full details given in §B.2.

The ε regularization was set to 0.01 and the network was trained until convergence over 10 seeds. DUALSORT initializer was ran with 3 iterations, which, as discussed in §3.1, is slightly cheaper than two Sinkhorn iterations.

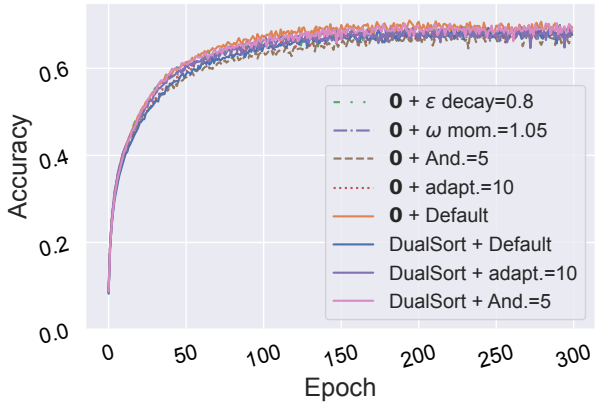


Figure 5: Accuracy of CNN classifier by Sinkhorn methods for CIFAR-100 with soft-error loss and $\varepsilon = 0.01$

Table 3: Soft-Error: CIFAR 100, mean ± std of Sinkhorn iter./ training step, over 10 seeds

	Iterations	Runtime ($\times 10^{-2}$s)
Zero	17.9 ± 0.1	8.23 ± 0.2
Anderson	12.3 ± 0.2	5.74 ± 0.2
Momentum	15.7 ± 0.2	7.70 ± 0.2
Adaptive	15.2 ± 0.2	7.38 ± 0.3
ε-decay	17.0 ± 0.1	7.99 ± 0.2
DUALSORT	9.7 ± 0.1	5.07 ± 0.3
DUALSORT, Adap.	10.3 ± 0.1	5.27 ± 0.3
DUALSORT, Ande.	$\mathbf{8.2 \pm 0.1}$	$\mathbf{3.72 \pm 0.3}$

Accuracy on the evaluation set is shown in Figure 5 for 300 epochs. It is clear that, as expected, the Sinkhorn initialization procedure does not affect training nor accuracy. However, Table 3 shows that the DUALSORT initializer drastically reduces the number of Sinkhorn iterations needed for convergence, to compute the soft-error loss and its gradients at each evaluation.

4.3 Differentiable Clustering

We demonstrate the performance improvement from the Gaussian initializer on the task of deep differentiable clustering, with the experimental setup of (Genevay et al., 2018). Differentiable clustering aims at producing a latent representation amenable to clustering. This is achieved using a variational autoencoder (Kingma et al., 2014) with learnable, discrete cluster embeddings, and an additional loss term allocating encodings to cluster embeddings using OT.

For data of dimension d_x and latent dimension d_z, let $E_\theta : \mathbb{R}^{d_x} \to \mathbb{R}^{2\times d_z}$ and $D_\theta : \mathbb{R}^{d_z} \to \mathbb{R}^{d_x}$ denote an encoder and decoder respectively, parameterized by θ. Let $\mu_\phi \in \mathbb{R}^{K\times d_z}$ denote cluster embeddings for K clusters. The objective of differentiable clustering is to learn E_θ, D_θ and embeddings $\mu_\phi \in \mathbb{R}^{K\times d_z}$. This may be achieved by minimizing the loss $\ell^{\text{ae}}(\theta) + \ell^{\text{OT}}(\phi, \theta)$ for each batch of data $(x_i)_i$. Here $\ell^{\text{ae}}(\theta)$ is the standard variational auto-encoder loss and $\ell^{\text{OT}}(\phi, \theta)$ is the regularized OT loss from (1) between $\mu = \sum_{k=1}^K \frac{1}{K}\delta_{\mu_{\phi k}}$ and $\nu = \sum_{i=1}^n \frac{1}{n}\delta_{z_i}$. $z_i = \mathbf{m}_i + \sigma_i u_i$, where $(\mathbf{m}_i, \sigma_i) = E_\theta(x_i)$, $u_i \sim \mathcal{N}(\mathbf{0}_{d_z}, \mathbf{I}_{d_z})$, and $\tilde{x}_i = D_\theta(z_i)$.

Table 4: Avg. Sinkhorn iter./training step and runtime / training step mean ± std for differentiable clustering VAE, 10 seeds, $\epsilon = 0.001$

	Iterations	Runtime ($\times 10^{-3}$ s)
Zero	354.1 ± 7.0	25.4 ± 0.2
ε-decay	340.5 ± 17.8	25.1 ± 0.1
Anderson	844.4 ± 26.2	144 ± 6.7
Momentum	342.5 ± 3.7	33.1 ± 1.7
Adaptive	96.6 ± 4.1	9.35 ± 0.02
Gaus	196.6 ± 6.7	16.2 ± 0.6
Gaus, Adapt.	$\mathbf{68.7 \pm 1.3}$	$\mathbf{8.00 \pm 0.1}$

We demonstrate this task for MNIST (Deng, 2012) over 10 seeds. Fully connected networks with 4 hidden layers were used for E_θ and D_θ, where $d_z = 32$ and $d_x = 784$, further experimental details are given in §B.3. Table 4 shows that the Gaussian initializer outperforms the zero initialization for default Sinkhorn and all other combinations of default Sinkhorn plus acceleration techniques. Performance metrics and samples from the generative model are given in Appendix B.3.

4.4 Differentiable Particle Filtering

As introduced in Corenflos et al. (2021), the Sinkhorn algorithm provides an approximate differentiable resampling scheme, hence enables end-to-end differentiable particle filtering. Consider a simple linear state space model consisting of latent states $x_t \in \mathbb{R}^2$ where $x_0 = \mathbf{0}$, $X_t|x_{t-1} \sim f(\cdot|x_{t-1}) = \mathcal{N}(0.5\mathbb{I}x_{t-1}, \mathbb{I})$ and observations $y_t \in \mathbb{R}^2$, $y_t \sim g(\cdot|x_t) = \mathcal{N}(x_t, \mathbb{I})$ for $t \in \{1, \dots, T\}$, and time series length $T = 500$. Differentiable resampling via OT consists of applying the Sinkhorn algorithm between weighted and unweighted pointclouds of N simulated latent states at each timepoint t, for each forward pass. For full details see Corenflos et al. (2021).

For batch size $B = 4$ and time steps $T = 500$, under a naive implementation, each forward pass requires $T \times B$ Sinkhorn layers evaluations. This can be quite slow. As shown in Table 5, the Gaussian initializer is effective at reducing the runtime by reducing the number of Sinkhorn iterations by approximately 33% to 50% relative to the default Sinkhorn with $\mathbf{0}$ initialization.

4.5 Document Similarity

In this experiment, we compare the GAUS, GMM and SUBSAMPLE initializers. Documents were gathered from the 20 *Newsgroup* dataset (Lang, 1995) and each word, $(w_i)_{i=1}^n$, in

Table 5: Mean ± std number of Sinkhorn iterations and runtime over 3 seeds for the forward pass of a particle filter with N particles, batch size 4 of simple linear state space model, $T = 500$.

N	Initializer	Iterations ('000s)	Runtime /s
32	Gaus	440 ± 2.5	12.08 ± 0.25
	0	611 ± 3.4	15.46 ± 0.35
64	Gaus	349 ± 2.9	9.62 ± 1.29
	0	532 ± 3.4	12.49 ± 0.69
128	Gaus	269 ± 0.7	7.03 ± 1.21
	0	471 ± 2.3	10.18 ± 0.88
256	Gaus	216 ± 1.5	6.340.78
	0	439 ± 1.9	11.01 ± 0.59
512	Gaus	176 ± 1.3	14.43 ± 1.40
	0	422 ± 1.7	30.17 ± 1.06

the vocabulary across documents is embedded using the pre-trained GloVe word embeddings (Pennington et al., 2014) as $(e_i)_{i=1}^n$ where $e_i \in \mathbb{R}^{50}$.

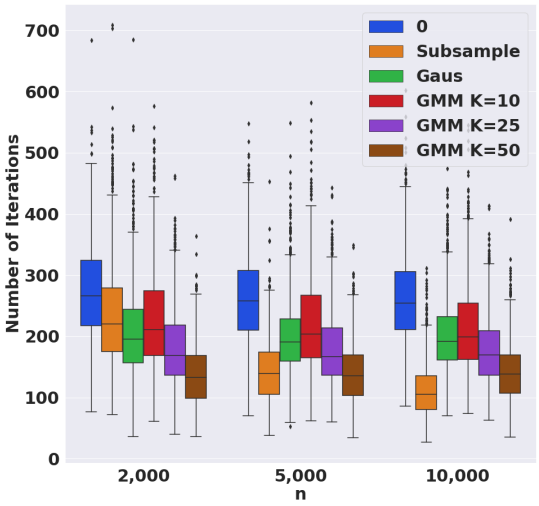


Figure 6: Distribution of number of Sinkhorn iterations required for Sinkhorn convergence between 1225 pairs of Newsgroup documents, represented as word embeddings histograms, n being the total vocabulary size. The same convergence threshold for Sinkhorn is used for all n.

In a similar setup to Kusner et al. (2015), each document may be represented as a histogram with weights $(a_i)_{i=1}^n$ corresponding to word-frequency, $\nu_i = \sum_{i=1}^n a_i \delta_{e_i}$, and we compute pairwise OT distances between 50 documents, resulting in $1,225$ pairs. We report the number of Sinkhorn iterations and runtime required for convergence for the default zero intializer (**0**), the proposed GAUS initializer, the GMM initializers with full covariance matrices and $K \in \{10, 25, 50\}$ components, and the SUBSAMPLE initializer. A subset of the vocabulary of size $n \in \{2 \times 10^3, 5 \times 10^3, 10^4\}$ was used, and corresponding subsample of size $100, 500$ and $1,000$ for the SUBSAMPLE initializer. Regularization was $\varepsilon = 0.001$.

The distribution of results are shown in Figure 6 and Figure 7 illustrating that improvements can be obtained for a range of K. Notice however that GAUS beats the GMM for low K, we suspect this is due to the additional approximation (8). Although often resulting in lower number of fine-tuning Sinkhorn iterations, the preprocessing cost of running the SUBSAMPLE initializer is expensive, and only exhibits better aggregate runtime performance for large $n = 10,000$, which was expected. A GMM was first fitted to each document, before being used for initializing Sinkhorn potentials. As Figure 7 shows, although the cost of fitting GMMs results in limited runtime savings for $n = 2,000$, there are significant runtime savings for $n = 5,000$ and $n = 10,000$. See further discussion in §C.

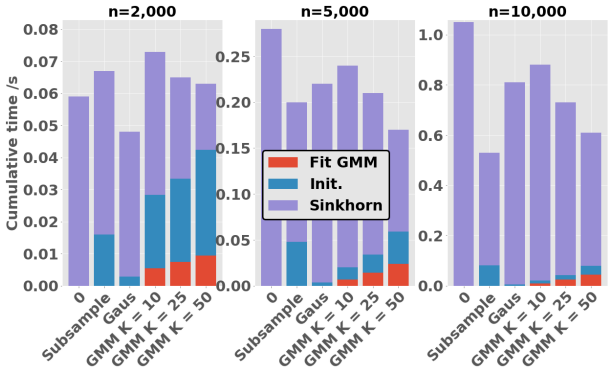


Figure 7: Average wall clock time for computing OT between each pair of word-embeddings ($1,225$ problems) for vocabulary of size $n = 2 \times 10^3; 5 \times 10^3; 10^4$; split by initialization time (Init), time to compute Gaussian mixture models (Fit GMM) and Sinkhorn iterations (Sinkhorn).

5 Conclusion

We have introduced efficient and robust Sinkhorn potential initialization schemes: DUALSORT, GAUS, GMM and demonstrated how these carefully chosen initializers can significantly improve the performance of the Sinkhorn algorithm for a variety of tasks. These GPU-friendly initializers may also be embedded in end-to-end differentiable procedures by relying on implicit differentiation, as demonstrated in various tasks presented in our experiments (ranking, clustering, filtering), and are complementary to most common acceleration methods, creating an interesting space to optimize further the execution of Sinkhorn. Initialization is a neglected area of computational OT, and we hope that these promising results can inspire new research to other areas, such as initalizing calls to Sinkhorn in the internal loops of the Gromov-Wasserstein or barycenter problem. We also hope they can help extending OT's reach to data-hungry application areas, such as single-cell or NLP tasks that involve typically a large number of samples.

References

Adams, R. P. and Zemel, R. S. (2011). Ranking via sinkhorn propagation. *arXiv preprint arXiv:1106.1925*.

Ahuja, R. K., Magnanti, T. L., and Orlin, J. B. (1988). Network flows.

Altschuler, J., Bach, F., Rudi, A., and Niles-Weed, J. (2019). Massively scalable sinkhorn distances via the nyström method. *Advances in neural information processing systems*, 32.

Amos, B., Cohen, S., Luise, G., and Redko, I. (2022). Meta optimal transport.

Anderson, D. G. (1965). Iterative procedures for nonlinear integral equations. *Journal of the ACM (JACM)*, 12(4):547–560.

Bertsimas, D. and Tsitsiklis, J. N. (1997). *Introduction to linear optimization*, volume 6. Athena Scientific Belmont, MA.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., et al. (2018). Jax: composable transformations of python+ numpy programs. *Version 0.2*, 5:14–24.

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

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. (2020). Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924.

Chen, Y., Georgiou, T. T., and Tannenbaum, A. (2018). Optimal transport for gaussian mixture models. *IEEE Access*, 7:6269–6278.

Chiappori, P.-A., McCann, R. J., and Pass, B. (2017). Multi- to one-dimensional optimal transport. *Communications on Pure and Applied Mathematics*, 70(12):2405–2444.

Chizat, L., Roussillon, P., Léger, F., Vialard, F.-X., and Peyré, G. (2020). Faster wasserstein distance estimation with the sinkhorn divergence. *Advances in Neural Information Processing Systems*, 33:2257–2269.

Corenflos, A., Thornton, J., Deligiannidis, G., and Doucet, A. (2021). Differentiable particle filtering via entropy-regularized optimal transport. In *International Conference on Machine Learning*, pages 2100–2111. PMLR.

Courty, N., Flamary, R., Habrard, A., and Rakotomamonjy, A. (2017). Joint distribution optimal transportation for domain adaptation. *Advances in Neural Information Processing Systems*, 30.

Courty, N., Flamary, R., and Tuia, D. (2014). Domain adaptation with regularized optimal transport. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 274–289. Springer.

Cuturi, M. (2013). Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26.

Cuturi, M., Meng-Papaxanthos, L., Tian, Y., Bunne, C., Davis, G., and Teboul, O. (2022). Optimal transport tools (ott): A jax toolbox for all things wasserstein. *arXiv preprint arXiv:2201.12324*.

Cuturi, M. and Peyré, G. (2015). A smoothed dual approach for variational wasserstein problems. *arXiv preprint arXiv:1503.02533*.

Cuturi, M., Teboul, O., Niles-Weed, J., and Vert, J.-P. (2020). Supervised quantile normalization for low rank matrix factorization. In *International Conference on Machine Learning*, pages 2269–2279. PMLR.

Cuturi, M., Teboul, O., and Vert, J.-P. (2019). Differentiable ranking and sorting using optimal transport. *Advances in neural information processing systems*, 32.

Dantzig, G. B., Ford Jr, L. R., and Fulkerson, D. R. (1956). A primal–dual algorithm. Technical report, RAND CORP SANTA MONICA CA.

Deng, L. (2012). The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142.

d'Aspremont, A., Scieur, D., Taylor, A., et al. (2021). Acceleration methods. *Foundations and Trends® in Optimization*, 5(1-2):1–245.

Feydy, J. (2020). *Geometric data analysis, beyond convolutions*. PhD thesis, Université Paris-Saclay Gif-sur-Yvette, France.

Flamary, R., Courty, N., Gramfort, A., Alaya, M. Z., Boisbunon, A., Chambon, S., Chapel, L., Corenflos, A., Fatras, K., Fournier, N., et al. (2021). Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8.

Genevay, A., Dulac-Arnold, G., and Vert, J.-P. (2019). Differentiable deep clustering with cluster size constraints. *arXiv preprint arXiv:1910.09036*.

Genevay, A., Peyré, G., and Cuturi, M. (2018). Learning generative models with sinkhorn divergences. In *International Conference on Artificial Intelligence and Statistics*, pages 1608–1617. PMLR.

Hashimoto, T., Gifford, D., and Jaakkola, T. (2016). Learning Population-Level Diffusions with Generative Recurrent Networks. volume 33.

Higham, N. J. (2008). *Functions of matrices: theory and computation*. SIAM.

Janati, H., Bazeille, T., Thirion, B., Cuturi, M., and Gramfort, A. (2020). Multi-subject meg/eeg source imaging with sparse multi-task regression. *NeuroImage*, 220:116847.

Kingma, D. P., Mohamed, S., Jimenez Rezende, D., and Welling, M. (2014). Semi-supervised learning with deep

generative models. *Advances in neural information processing systems*, 27.

Kosowsky, J. and Yuille, A. L. (1994). The invisible hand algorithm: Solving the assignment problem with statistical physics. *Neural networks*, 7(3):477–490.

Kusner, M., Sun, Y., Kolkin, N., and Weinberger, K. (2015). From word embeddings to document distances. In *International conference on machine learning*, pages 957–966. PMLR.

Lang, K. (1995). Newsweeder: Learning to filter netnews. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 331–339.

Lehmann, T., Von Renesse, M.-K., Sambale, A., and Uschmajew, A. (2021). A note on overrelaxation in the sinkhorn algorithm. *Optimization Letters*, pages 1–12.

Luise, G., Rudi, A., Pontil, M., and Ciliberto, C. (2018). Differential properties of sinkhorn approximation for learning with wasserstein distance. *Advances in Neural Information Processing Systems*, 31.

Meyron, J. (2019). Initialization procedures for discrete and semi-discrete optimal transport. *Computer-Aided Design*, 115:13–22.

Monge, G. (1781). Mémoire sur la théorie des déblais et des remblais. *Histoire de l'Académie Royale des Sciences*, pages 666–704.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Peyré, G., Cuturi, M., et al. (2019). Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607.

Pooladian, A.-A. and Niles-Weed, J. (2021). Entropic estimation of optimal transport maps.

Salimans, T., Zhang, H., Radford, A., and Metaxas, D. (2018). Improving GANs using optimal transport. In *International Conference on Learning Representations*.

Sander, M. E., Ablin, P., Blondel, M., and Peyré, G. (2022). Sinkformers: Transformers with doubly stochastic attention. In *International Conference on Artificial Intelligence and Statistics*, pages 3515–3530. PMLR.

Santambrogio, F. (2015). *Optimal transport for applied mathematicians*. Birkhauser.

Sarlin, P.-E., DeTone, D., Malisiewicz, T., and Rabinovich, A. (2020). Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Scetbon, M. and Cuturi, M. (2020). Linear time sinkhorn divergences using positive features. *Advances in Neural Information Processing Systems*, 33:13468–13480.

Schiebinger, G., Shu, J., Tabaka, M., Cleary, B., Subramanian, V., Solomon, A., Gould, J., Liu, S., Lin, S., Berube, P., et al. (2019). Optimal-Transport Analysis of Single-Cell Gene Expression Identifies Developmental Trajectories in Reprogramming. *Cell*, 176(4).

Schmitz, M. A., Heitz, M., Bonneel, N., Ngole, F., Coeurjolly, D., Cuturi, M., Peyré, G., and Starck, J.-L. (2018). Wasserstein dictionary learning: Optimal transport-based unsupervised nonlinear dictionary learning. *SIAM Journal on Imaging Sciences*, 11(1):643–678.

Schmitzer, B. (2019). Stabilized sparse scaling algorithms for entropy regularized transport problems. *SIAM Journal on Scientific Computing*, 41(3):A1443–A1481.

Sejourne, T., Vialard, F.-X., and Peyré, G. (2022). Faster unbalanced optimal transport: Translation invariant sinkhorn and 1-d frank-wolfe. In Camps-Valls, G., Ruiz, F. J. R., and Valera, I., editors, *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pages 4995–5021. PMLR.

Sinkhorn, R. (1967). Diagonal equivalence to matrices with prescribed row and column sums. *American Mathematical Monthly*, 74:402–405.

Solomon, J., De Goes, F., Peyré, G., Cuturi, M., Butscher, A., Nguyen, A., Du, T., and Guibas, L. (2015). Convolutional Wasserstein distances: efficient optimal transportation on geometric domains. *ACM Transactions on Graphics*, 34(4):66:1–66:11.

Thibault, A., Chizat, L., Dossal, C., and Papadakis, N. (2021). Overrelaxed sinkhorn–knopp algorithm for regularized optimal transport. *Algorithms*, 14(5):143.

Xie, Y., Dai, H., Chen, M., Dai, B., Zhao, T., Zha, H., Wei, W., and Pfister, T. (2020a). Differentiable top-k with optimal transport. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 20520–20531. Curran Associates, Inc.

Xie, Y., Wang, X., Wang, R., and Zha, H. (2020b). A fast proximal point method for computing exact wasserstein distance. In *Uncertainty in artificial intelligence*, pages 433–453. PMLR.

A Dual Potential Comparison

For balanced OT problems, as considered here, Dual potentials $\mathbf{f}, \mathbf{g}$ are unique up to constant shifts i.e. $\mathbf{f} - s, \mathbf{g} + s$ for $s \in \mathbb{R}$. Therefore, in order to compare potentials $\mathbf{f} \in \mathbb{R}^n$, we center $\mathbf{f}$, as $\mathbf{f} \leftarrow \mathbf{f} - \frac{1}{n}\sum_i \mathbf{f}_i$.

A.1 From Optimal Primal to Dual Vectors

Properties of the optimal primal $\mathbf{P}^\star$. Taking the 1D case as motivation, we introduce a method to recover optimal dual potentials $\mathbf{f}^\star, \mathbf{g}^\star$ from an optimal primal solution $\mathbf{P}^\star$. To that end, one can cast an OT problem as a min-cost-flow problem on a bipartite graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with vertices composed of source nodes $S = \{1, \dots, n\}$ and target nodes $T = \{1', \dots, m'\}$, $\mathcal{V} = S \cup T$, and edge set $\mathcal{E} = \{(i, j'), i = 1, \dots, n; j = 1, \dots, m\}$ linking them. The KKT conditions state that, writing $\mathcal{E}(\mathbf{P}) = \{(i, j') | \mathbf{P}_{i,j} > 0\}$ one has that the graph $(\mathcal{V}, \mathcal{E}(\mathbf{P}^\star))$ is necessarily a forest (Peyré et al., 2019, Prop. 3.4).

We write $\mathcal{T}_1, \dots, \mathcal{T}_K$ for the K trees forming that forest, where $1 \leq K \leq \min(n, m)$, and write t_k for their size.

We use the lexicographic order to define the root node of each tree, chosen to be the smallest *source* node $s(k)$ contained in $\mathcal{T}_k$. For convenience, we assume that trees are ordered following $s(k)$, and therefore that $\mathcal{T}_1$ has 1 as its root node. For each tree k, we introduce $p^k = (p^k_1, \dots, p^k_{t_k - 1})$ for a pre-order breadth-first-traversal of $\mathcal{T}_k$ originating at $s(k)$, enumerating $t_k - 1$ edges, namely pairs in $S \times T$ or $T \times S$, guaranteed to be such that any parent node in the tree is visited before its descendants. $\iota(j)$ denotes the smallest source index i such that $(i, j') \in \mathcal{E}(\mathbf{P}^\star)$.

Algorithm 3: Recover dual from primal

Input: Cost matrix $\mathbf{C}$ and graph $(\mathcal{V}, \mathcal{E}(\mathbf{P}^\star))$
Initialize: $\mathbf{f} = 0$.
while not converged **do**
  for $k \in \{2, \dots, K\}$ **do**
    $\mathbf{f}_{s(k)} \leftarrow \min_j c_{s(k),j} - c_{\iota(j),j} + \mathbf{f}_{\iota(j)}$
  end for
  for $k \in \{1, \dots, K\}$ **do**
    $\mathbf{f} \leftarrow \textsc{UpdateTree}(\mathbf{C}, \mathbf{f}, k)$
  end for
end while
Return $\mathbf{f}$

Algorithm 4: UPDATETREE

Input: Cost matrix $\mathbf{C}$, $\mathbf{f}$, tree index k
for $e = (a, b) \in p^k$ **do**
  if $a \in S, b \in T$ **then**
    $g \leftarrow c_{a,b} - \mathbf{f}_a$
  else
    $\mathbf{f}_a \leftarrow c_{a,b} - g$
  end if
end for
Return $\mathbf{f}$

Complementary and Feasibility Constraints. Complementary slackness provides a set of $n + m - K$ linear equations (10), while feasibility constraints are given in (11).

$$(i, j') \in \mathcal{E}(\mathbf{P}^\star) \Leftrightarrow \mathbf{f}^\star_i + \mathbf{g}^\star_j = c_{i,j} \ , \tag{10}$$

$$\forall i \leq n, j \leq m, \ \mathbf{f}_i + \mathbf{g}_j \leq c_{i,j} \, . \tag{11}$$

For the special case $K = 1$, which happens for instance when n and m are co-primes and weights are uniform, the set of linear equations (10) suffices to recover the $n + m$ dual variables, with the convention that the first entry be 0. When, on the contrary, $K > 1$, that set of $n + m - K$ equations is no longer sufficient. For example, $K = n = m$ for the optimal assignment

problem, in which $(\mathcal{V}, \mathcal{E}(\mathbf{P}^\star))$ describes a set of n isolated trees, and only n equality relations are available for $2n$ variables. In such cases, one must additionally use the feasibility constraint (11) to obtain optimal dual variables (Peyré et al., 2019, Prop 3.3).

The c-transform $\mathbf{g}_i^c := \min_j c_{i,j} - \mathbf{g}_j$ can be used to enforce constraints (11), however, it may no longer satisfy the complementary condition (10). This is remedied by updating all source nodes i in tree k by starting from $s(k)$ as detailed in Algorithm 4. Repeated application of these updates, Algorithm 3, guarantees convergence.

Lemma 1. *Given the optimal coupling matrix $\mathbf{P}^\star$ solving OT problem* (1) *with $\varepsilon = 0$, the procedure defined in Algorithm 3 converges to the optimal dual potentials for dual problem* (5)*.*

The proof is provided in §E, and uses the fact that Algorithm 3 is a primal-dual method (Dantzig et al., 1956), tweaked because the primal solution $\mathbf{P}^\star$ is known.

B Further Experimental Details

B.1 Differentiable Sorting Details

Regularization $\varepsilon = 0.01$ was used, as per (Cuturi et al., 2019). In this experiment arrays of size $n \in \{16, 32, 64, 128, 256, 512, 1024\}$ were sampled from the Gaussian blob dataset (Pedregosa et al., 2011) for 200 different seeds. At each seed, 1-dimensional Gaussian data is generated from 5 random centers with centers uniformly distributed in $(-10, 10)$ with standard deviation 3.

Baseline acceleration methods (Anderson acceleration, momentum, adaptive momentum, ϵ decay) were considered to augment the Sinkhorn algorithm, using the implementations from (Cuturi et al., 2022). The momentum hyper-parameter ω was set at 1.05 from a grid search of $\{0.8, 1.05, 1.1, 1.3\}$. Adaptive momentum consists of adjusting the momentum parameters every *adapt_iters* number of iterations where *adapt_iters* was set to 10 from a search on $\{10, 20, 50, 200\}$. ϵ decay consisted of gradually reducing the regularization term from 5ϵ to ϵ by a factor of 0.8, from a search of decay factors from $\{0.8, 0.95\}$. The Anderson acceleration parameter was set to 5 from a search on $\{3, 5, 8, 10, 15\}$.

B.2 Soft Error Details

Regularization $\epsilon = 0.01$ was used for the soft-error task. The soft 0/1 error objective described in (Cuturi et al., 2019) was used, with a neural network classifier consisting of two CNN blocks with 32 and 64 features respectively, and a hidden layer of hidden size 512. Each CNN block consists of two CNN layers with 3×3 kernel, relu activations between CNN layers and a max pooling layer at the end of each block. Implementation including neural network architecture was taken from (Cuturi et al., 2022)[1]. Our proposed method was compared to other acceleration baselines using the same grid of hyperparameters as described in §B.1. Batch size was set to 64 and learning rate 0.001.

B.3 Differentiable Clustering Details

The experiment was repeated for $\epsilon = 0.1$ and $\epsilon = 0.01$ and again compared to other acceleration baselines using the same grid of hyperparameters as described in §B.1. Batch size was set to 256 and learning rate 0.001.

Latent dimension was set to $d_z = 32$ and MNIST (Deng, 2012) images are of size $d_x = 28 \times 28$.The decoder $D_\theta : \mathbb{R}^{d_x} \to \mathbb{R}^{2 \times d_z}$ consists of 4 hidden $[512, 512, 256, 256]$ followed by a final linear layer converting the outputted embedding to a vector of dimension 784. The encoder $E_\theta : \mathbb{R}^{d_x} \to \mathbb{R}^{2 \times d_z}$ consists of 4 hidden layers of depths $[512, 512, 256, 256]$ with relu activations, the final embeddings is mapped to $\mathbf{m}_i \in \mathbb{R}^{d_z}$ and $logvar_i \in \mathbb{R}^{d_z}$ by two separate linear layers without activations, where $\sigma_i = \exp(0.5 \times logvar_i)$. For batch $(x_i)_i$, the standard VAE loss $\ell^{ae}(\theta) = \sum_i \|x_i - \tilde{x}_i\|_2^2 - 0.5 \sum_i (1 + 2 * \log(\sigma_i) - \mathbf{m}_i^2 - \sigma_i^2)$. Recall $\tilde{x}_i = D_\theta(z_i)$ and $z_i = \mathbf{m}_i + \sigma_i u_i$, $u_i \sim \mathcal{N}(\mathbf{0}_{d_z}, \mathbf{I}_{d_z})$.

As discussed in (Genevay et al., 2019), clusters may be used as an unsupervised classifier and accuracy is reported in Table 6, illustrating that the clusters are meaningful. In addition, samples from the clustered latent space may be used to generate new samples as a form of conditional generation, again shown in Figure 8.

Accuracy for each cluster is defined as in (Genevay et al., 2019), as follows. Accuracy for label l in cluster k is by $acc_{l,k} = \frac{\sum_i \mathbf{I}_{y_i == l, \tilde{y}_i == k}}{\sum_i \mathbf{I}_{y_i == k}}$ where $\tilde{y}_i = \arg\min_k \|z_i - \mu_{\phi,k}\|_2^2$ and y_i is the true label of x_i. We write the top label accuracy for

[1] https://github.com/ott-jax/ott/tree/main/ott/examples/soft_error

Figure 8: Generated Samples

each cluster k as $\max_l acc_{l,k}$. When using 10 clusters for 10 labels for MNIST, each cluster's top label accuracy corresponds to a different label, one cluster for each digit. Table 6 shows that the clusters manage to capture geometrically meaningful information corresponding to each label.

Table 6: Evaluation Accuracy of trained clustered VAE for MNIST

Digit	0	1	2	3	4	5	6	7	8	9
Accuracy	0.91	0.66	0.42	0.56	0.80	0.61	0.68	0.64	0.90	0.78

C Overhead Analysis

Although timings are highly dependent on hardware and implementation, we provide some experimental examples running on a single V100 GPU and 4 CPUs. This shows that the time overhead for DualSort and Gaussian initializers are inconsequential relative to speed-up in terms of both time and iteration count for the savings in Sinkhorn iterations. The Gaussian mixture model (GMM) is computationally more expensive than the other proposed initializers, however the table below shows that it can also result in time savings.

C.1 Differentiable Sorting

Table 7: Average time in seconds for DualSort with 3 iterations and Sinkhorn iterations to convergence over 200 soft sorting problems of dimension n

n	**Initializer**	**Initialization**	**Iterations**
32	**0**	-	0.28
	DualSort	0.0012	0.22
64	**0**	-	0.22
	DualSort	0.0012	0.088
128	**0**	-	0.17
	DualSort	0.0012	0.066
256	**0**	-	0.17
	DualSort	0.0012	0.049
512	**0**	-	0.13
	DualSort	0.0012	0.050
1024	**0**	-	0.14
	DualSort	0.0012	0.058

It can be seen that the DualSort initialization procedure is extremely efficient and does not have significant impact on the total run-time. The timings above are averaged per OT problem over 200 runs with different seeds.

C.2 Gaussian and GMM

In this section we consider timings for the word embedding/ document similarity experiment.

For the GMM initializer, the *pre-compute* is the average time to compute each GMM (1 per document), divided by the number of OT problems. Each GMM is reused multiple times, so the cost is split. Each GMM was computed using scikit-learn (Pedregosa et al., 2011) on CPU, for lack of a convenient GPU implementation. There exists open-source GPU implementations [2] of Gaussian mixture models for diagonal component covariance matrices which are significantly faster, and may be worth further investigation for more efficient implementation. Similarly, one may amortize inference in GMMs or provide a warm-start from a pooled GMM to initialize fitting the GMM. We use the default K-means initializer from scikit learn. The *Initialization* field reports the time to compute the approximate dual potentials given the GMM parameters.

For the Gaussian initializer, the mean and variance parameters are inexpensive to compute, hence were not computed and cached but instead computed repeatedly on the fly for each OT problem. Hence the total initialization compute time is reported in the *Initialization* column. Further computational savings could be made by caching the Gaussian parameters for each document. Note that the dimension for the Gaussian OT approximation is $d = 50$ and given the Gaussian initialization is negligible here, it would also be negligible for lower dimensional settings.

Table 8: Time, in seconds, per OT problem split by task, averaged over $1,225$ OT problems, from each pair of 50 documents from the Newsgroup 20 dataset with a subset of vocabulary of size n.

n	**Initializer**	**Pre-compute**	**Initialization**	**Sinkhorn Iter.**	**Total**
$2,000$	$\mathbf{0}$	-	-	0.059	0.059
	Subsample	-	0.016	0.051	0.067
	Gaus	-	0.0028	0.045	0.048
	GMM $K = 10$	0.0027	0.023	0.047	0.073
	GMM $K = 25$	0.0037	0.026	0.035	0.065
	GMM $K = 50$	0.0047	0.033	0.027	0.063
$5,000$	$\mathbf{0}$	-	-	0.28	0.28
	Subsample	-	0.048	0.15	0.20
	Gaus	-	0.0036	0.22	0.22
	GMM $K = 10$	0.0035	0.013	0.23	0.24
	GMM $K = 25$	0.0070	0.030	0.18	0.22
	GMM $K = 50$	0.012	0.035	0.13	0.17
$10,000$	$\mathbf{0}$	-	-	1.05	1.05
	Subsample	-	0.082	0.45	0.53
	Gaus	-	0.0053	0.81	0.81
	GMM $K = 10$	0.0042	0.013	0.86	0.88
	GMM $K = 25$	0.012	0.019	0.70	0.73
	GMM $K = 50$	0.022	0.035	0.56	0.62

D Gaussian Potential

In this section we derive explicitly the Gaussian potential. The transport map T solving the Monge problem (4) from a non-degenerate Gaussian measure $\mu = \mathcal{N}(\mathbf{m}_\mu, \mathbf{\Sigma}_\mu)$ to another Gaussian $\nu = \mathcal{N}(\mathbf{m}_\nu, \mathbf{\Sigma}_\nu)$ can be recovered in closed-form as $T^\star(x) := \mathbf{A}(x - \mathbf{m}_\mu) + \mathbf{m}_\nu$, where $\mathbf{A} = \mathbf{\Sigma}_\mu^{-\frac{1}{2}}(\mathbf{\Sigma}_\mu^{\frac{1}{2}}\mathbf{\Sigma}_\nu\mathbf{\Sigma}_\mu^{\frac{1}{2}})^{\frac{1}{2}}\mathbf{\Sigma}_\mu^{-\frac{1}{2}}$, see e.g. (Peyré et al., 2019, Chapter 2.6) for a discussion. Brenier's theorem (Brenier, 1987) states that for cost $c : (x, y) \to \frac{\|x-y\|^2}{2}$ this map is uniquely defined as the gradient of a convex function φ, and it can be verified that $T^\star(x) = \nabla\varphi(x)$ where $\varphi(x) = \frac{1}{2}(x - m_\mu)^T\mathbf{A}(x - m_\mu) + m_\nu^T x$.

The convex function $\varphi(x)$ is related to dual potential f through $\varphi(x) = \frac{\|x\|^2}{2} - f(x)$ hence

$$f^\star(x) = \frac{\|x\|^2}{2} - \frac{1}{2}(x - m_\mu)^T\mathbf{A}(x - m_\mu) - m_\nu^T x.$$

For cost $c : (x, y) \to \|x - y\|^2$, the optimal potential is therefore

$$f^\star(x) = \|x\|^2 - (x - m_\mu)^T\mathbf{A}(x - m_\mu) - 2m_\nu^T x.$$

[2] https://github.com/borchero/pycave

E Convergence of Sorting Initializer and DualSort Details

E.1 Proof of Primal Dual Convergence

Recovering optimal dual potentials corresponding to the primal solution is equivalent to finding any vector of shortest paths $\mathbf{f}$ from a single node e.g. node 1, in the network to each of the other nodes, see e.g. (Bertsimas and Tsitsiklis, 1997, Theorem 7.17) and (Ahuja et al., 1988, Chapter 9).

Algorithm 3 computes the shortest path using a particular case of a method known as *label correcting* (Bertsimas and Tsitsiklis, 1997, Chapter 7). Given there are no cycles, the proposed method recovers the shortest path by (Bertsimas and Tsitsiklis, 1997, Theorem 7.18) and hence recovers the optimal dual potentials.

Algorithm 3 exploits the primal solution efficiently by correcting all nodes in the same tree, hence the iterations are dependent on the number of trees and not necessarily the number of nodes.

The minimization step, $\mathbf{f}_{s(k)} \leftarrow \min_j c_{s(k),j} - c_{\iota(j),j} + \mathbf{f}_{\iota(j)}$ follows traditional label correcting methods. However, a key insight is updating nodes along tree of $s(k)$ is equivalent to updating the minimum path to each node in the tree.

$\mathbf{f}_i$ is the shortest path to node i if $\mathbf{f}_i \leq c_{i,j} - c_{\iota(j),j} + \mathbf{f}_{\iota(j)}\ \forall j$, which is equivalent to $\mathbf{f}_i + \mathbf{g}_j \leq c_{i,j}$ and may be interpreted as $\mathbf{f}_i$ being less than the route to any other source node $\mathbf{f}_{\iota(j)}$ then to $\mathbf{f}_i$ via sink node j, at cost $c_{i,j} - c_{\iota(j),j}$.

E.2 DualSort Algorithm

The DUALSORT algorithm is given sequentially below in Algorithm 2. Without loss of generality, we assume that x_i is rearranged in increasing order, so that the sorting permutation σ is the identity. Let diag denote the operator used to extract the diagonal of a matrix, so that $\text{diag}(\mathbf{C}) \in \mathbb{R}^n$ and one has $[\text{diag}(\mathbf{C})]_i = c_{i,i}$, and write $\mathbf{1}$ for the vector of size n with all entries 1. The inner loop can be carried out in two different ways, either using a vectorized update or looping through coordinates one at a time. These two updates are distinct, and we do observe that cycling through coordinates in Gauss-Seidel fashion converges faster in terms of total number of updates. However, that perspective misses the fact that vectorized updates utilize more efficiently accelerators from a runtime perspective. Additionally, these updates are equal to, in terms of complexity to the Sinkhorn iterations, making it easier to discuss the benefits of our initializers. For these reasons, we use the `vectorized=True` flag in our experiments.

Algorithm 5: DUALSORT Initializer

Input: Cost matrix $\mathbf{C}$, primal solution, $\mathbf{P}$, `vectorized` flag
Initialize: $\mathbf{f} = 0$
while not converged **do**
  if `vectorized` **then**
    $\mathbf{f} \leftarrow \min_{\text{axis=1}} \left(\mathbf{C} - \text{diag}(\mathbf{C})\mathbf{1}^T + \mathbf{f}\mathbf{1}^T\right)$
  else
    for $i \in \{1, \ldots, n\}$ **do**
      $\mathbf{f}_i \leftarrow \left(\min_j c_{i,j} - c_{j,j} + \mathbf{f}_j\right)$
    end for
  end if
end while
Return $\mathbf{f}$

E.3 Number of DualSort Iterations

Figure 9 illustrates the convergence of the DualSort algorithm when compared to the true potentials found from linear programming. Visually, from the right plot of Figure 9, the approximate dual is close to the true dual after just one iteration. However the squared error (left plot) is still large. After 3 iterations, the error is significantly reduced and after 10, the error is not noticeable.

Figure 10 shows how the performance of the initializer improves significantly from 1 initialization iteration to 3 or 10 for the CIFAR-100 soft-error classification task. Here performance is measured in how many additional Sinkhorn iterations are required after initialization for convergence. Note however that empirically there is not much difference between 3 and 10,

Figure 9: Single sample of size 32 from Gaussian blob dataset with 5 centers. Left: squared error vs true potential by number of DualSort iterations. Right: Potential from linear solver vs DualSort approximations.

hence 3 was used in experiments.

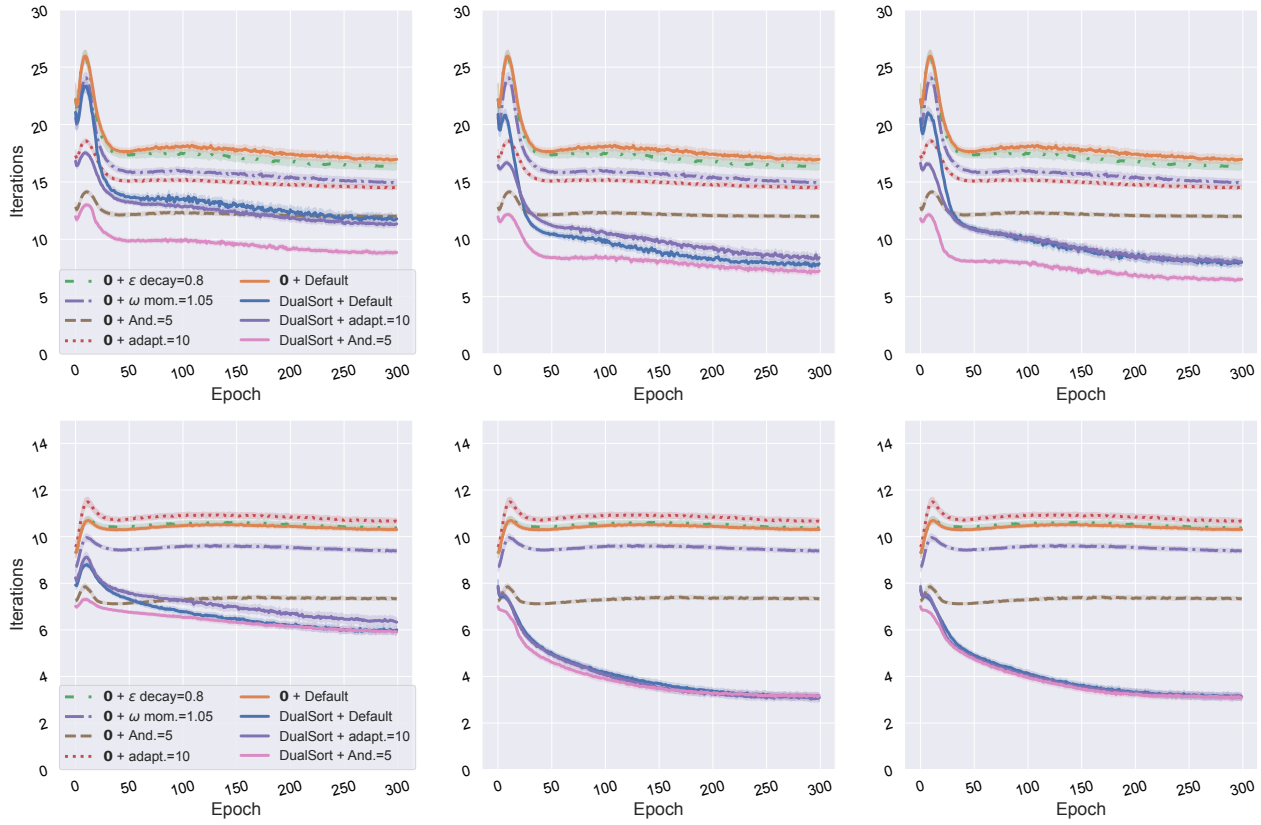


Figure 10: Number of Sinkhorn iterations per training step when using soft error loss for CIFAR-100 classifier. Top: threshold=0.01, bottom: threshold=0.05. Number of vectorized DualSort iterations 1,3,10 (left to right)

F Threshold Analysis

Convergence of each the Sinkhorn for each problem was determined according to a threshold tolerance, τ, for how close the marginals from the coupling derived from potentials are to the true marginals. For OT problem between $\mu = \sum_{i=1}^{n} a_i \delta_{x_i}$ and $\nu = \sum_{j=1}^{n} b_i \delta_{y_j}$, and denote potentials after l Sinkhorn iterations as $\mathbf{f}^{(l)}$, $\mathbf{g}^{(l)}$, then the corresponding coupling may be

written elementwise as $\mathbf{p}_{i,j}^{(l)} = \exp \frac{\mathbf{f}_i^{(l)} + \mathbf{g}_j^{(l)} - c_{i,j}}{\epsilon}$ and the threshold condition may be written

$$\sum_i |\sum_j \mathbf{p}_{i,j}^{(l)} - a_i| + \sum_j |\sum_i \mathbf{p}_{i,j}^{(l)} - b_j| < \tau.$$

We use $\tau = 0.01$ for speed. But also note that a higher threshold $\tau = 0.05$ leads to faster convergence without drop in performance, as evidenced in Figure 11 for the soft error classification task on CIFAR-100. Figure 10 also illustrates that the DualSort initializer appears to exhibit relatively better performance to the zero initialization for a higher convergence threshold.

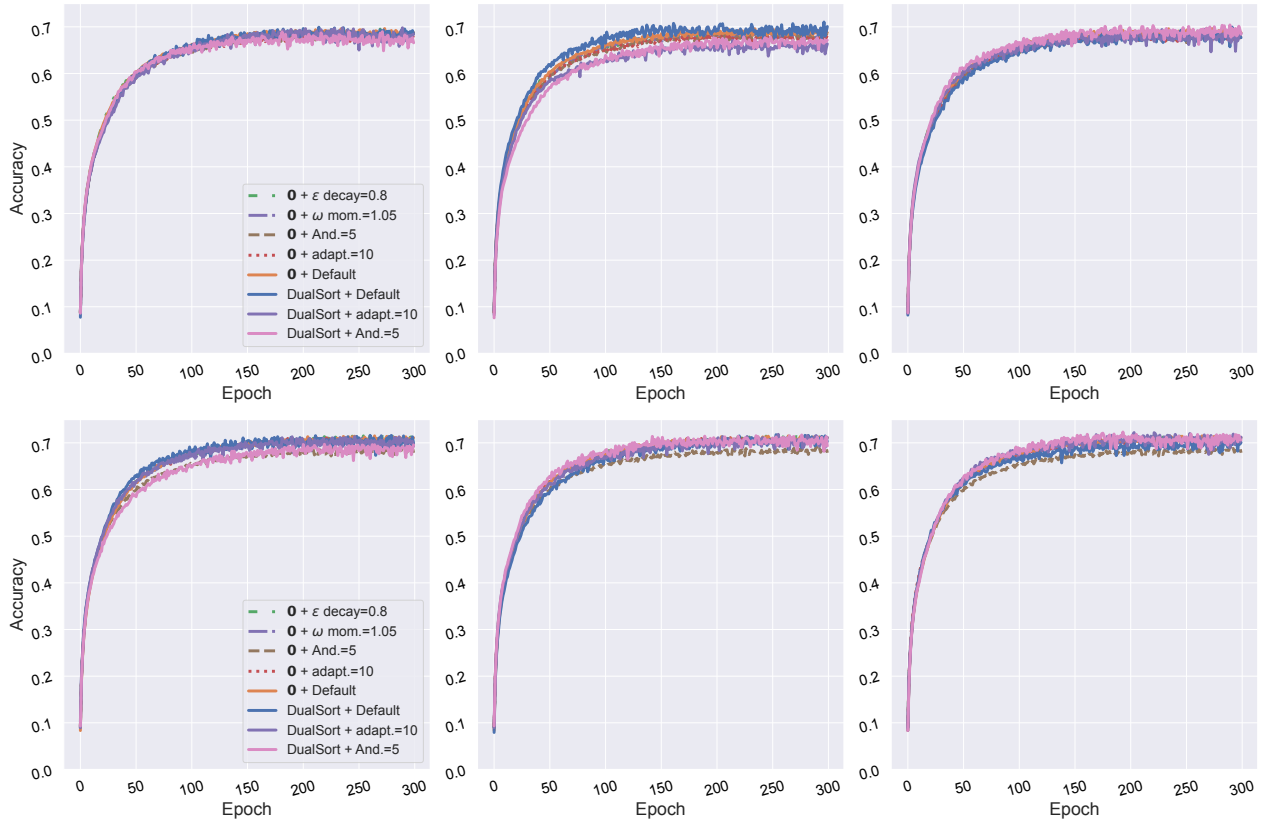


Figure 11: Evaluation accuracy through training when using soft error loss for CIFAR-100 classifier. Top: threshold=0.01, bottom: threshold=0.05. Number of vectorized DualSort iterations 1,3,10 (left to right)

G Other Details

Societal Impact. We are not aware of any direct negative societal impacts in this work. We acknowledge that the Sinkhorn algorithm may be used in various applications across compute vision and tracking with negative impacts, and this work may enable further such applications.
Code. Code for initializers has been incorporated into OTT library (Cuturi et al., 2022).
Open source software and licences. (Cuturi et al., 2022) has an Apache licence.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Rethinking Initialization of the Sinkhorn Algorithm
Publication Status	□ Accepted for Publication
Publication Details	James Thornton, Marco Cuturi, published at AISTATS 2023

Student Confirmation

Student Name:	James Thornton		

Contribution to the Paper	- I identified the gap in the literature and motivation.
	- I devised primary methodology and identified the flexibility in choosing initialization, which was then built upon by Marco:
	  - the initializer does not need to be differentiable due to implicit function theorem,
	  - the Sinkhorn algorithm will converge regardless of initialization,
	  - initializers (and Sinkhorn) may be embedded in larger neural networks,
	  - only need to initialize one of the two dual potentials (pointed out by Marco).
	- I introduced the Gaussian initializer, Gaussian mixture and subsample initializers.
	- Marco introduced the sorting initializer.
	- I wrote the code and carried out the experiments using Marco's OTT-JAX library, and guidance from Marco.
	- I wrote the initial draft of the paper, which was then refined with Marco.		

Signature 		Date	23/03/202

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Arnaud Doucet, Professor of Statistics		
Supervisor comments I agree with the statement of contributions.		
Signature 	Date	23/03/2023

This completed form should be included in the thesis, at the end of the relevant chapter.

Chapter 5

Diffusion Schrödinger Bridge with Applications to Score-Based Generative Modeling

Diffusion Schrödinger Bridge with Applications to Score-Based Generative Modeling

Valentin De Bortoli
Department of Statistics,
University of Oxford, UK

James Thornton
Department of Statistics,
University of Oxford, UK

Jeremy Heng
ESSEC Business School,
Singapore

Arnaud Doucet
Department of Statistics,
University of Oxford, UK

Abstract

Progressively applying Gaussian noise transforms complex data distributions to approximately Gaussian. Reversing this dynamic defines a generative model. When the forward noising process is given by a Stochastic Differential Equation (SDE), Song et al. (2021) demonstrate how the time inhomogeneous drift of the associated reverse-time SDE may be estimated using score-matching. A limitation of this approach is that the forward-time SDE must be run for a sufficiently long time for the final distribution to be approximately Gaussian while ensuring that the corresponding time-discretization error is controlled. In contrast, solving the Schrödinger Bridge (SB) problem, *i.e.* an entropy-regularized optimal transport problem on path spaces, yields diffusions which generate samples from the data distribution in finite time. We present Diffusion SB (DSB), an original approximation of the Iterative Proportional Fitting (IPF) procedure to solve the SB problem, and provide theoretical analysis along with generative modeling experiments. The first DSB iteration recovers the methodology proposed by Song et al. (2021), with the flexibility of using shorter time intervals, as subsequent DSB iterations reduce the discrepancy between the final-time marginal of the forward (resp. backward) SDE with respect to the Gaussian prior (resp. data) distribution. Beyond generative modeling, DSB offers a computational optimal transport tool as the continuous state-space analogue of the popular Sinkhorn algorithm (Cuturi, 2013).

1 Introduction

Score-Based Generative Modeling (SGM) is a recently developed approach to probabilistic generative modeling that exhibits state-of-the-art performance on several audio and image synthesis tasks; see *e.g.* Song and Ermon (2019); Cai et al. (2020); Chen et al. (2021a); Kong et al. (2021); Gao et al. (2020); Jolicoeur-Martineau et al. (2021b); Ho et al. (2020); Song and Ermon (2020); Song et al. (2020, 2021); Niu et al. (2020); Durkan and Song (2021); Hoogeboom et al. (2021); Saharia et al. (2021); Luhman and Luhman (2021, 2020); Nichol and Dhariwal (2021); Popov et al. (2021); Dhariwal and Nichol (2021). Existing SGMs generally consist of two parts. Firstly, noise is incrementally added to the data in order to obtain a perturbed data distribution approximating an easy-to-sample *prior* distribution *e.g.* Gaussian. Secondly, a neural network is used to learn the reverse-time denoising dynamics, which when initialized at this prior distribution, defines a generative model (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song and Ermon, 2019; Song et al., 2021). Song et al. (2021) have shown that one could fruitfully view the noising process as a Stochastic Differential Equation (SDE) that progressively perturbs the initial data distribution into an approximately Gaussian one.

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

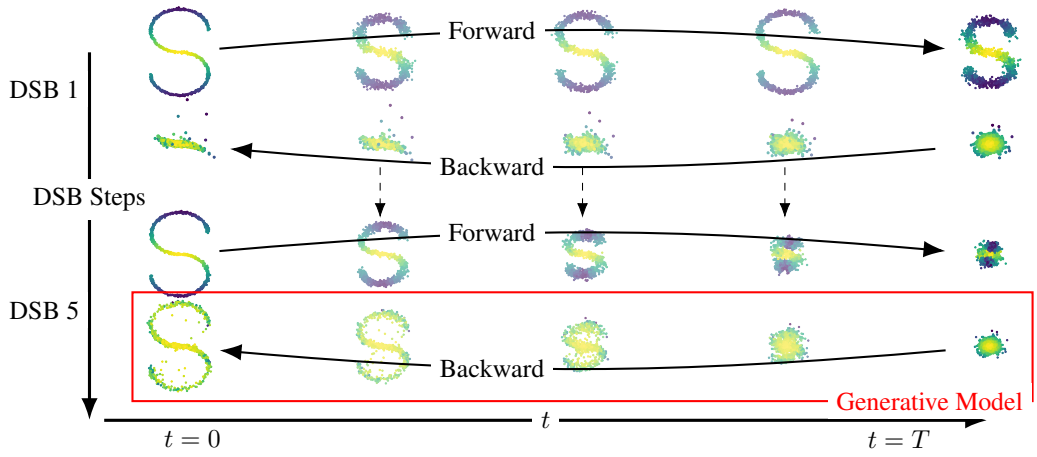


Figure 1: The reference forward diffusion initialized from the 2-dimensional data distribution fails to converge to the Gaussian prior in $T = 0.2$ diffusion-time ($N = 20$ discrete time steps), and the reverse diffusion initialized from the Gaussian prior does not converge to the data distribution. However, convergence does occur after 5 DSB iterations.

The corresponding reverse-time SDE is an inhomogeneous diffusion whose drift depends on the logarithmic gradients of the perturbed data distributions, *i.e.* the scores. In practice, these scores are approximated using neural networks and score-matching techniques (Hyvärinen and Dayan, 2005; Vincent, 2011) while numerical SDE integrators are used for the sampling procedure.

Although SGM provides state-of-the-art results (Dhariwal and Nichol, 2021), sample generation is computationally expensive. In order to learn the reverse-time SDE from the prior, *i.e.* the generative model, the forward noising SDE must be run for a sufficiently long time to converge to the prior and the step size must be sufficiently small to obtain a good numerical approximation of this SDE. By reformulating generative modeling as a Schrödinger bridge (SB) problem, we mitigate this issue and propose a novel algorithm to solve SB problems. Our detailed contributions are as follows.

Generative modeling as a Schrödinger bridge problem. The SB problem is a famous entropy-regularized Optimal Transport (OT) problem introduced by Schrödinger (1932); see *e.g.* (Léonard, 2014b; Chen et al., 2021b) for reviews. Given a reference diffusion with finite time horizon T, a data distribution and a prior distribution, solving the SB amounts to finding the closest diffusion to the reference (in terms of Kullback–Leibler divergence on path spaces) which admits the data distribution as marginal at time $t = 0$ and the prior at time $t = T$. The reverse-time diffusion solving this SB problem provides a new SGM algorithm which enables approximate sample generation from the data distribution using shorter time intervals compared to the original SGM methods. Our method differs from the entropy-regularized OT formulation in (Genevay et al., 2018), which deals with discrete distributions and relies on a static formulation of SB, as opposed to our dynamical approach for continuous distributions which operates on path spaces. It also differs from (Finlay et al., 2020) which approximates the SB solution by a diffusion whose drift is computed using potentials of the dual formulation of SB. Finally, Wang et al. (2021) have recently proposed to perform generative modeling by solving not one but two SB problems. Contrary to us, they do not formulate generative modeling as computing the SB between the data and prior distributions.

Solving the Schrödinger bridge problem using score-based diffusions. The SB problem can be solved using Iterative Proportional Fitting (IPF) (Fortet, 1940; Kullback, 1968; Chen et al., 2021b). We propose Diffusion SB (DSB), a novel implementation of IPF using score-based diffusion techniques. DSB does not require discretizing the state-space (Chen et al., 2016; Reich, 2019), approximating potential functions using regression (Bernton et al., 2019; Dessein et al., 2017; Pavon et al., 2021), nor performing kernel density estimation (Pavon et al., 2021). The first DSB iteration recovers the method proposed by Song et al. (2021), with the flexibility of using shorter time intervals, as additional DSB iterations reduce the discrepancy between the final-time marginal of the forward (resp. backward) SDE w.r.t. the prior (resp. data) distribution; see Figure 1 for an illustration. An algorithm akin to DSB has been proposed concurrently and independently by Vargas et al. (2021); the main difference

with our algorithm is that they estimate the drifts of the SDEs using Gaussian processes while we use neural networks and score matching ideas.

Theoretical results. We provide the first quantitative convergence results for the methodology of Song et al. (2021). In particular, we show that while we simulate Langevin-type diffusions in potentially extremely high-dimensional spaces, the SGM approach does *not* suffer from poor mixing times. Additionally, we derive novel quantitative convergence results for IPF in continuous state-space which do not rely on classical compactness assumptions (Chen et al., 2016; Ruschendorf et al., 1995) and improve on the recent results of Léger (2020). Finally, we show that DSB may be viewed as the time discretization of a dynamic version of IPF on path spaces based on forward/backward diffusions.

Experiments. We validate our methodology by generating image datasets such as MNIST and CelebA. In particular, we show that using multiple steps of DSB always improve the generative model. We also show how DSB can be used to interpolate between two data distributions.

Notation. In the continuous-time setting, we set $\mathcal{C} = \mathrm{C}([0,T],\mathbb{R}^d)$ the space of continuous functions from $[0,T]$ to $\mathbb{R}^d$ and $\mathcal{B}(\mathcal{C})$ the Borel sets on $\mathcal{C}$. For any measurable space $(\mathsf{E},\mathcal{E})$, we denote by $\mathscr{P}(\mathsf{E})$ the space of probability measures on $(\mathsf{E},\mathcal{E})$. For any $\ell \in \mathbb{N}$, let $\mathscr{P}_\ell = \mathscr{P}((\mathbb{R}^d)^\ell)$. When it is defined, we denote $\mathrm{H}(p) = -\int_{\mathbb{R}^d} p(x)\log p(x)\mathrm{d}x$ as the entropy of p and $\mathrm{KL}(p|q)$ as the Kullback–Leibler divergence between p and q. When there is no ambiguity, we use the same notation for distributions and their densities. All proofs are postponed to the supplementary.

2 Denoising Diffusion, Score-Matching and Reverse-Time SDEs

2.1 Discrete-Time: Markov Chains and Time Reversal

Consider a data distribution with positive density p_{data}[1], a positive prior density p_{prior} w.r.t. Lebesgue measure both with support on $\mathbb{R}^d$ and a Markov chain with initial density $p_0 = p_{\text{data}}$ on $\mathbb{R}^d$ evolving according to positive transition densities $p_{k+1|k}$ for $k \in \{0,\dots,N-1\}$. Hence for any $x_{0:N} = \{x_k\}_{k=0}^N \in \mathcal{X} = (\mathbb{R}^d)^{N+1}$, the joint density may be expressed as

$$p(x_{0:N}) = p_0(x_0)\prod_{k=0}^{N-1} p_{k+1|k}(x_{k+1}|x_k). \tag{1}$$

This joint density also admits the backward decomposition

$$p(x_{0:N}) = p_N(x_N)\prod_{k=0}^{N-1} p_{k|k+1}(x_k|x_{k+1}), \text{ with } p_{k|k+1}(x_k|x_{k+1}) = \frac{p_k(x_k)p_{k+1|k}(x_{k+1}|x_k)}{p_{k+1}(x_{k+1})}, \tag{2}$$

where $p_k(x_k) = \int p_{k|k-1}(x_k|x_{k-1})p_{k-1}(x_{k-1})\mathrm{d}x_{k-1}$ is the marginal density at step $k \geq 1$. For the purpose of generative modeling, we will choose transition densities such that $p_N(x_N) = \int p(x_{0:N})\mathrm{d}x_{0:N-1} \approx p_{\text{prior}}(x_N)$ for large N, where p_{prior} is an easy-to-sample *prior* density. One may sample approximately from p_{data} using ancestral sampling with the reverse-time decomposition (2), *i.e.* first sample $X_N \sim p_{\text{prior}}$ followed by $X_k \sim p_{k|k+1}(\cdot|X_{k+1})$ for $k \in \{N-1,\dots,0\}$. This idea is at the core of all recent SGM methods. The reverse-time transitions in (2) cannot be simulated exactly but may be approximated if we consider a forward transition density of the form

$$p_{k+1|k}(x_{k+1}|x_k) = \mathcal{N}(x_{k+1}; x_k + \gamma_{k+1}f(x_k), 2\gamma_{k+1}\mathbf{I}), \tag{3}$$

with drift $f : \mathbb{R}^d \to \mathbb{R}^d$ and stepsize $\gamma_{k+1} > 0$. We first make the following approximation from (2)

$$\begin{aligned} p_{k|k+1}(x_k|x_{k+1}) &= p_{k+1|k}(x_{k+1}|x_k)\exp[\log p_k(x_k) - \log p_{k+1}(x_{k+1})] \\ &\approx \mathcal{N}(x_k; x_{k+1} - \gamma_{k+1}f(x_{k+1}) + 2\gamma_{k+1}\nabla\log p_{k+1}(x_{k+1}), 2\gamma_{k+1}\mathbf{I}), \end{aligned} \tag{4}$$

using that $p_k \approx p_{k+1}$, a Taylor expansion of $\log p_{k+1}$ at x_{k+1} and $f(x_k) \approx f(x_{k+1})$. In practice, the approximation holds if $\|x_{k+1} - x_k\|$ is small which is ensured by choosing γ_{k+1} small enough. Although $\nabla\log p_{k+1}$ is not available, one may obtain an approximation using denoising score-matching methods (Hyvärinen and Dayan, 2005; Vincent, 2011; Song et al., 2021).

Assume that the conditional density $p_{k+1|0}(x_{k+1}|x_0)$ is available analytically as in (Ho et al., 2020; Song et al., 2021). We have $p_{k+1}(x_{k+1}) = \int p_0(x_0)p_{k+1|0}(x_{k+1}|x_0)\mathrm{d}x_0$ and elementary calculations show that $\nabla\log p_{k+1}(x_{k+1}) = \mathbb{E}_{p_{0|k+1}}[\nabla_{x_{k+1}}\log p_{k+1|0}(x_{k+1}|X_0)]$. We can therefore

[1] In this presentation, we assume that all distributions admit a density w.r.t. the Lebesgue measure for simplicity. However, the algorithms presented here only require having access to samples from p_{data} and p_{prior}.

formulate score estimation as a regression problem and use a flexible class of functions, *e.g.* neural networks, to parametrize an approximation $s_{\theta^\star}(k, x_k) \approx \nabla \log p_k(x_k)$ such that

$$\theta^\star = \arg\min_\theta \sum_{k=1}^N \mathbb{E}_{p_{0,k}}[||s_\theta(k, X_k) - \nabla_{x_k} \log p_{k|0}(X_k|X_0)||^2],$$

where $p_{0,k}(x_0, x_k) = p_0(x_0)p_{k|0}(x_k|x_0)$ is the joint density at steps 0 and k. If $p_{k|0}$ is not available, we use $\theta^\star = \arg\min_\theta \sum_{k=1}^N \mathbb{E}_{p_{k-1,k}}[||s_\theta(k, X_k) - \nabla_{x_k} \log p_{k|k-1}(X_k|X_{k-1})||^2]$. In summary, SGM involves first estimating the score function $s_{\theta^\star}$ from noisy data, and then sampling X_0 using $X_N \sim p_{\text{prior}}$ and the approximation (4), *i.e.*

$$X_k = X_{k+1} - \gamma_{k+1} f(X_{k+1}) + 2\gamma_{k+1} s_{\theta^\star}(k+1, X_{k+1}) + \sqrt{2\gamma_{k+1}} Z_{k+1}, Z_{k+1} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \mathbf{I}). \quad (5)$$

The random variable X_0 is approximately $p_0 = p_{\text{data}}$ distributed if $p_N(x_N) \approx p_{\text{prior}}(x_N)$. In what follows, we let $\{Y_k\}_{k=0}^N = \{X_{N-k}\}_{k=0}^N$ and remark that $\{Y_k\}_{k=0}^N$ satisfies a forward recursion.

2.2 Continuous-Time: SDEs, Reverse-Time SDEs and Theoretical results

For appropriate transition densities, Song et al. (2021) showed that the forward and reverse-time Markov chains may be viewed as discretized diffusions. We derive the continuous-time limit of the procedure presented in Section 2.1 and establish convergence results. The Markov chain with kernel (3) corresponds to an Euler–Maruyama discretization of $(\mathbf{X}_t)_{t\in[0,T]}$, solving the following SDE

$$\mathrm{d}\mathbf{X}_t = f(\mathbf{X}_t)\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t, \quad \mathbf{X}_0 \sim p_0 = p_{\text{data}}, \quad (6)$$

where $(\mathbf{B}_t)_{t\in[0,T]}$ is a Brownian motion and $f : \mathbb{R}^d \to \mathbb{R}^d$ is regular enough so that (strong) solutions exist. Under conditions on f, it is well-known (see Haussmann and Pardoux (1986); Föllmer (1985); Cattiaux et al. (2021) for instance) that the reverse-time process $(\mathbf{Y}_t)_{t\in[0,T]} = (\mathbf{X}_{T-t})_{t\in[0,T]}$ satisfies

$$\mathrm{d}\mathbf{Y}_t = \{-f(\mathbf{Y}_t) + 2\nabla \log p_{T-t}(\mathbf{Y}_t)\}\,\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t, \quad (7)$$

with initialization $\mathbf{Y}_0 \sim p_T$, where p_t denotes the marginal density of $\mathbf{X}_t$.

The reverse-time Markov chain $\{Y_k\}_{k=0}^N$ associated with (5) corresponds to an Euler–Maruyama discretization of (7), where the score functions $\nabla \log p_t(x)$ are approximated by $s_{\theta^\star}(t, x)$.

In what follows, we consider $f(x) = -\alpha x$ for $\alpha \geq 0$. This framework includes the one of Song and Ermon (2019) ($\alpha = 0$, $p_{\text{prior}}(x) = \mathcal{N}(x; 0, 2T\,\mathbf{I})$) for which $(\mathbf{X}_t)_{t\in[0,T]}$ is simply a Brownian motion and Ho et al. (2020) ($\alpha > 0$, $p_{\text{prior}}(x) = \mathcal{N}(x; 0, \mathbf{I}/\alpha)$) for which it is an Ornstein–Uhlenbeck process, see Appendix C.3 for more details. Contrary to Song et al. (2021) we consider time homogeneous diffusions. Both approaches approximate (5) using distinct discretizations but our setting leverages the ergodic properties of the Ornstein–Uhlenbeck process to establish Theorem 1.

Theorem 1. *Assume that there exists* $\mathtt{M} \geq 0$ *such that for any* $t \in [0, T]$ *and* $x \in \mathbb{R}^d$

$$\|s_{\theta^\star}(t, x) - \nabla \log p_t(x)\| \leq \mathtt{M}, \quad (8)$$

with $s_{\theta^\star} \in \mathrm{C}([0, T] \times \mathbb{R}^d, \mathbb{R}^d)$. *Assume that* $p_{\text{data}} \in \mathrm{C}^3(\mathbb{R}^d, (0, +\infty))$ *is bounded and that there exist* $d_1, A_1, A_2, A_3 \geq 0$, $\beta_1, \beta_2, \beta_3 \in \mathbb{N}$ *and* $\mathtt{m}_1 > 0$ *such that for any* $x \in \mathbb{R}^d$ *and* $i \in \{1, 2, 3\}$

$$\|\nabla^i \log p_{\text{data}}(x)\| \leq A_i(1 + \|x\|^{\beta_i}), \quad \langle \nabla \log p_{\text{data}}(x), x\rangle \leq -\mathtt{m}_1 \|x\|^2 + d_1 \|x\|,$$

with $\beta_1 = 1$. *Then for any* $\alpha \geq 0$, *there exist* $B_\alpha, C_\alpha, D_\alpha \geq 0$ *such that for any* $N \in \mathbb{N}$ *and* $\{\gamma_k\}_{k=1}^N$ *with* $\gamma_k > 0$ *for any* $k \in \{1, \ldots, N\}$, *the following bounds on the total variation distance hold:*

(a) if $\alpha > 0$, *we have* $\|\mathcal{L}(X_0) - p_{\text{data}}\|_{\text{TV}} \leq C_\alpha(\mathtt{M} + \bar{\gamma}^{1/2}) \exp[D_\alpha T] + B_\alpha \exp[-\alpha^{1/2} T]$;

(b) if $\alpha = 0$, *we have* $\|\mathcal{L}(X_0) - p_{\text{data}}\|_{\text{TV}} \leq C_0(\mathtt{M} + \bar{\gamma}^{1/2}) \exp[D_0 T] + B_0(T^{-1} + T^{-1/2})$;

where $T = \sum_{k=1}^N \gamma_k$, $\bar{\gamma} = \sup_{k\in\{1,\ldots,N\}} \gamma_k$ *and* $\mathcal{L}(X_0)$ *is the distribution of* X_0 *given in* (5).

Proof. We provide here a sketch of the proof. The whole proof is detailed in Appendix C.2. Denote $\mathbb{P} \in \mathscr{P}(\mathcal{C})$ the path measure associated with (6) and $\mathbb{P}^R$ its time-reversal. Denote Q_N the Markov kernel taking us from Y_0 to Y_N induced by (5). We have

$$\|p_{\text{prior}}\mathrm{Q}_N - p_{\text{data}}\|_{\text{TV}} = \|p_{\text{prior}}\mathrm{Q}_N - p_{\text{data}}\mathbb{P}_{T|0}(\mathbb{P}^R)_{T|0}\|_{\text{TV}}$$

$$\leq \|p_{\text{prior}}\mathrm{Q}_N - p_{\text{prior}}(\mathbb{P}^R)_{T|0}\|_{\mathrm{TV}} + \|p_{\text{prior}}(\mathbb{P}^R)_{T|0} - p_{\text{data}}\mathbb{P}_{T|0}(\mathbb{P}^R)_{T|0}\|_{\mathrm{TV}}$$
$$\leq \|p_{\text{prior}}\mathrm{Q}_N - p_{\text{prior}}(\mathbb{P}^R)_{T|0}\|_{\mathrm{TV}} + \|p_{\text{prior}} - p_T\|_{\mathrm{TV}}.$$

We control the first term by bounding the discretization error of Q_N when compared to $(\mathbb{P}^R)_{T|0}$ via the Girsanov theorem. The second term is controlled using the mixing properties of the forward diffusion process. □

Condition (8) ensures that the neural network approximates the score with a given precision $\mathtt{M} \geq 0$. Under (8) and conditions on p_{data}, Theorem 1 states how the Markov chain defined by (5) approximates p_{data} in the total variation norm $\|\cdot\|_{\mathrm{TV}}$. The bounds of Theorem 1 show that there is a trade-off between the mixing properties of the forward diffusion which increases with α, and the quality of the discrete-time approximation which deteriorates as α and T increase, since $B_\alpha, C_\alpha D_\alpha \to_{\alpha\to+\infty} +\infty$. Indeed increasing α makes the drift steeper and the continuous-time process converges faster but smaller step sizes are required in order to control the error between the discrete and the continuous-time processes. Theorem 1 is the first theoretical result assessing the convergence of SGM methods. Indeed while Block et al. (2020) establish convergence results for a *time-homogeneous* Langevin diffusion targeting a density whose score is approximated by a neural network, all SGM methods used in practice rely on *time-inhomogeneous* processes. Contrary to the time-homogeneous case, this approach does not suffer from poor mixing times as the mixing time dependency in the bounds of Theorem 1 is entirely determined by the mixing time of the *forward* process, given by a simple Brownian motion or an Ornstein–Ulhenbeck process, and is independent of the dimension. Finally, note that (8) is a strong assumption. In practice we expect to obtain such bounds in expectation over X with high probability w.r.t. the data distribution as in (Block et al., 2020, Proposition 9). Our results are also related to (Tzen and Raginsky, 2019, Theorem 3.1) which establishes the expressiveness of related generative models using tools from stochastic control.

3 Diffusion Schrödinger Bridge and Generative Modeling

3.1 Schrödinger Bridges

The SB problem is a classical problem appearing in applied mathematics, optimal control and probability; see *e.g.* Föllmer (1988); Léonard (2014b); Chen et al. (2021b). In the discrete-time setting, it takes the following (dynamic) form. Consider as *reference* density $p(x_{0:N})$ given by (1), describing the process adding noise to the data. We aim to find $\pi^\star \in \mathscr{P}_{N+1}$ such that

$$\pi^\star = \arg\min \left\{ \mathrm{KL}(\pi|p) \,:\, \pi \in \mathscr{P}_{N+1},\ \pi_0 = p_{\text{data}},\ \pi_N = p_{\text{prior}} \right\}. \tag{9}$$

Assuming $\pi^\star$ is available, a generative model can be obtained by sampling $X_N \sim p_{\text{prior}}$, followed by the reverse-time dynamics $X_k \sim \pi^\star_{k|k+1}(\cdot|X_{k+1})$ for $k \in \{N-1, \dots, 0\}$. Before deriving a method to approximate $\pi^\star$ in Section 3.2, we highlight some desirable features of Schrödinger bridges.

Static Schrödinger bridge problem. First, we recall that the dynamic formulation (9) admits a static analogue. Using *e.g.* Léonard (2014a, Theorem 2.4), the following decomposition holds for any $\pi \in \mathscr{P}_{N+1}$, $\mathrm{KL}(\pi|p) = \mathrm{KL}(\pi_{0,N}|p_{0,N}) + \mathbb{E}_{\pi_{0,N}}[\mathrm{KL}(\pi_{|0,N}|p_{|0,N})]$, where for any $\mu \in \mathscr{P}_{N+1}$ we have $\mu = \mu_{0,N}\mu_{|0,N}$ with $\mu_{|0,N}$ the conditional distribution of $X_{1:N-1}$ given X_0, X_N[2]. Hence we have $\pi^\star(x_{0:N}) = \pi^{\mathrm{s},\star}(x_0, x_N)p_{|0,N}(x_{1:N-1}|x_0, x_N)$ where $\pi^{\mathrm{s},\star} \in \mathscr{P}_2$ with marginals $\pi_0^{\mathrm{s},\star}$ and $\pi_N^{\mathrm{s},\star}$ is the solution of the static SB problem

$$\pi^{\mathrm{s},\star} = \arg\min \left\{ \mathrm{KL}(\pi^{\mathrm{s}}|p_{0,N}) \,:\, \pi^{\mathrm{s}} \in \mathscr{P}_2,\ \pi_0^{\mathrm{s}} = p_{\text{data}},\ \pi_N^{\mathrm{s}} = p_{\text{prior}} \right\}. \tag{10}$$

Link with optimal transport. Under mild assumptions, the static SB problem can be seen as an entropy-regularized optimal transport problem since (10) is equivalent to

$$\pi^{\mathrm{s},\star} = \arg\min \left\{ -\mathbb{E}_{\pi^{\mathrm{s}}}[\log p_{N|0}(X_N|X_0)] - \mathrm{H}(\pi^{\mathrm{s}}) \,:\, \pi^{\mathrm{s}} \in \mathscr{P}_2,\ \pi_0^{\mathrm{s}} = p_{\text{data}},\ \pi_N^{\mathrm{s}} = p_{\text{prior}} \right\}.$$

If $p_{k+1|k}(x_{k+1}|x_k) = \mathcal{N}(x_{k+1}; x_k, \sigma_{k+1}^2)$ as in Song and Ermon (2019), then $p_{N|0}(x_N|x_0) = \mathcal{N}(x_N; x_0, \sigma^2)$ with $\sigma^2 = \sum_{k=1}^N \sigma_k^2$ which induces a quadratic cost and

$$\pi^{\mathrm{s},\star} = \arg\min \left\{ \mathbb{E}_{\pi^{\mathrm{s}}}[||X_0 - X_N||^2] - 2\sigma^2 \mathrm{H}(\pi^{\mathrm{s}}) \,:\, \pi^{\mathrm{s}} \in \mathscr{P}_2,\ \pi_0^{\mathrm{s}} = p_{\text{data}},\ \pi_N^{\mathrm{s}} = p_{\text{prior}} \right\}.$$

[2] See Appendix D.1 for a rigorous presentation using the disintegration theorem for probability measures.

Mikami (2004) showed that $\pi^{\mathrm{s},\star} \to \pi^\star_{\mathcal{W}}$ weakly and $2\sigma^2\,\mathrm{KL}(\pi^{\mathrm{s},\star}|p_{0,N}) \to \mathcal{W}_2^2(p_{\text{data}}, p_{\text{prior}})$ as $\sigma \to 0$, where $\pi^\star_{\mathcal{W}}$ is the optimal transport plan between p_{data} and p_{prior} and $\mathcal{W}_2$ is the 2-Wasserstein distance. Note that the transport cost $c(x, x') = -\log p_{N|0}(x'|x)$ is not necessarily symmetric.

3.2 Iterative Proportional Fitting and Time Reversal

In all but trivial cases, the SB problem does not admit a closed-form solution. However, it can be solved using Iterative Proportional Fitting (IPF) (Fortet, 1940; Kullback, 1968; Ruschendorf et al., 1995) which is defined by the following recursion for $n \in \mathbb{N}$ with initialization $\pi^0 = p$ given in (1):

$$\pi^{2n+1} = \arg\min \left\{ \mathrm{KL}(\pi|\pi^{2n}) \,:\, \pi \in \mathscr{P}_{N+1},\ \pi_N = p_{\text{prior}} \right\}, \tag{11}$$
$$\pi^{2n+2} = \arg\min \left\{ \mathrm{KL}(\pi|\pi^{2n+1}) \,:\, \pi \in \mathscr{P}_{N+1},\ \pi_0 = p_{\text{data}} \right\}.$$

This sequence is well-defined if there exists $\tilde{\pi} \in \mathscr{P}_{N+1}$ such that $\tilde{\pi}_0 = p_{\text{data}}$, $\tilde{\pi}_N = p_{\text{prior}}$ and $\mathrm{KL}(\tilde{\pi}|p) < +\infty$. A standard representation of π^n is obtained by updating the joint density p using potential functions, see Appendix D.2 for details. However, this representation of the IPF iterates is difficult to approximate as it requires approximating the potentials. Our methodology builds upon an alternative representation that is better suited to numerical approximations for generative modeling where one has access to samples of p_{data} and p_{prior}.

Proposition 2. *Assume that* $\mathrm{KL}(p_{\text{data}} \otimes p_{\text{prior}}|p_{0,N}) < +\infty$. *Then for any* $n \in \mathbb{N}$, π^{2n} *and* π^{2n+1} *admit positive densities w.r.t. the Lebesgue measure denoted as* p^n *resp.* q^n *and for any* $x_{0:N} \in \mathcal{X}$, *we have* $p^0(x_{0:N}) = p(x_{0:N})$ *and*

$$q^n(x_{0:N}) = p_{\text{prior}}(x_N) \textstyle\prod_{k=0}^{N-1} p^n_{k|k+1}(x_k|x_{k+1}),\ p^{n+1}(x_{0:N}) = p_{\text{data}}(x_0) \prod_{k=0}^{N-1} q^n_{k+1|k}(x_{k+1}|x_k).$$

In practice we have access to $p^n_{k+1|k}$ and $q^n_{k|k+1}$. Hence, to compute $p^n_{k|k+1}$ and $q^n_{k+1|k}$ we use

$$p^n_{k|k+1}(x_k|x_{k+1}) = \frac{p^n_{k+1|k}(x_{k+1}|x_k) p^n_k(x_k)}{p^n_{k+1}(x_{k+1})},\ q^n_{k+1|k}(x_{k+1}|x_k) = \frac{q^n_{k|k+1}(x_k|x_{k+1}) q^n_{k+1}(x_{k+1})}{q^n_k(x_k)}.$$

To the best of our knowledge, this representation of the IPF iterates has surprisingly neither been presented nor explored in the literature. One may interpret these formulas as follows. At iteration $2n$, we have $\pi^{2n} = p^n$ with $p^0 = p$ given by the noising process (1). This forward process initalized with $p^n_0 = p_{\text{data}}$ defines reverse-time transitions $p^n_{k|k+1}$, which, when combined with an initialization p_{prior} at step N defines the reverse-time process $\pi^{2n+1} = q^n$. The forward transitions $q^n_{k+1|k}$ associated to q^n are then used to obtain $\pi^{2n+2} = p^{n+1}$. IPF then iterates this procedure.

3.3 Diffusion Schrödinger Bridge as Iterative Mean-Matching Proportional Fitting

To approximate the IPF recursion defined in Proposition 2, we use similar approximations to Section 2.1. If at step $n \in \mathbb{N}$ we have $p^n_{k+1|k}(x_{k+1}|x_k) = \mathcal{N}(x_{k+1}; x_k + \gamma_{k+1} f^n_k(x_k), 2\gamma_{k+1}\mathbf{I})$ where $p^0 = p$ and $f^0_k = f$, then we can approximate the reverse-time transitions in Proposition 2 by

$$\begin{aligned} q^n_{k|k+1}(x_k|x_{k+1}) &= p^n_{k+1|k}(x_{k+1}|x_k) \exp[\log p^n_k(x_k) - \log p^n_{k+1}(x_{k+1})] \\ &\approx \mathcal{N}(x_k; x_{k+1} + \gamma_{k+1} b^n_{k+1}(x_{k+1}), 2\gamma_{k+1}\mathbf{I}), \end{aligned}$$

with $b^n_{k+1}(x_{k+1}) = -f^n_k(x_{k+1}) + 2\nabla \log p^n_{k+1}(x_{k+1})$. We can also approximate the forward transitions in Proposition 2 by $p^{n+1}_{k+1|k}(x_{k+1}|x_k) \approx \mathcal{N}(x_{k+1}; x_k + \gamma_{k+1} f^{n+1}_k(x_k), 2\gamma_{k+1}\mathbf{I})$ with $f^{n+1}_k(x_k) = -b^n_{k+1}(x_k) + 2\nabla \log q^n_k(x_k)$. Hence we have $f^{n+1}_k(x_k) = f^n_k(x_k) - 2\nabla \log p^n_{k+1}(x_k) + 2\nabla \log q^n_k(x_k)$. It follows that one could estimate f^{n+1}_k, b^{n+1}_k by using score-matching to approximate $\{\nabla \log p^i_{k+1}(x)\}_{i=0}^n$, $\{\nabla \log q^i_k(x)\}_{i=0}^n$. This approach is prohibitively costly in terms of memory and compute, see Appendix E. We follow an alternative approach which avoids these difficulties.

Proposition 3. *Assume that for any* $n \in \mathbb{N}$ *and* $k \in \{0, \dots, N-1\}$,

$$q^n_{k|k+1}(x_k|x_{k+1}) = \mathcal{N}(x_k; B^n_{k+1}(x_{k+1}), 2\gamma_{k+1}\mathbf{I}),\ p^n_{k+1|k}(x_{k+1}|x_k) = \mathcal{N}(x_{k+1}; F^n_k(x_k), 2\gamma_{k+1}\mathbf{I}),$$

with $B^n_{k+1}(x) = x + \gamma_{k+1} b^n_{k+1}(x)$, $F^n_k(x) = x + \gamma_{k+1} f^n_k(x)$ *for any* $x \in \mathbb{R}^d$. *Then we have for any* $n \in \mathbb{N}$ *and* $k \in \{0, \dots, N-1\}$

$$B^n_{k+1} = \arg\min_{\mathrm{B} \in \mathrm{L}^2(\mathbb{R}^d, \mathbb{R}^d)} \mathbb{E}_{p^n_{k,k+1}}[\|\mathrm{B}(X_{k+1}) - (X_{k+1} + F^n_k(X_k) - F^n_k(X_{k+1}))\|^2], \tag{12}$$
$$F^{n+1}_k = \arg\min_{\mathrm{F} \in \mathrm{L}^2(\mathbb{R}^d, \mathbb{R}^d)} \mathbb{E}_{q^n_{k,k+1}}[\|\mathrm{F}(X_k) - (X_k + B^n_{k+1}(X_{k+1}) - B^n_{k+1}(X_k))\|^2]. \tag{13}$$

Proposition 3 shows how one can recursively approximate B_{k+1}^n and F_k^{n+1}. In practice, we use neural networks $B_{\beta^n}(k,x) \approx B_k^n(x)$ and $F_{\alpha^n}(k,x) \approx F_k^n(x)$. Note that the networks could also be learned jointly. In this case, at equilibrium, we would obtain a bridge between p_{data} and p_{prior} but not necessarily the Schrödinger bridge.

Network parameters α^n, β^n are learnt through gradient descent to minimize empirical versions of the sum over k of the loss functions given by (12) and (13) computed using M samples and denoted as $\hat{\ell}_n^b(\beta)$ and $\hat{\ell}_{n+1}^f(\alpha)$. The resulting algorithm approximating $L \in \mathbb{N}$ IPF iterations is called Diffusion Schrödinger Bridge (DSB) and is summarized in Algorithm 1 with $Z_k^j, \tilde{Z}_k^j \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \mathbf{I})$, see Figure 1 for an illustration.

Algorithm 1 Diffusion Schrödinger Bridge

for $n \in \{0, \dots, L\}$ **do**
  while not converged **do**
    Sample $\{X_k^j\}_{k,j=0}^{N,M}$, where $X_0^j \sim p_{\text{data}}$, and $X_{k+1}^j = F_{\alpha^n}(k, X_k^j) + \sqrt{2\gamma_{k+1}} Z_{k+1}^j$
    Compute $\hat{\ell}_n^b(\beta^n)$ approximating (12)
    $\beta^n \leftarrow$ Gradient Step$(\hat{\ell}_n^b(\beta^n))$
  end while
  while not converged **do**
    Sample $\{X_k^j\}_{k,j=0}^{N,M}$, where $X_N^j \sim p_{\text{prior}}$, and $X_{k-1}^j = B_{\beta^n}(k, X_k^j) + \sqrt{2\gamma_k} \tilde{Z}_k^j$
    Compute $\hat{\ell}_{n+1}^f(\alpha^{n+1})$ approximating (13)
    $\alpha^{n+1} \leftarrow$ Gradient Step$(\hat{\ell}_{n+1}^f(\alpha^{n+1}))$
  end while
end for
Output: (α^{L+1}, β^L)

The DSB algorithm is initialized using the reference dynamics $f_{\alpha^0}(k,x) = f(x)$. Once β^L is learnt we can easily approximately sample from p_{data} by sampling $X_N \sim p_{\text{prior}}$ and then using $X_{k-1} = B_{\beta^L}(k, X_k) + \sqrt{2\gamma_k} Z_k$ with $Z_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \mathbf{I})$. The resulting samples X_0 will be approximately distributed from p_{data}. Although DSB requires learning a sequence of network parameters, α^n, β^n, fewer diffusion steps are needed compared to standard SGM. In addition, as detailed in Appendix I, β^0 may be trained efficiently in a similar manner to previous SGM methods. Subsequent $\alpha^{n+1}, \beta^{n+1}$ are refinements of α^n, β^n, hence may be fine-tuned from previous iterations.

3.4 Convergence of Iterative Proportional Fitting

In this section, we investigate the theoretical properties of IPF. When the state-space is discrete and finite (Franklin and Lorenz, 1989; Peyré and Cuturi, 2019) or in the case where p_{data} and p_{prior} are compactly supported (Chen et al., 2016), IPF converges at a geometric rate w.r.t. the Hilbert-Birkhoff metric, see Lemmens and Nussbaum (2014) for a definition. Other than recent work by Léger (2020), only qualitative results exist in the general case where p_{data} or p_{prior} is not compactly supported (Ruschendorf et al., 1995; Rüschendorf and Thomsen, 1993). We establish here quantitative convergence of IPF in this non-compact setting as well as novel monotonicity results. We require only the following mild assumption.

A1. $p_N, p_{\text{prior}} > 0$, $|\mathrm{H}(p_{\text{prior}})| < +\infty$, $\int_{\mathbb{R}^d} |\log p_{N|0}(x_N|x_0)| p_{\text{data}}(x_0) p_{\text{prior}}(x_N) \mathrm{d}x_0 \mathrm{d}x_N < +\infty$.

Assumption **A**1 is satisfied in all of our experimental settings. We recall that for $\mu, \nu \in \mathscr{P}(\mathsf{E})$ with $(\mathsf{E}, \mathcal{E})$ a measurable space, the Jeffrey's divergence is given by $\mathrm{J}(\mu, \nu) = \mathrm{KL}(\mu|\nu) + \mathrm{KL}(\nu|\mu)$.

Proposition 4. *Assume* **A**1. *Then* $(\pi^n)_{n\in\mathbb{N}}$ *is well-defined and for any* $n \geq 1$ *we have*

$$\mathrm{KL}(\pi^{n+1}|\pi^n) \leq \mathrm{KL}(\pi^{n-1}|\pi^n), \qquad \mathrm{KL}(\pi^n|\pi^{n+1}) \leq \mathrm{KL}(\pi^n|\pi^{n-1}).$$

In addition, $(\|\pi^{n+1} - \pi^n\|_{\mathrm{TV}})_{n\in\mathbb{N}}$ *and* $(\mathrm{J}(\pi^{n+1}, \pi^n))_{n\in\mathbb{N}}$ *are non-increasing. Finally, we have* $\lim_{n\to+\infty} n \{\mathrm{KL}(\pi_0^n|p_{\text{data}}) + \mathrm{KL}(\pi_N^n|p_{\text{prior}})\} = 0$.

A more general result with additional monotonicity properties is given in Appendix F. Under similar assumptions, Léger (2020, Corollary 1) established $\mathrm{KL}(\pi_0^n|p_0) \leq C/n$ with $C \geq 0$ using a Bregman divergence gradient descent perspective. In contrast, our proof relies only on tools from information geometry. In addition, we improve the convergence rate and show that $(\pi^n)_{n\in\mathbb{N}}$ converges in total variation towards π^∞, *i.e.* we not only obtain convergence of the marginals but also convergence of the joint distribution. Under restrictive conditions on p_{data} and p_{prior}, Ruschendorf et al. (1995) showed that π^∞ is the Schrödinger bridge. In the following proposition, we avoid this assumption using results on automorphisms of measures (Beurling, 1960).

Proposition 5. *Assume* A1. *Then there exists a solution* $\pi^\star \in \mathscr{P}_{N+1}$ *to the SB problem and we have* $\lim_{n\to+\infty} \|\pi^n - \pi^\infty\|_{\mathrm{TV}} = 0$ *with* $\pi^\infty \in \mathscr{P}_{N+1}$. *Let* $h = p_{0,N}/(p_0 \otimes p_N)$ *and assume that* $h \in \mathrm{C}((\mathbb{R}^d)^2, (0,+\infty))$ *and that there exist* $\Phi_0, \Phi_N \in \mathrm{C}(\mathbb{R}^d, (0,+\infty))$ *such that*

$$\int_{\mathbb{R}^d\times\mathbb{R}^d} (|\log h(x_0, x_N)| + |\log \Phi_0(x_0)| + |\log \Phi_N(x_N)|) p_{\text{data}}(x_0) p_{\text{prior}}(x_N) \mathrm{d}x_0 \mathrm{d}x_N < +\infty,$$

with $h(x_0, x_N) \leq \Phi_0(x_0)\Phi_N(x_N)$. *If* p *is absolutely continuous w.r.t.* π^∞ *then* $\pi^\infty = \pi^\star$.

Proposition 5 extends previous IPF convergence results without the assumption that the mapping h is lower bounded, see Ruschendorf et al. (1995); Chen et al. (2016). Our assumption on h can be relaxed and replaced by a tighter condition on π^∞, see Appendix F.2. Proposition 4 suggests a convergence rate of order $o(n)$ for the IPF in the non-compact setting. However, in some situations, we recover geometric convergence rates with explicit dependency w.r.t. the problem constants, see Appendix G. In practice, we do not run IPF for $p_{\text{data}}, p_{\text{prior}}$ but using empirical versions of these distributions. Recent results in Deligiannidis et al. (2021) show that the iterates of IPF based on empirical distributions remain close to the iterates one would obtain using the true distributions, uniformly in time. In particular, the SB computed using the empirical distributions converges to the one computed using the true distributions as the number of samples goes to infinity.

3.5 Continuous-time IPF

We describe an IPF algorithm for solving SB problems in continuous-time. We show that DSB proposed in Algorithm 1 can be seen as a discretization of this IPF. Given a reference measure $\mathbb{P} \in \mathscr{P}(\mathcal{C})$, the continuous formulation of the SB involves solving the following problem

$$\Pi^\star = \arg\min \left\{ \mathrm{KL}(\Pi|\mathbb{P}) \,:\, \Pi \in \mathscr{P}(\mathcal{C}),\ \Pi_0 = p_{\text{data}},\ \Pi_T = p_{\text{prior}} \right\}, \quad T = \textstyle\sum_{k=0}^{N-1} \gamma_{k+1}. \tag{14}$$

Similarly to (11), we define the IPF $(\Pi^n)_{n\in\mathbb{N}}$ with $\Pi^0 = \mathbb{P}$ associated with (6) and for any $n \in \mathbb{N}$

$$\Pi^{2n+1} = \arg\min \left\{ \mathrm{KL}(\Pi|\Pi^{2n}) \,:\, \Pi \in \mathscr{P}(\mathcal{C}),\ \Pi_T = p_{\text{prior}} \right\},$$
$$\Pi^{2n+2} = \arg\min \left\{ \mathrm{KL}(\Pi|\Pi^{2n+1}) \,:\, \Pi \in \mathscr{P}(\mathcal{C}),\ \Pi_0 = p_{\text{data}} \right\}.$$

One can show that for any $n \in \mathbb{N}$, $\Pi^n = \pi^{\mathrm{s},n}\mathbb{P}_{|0,T}$, with $(\pi^{\mathrm{s},n})_{n\in\mathbb{N}}$ the IPF for the static SB problem. In particular, Proposition 4 and Proposition 5 extend to the continuous IPF framework. In what follows, for any $\mathbb{P} \in \mathscr{P}(\mathcal{C})$, we define $\mathbb{P}^R$ as the reverse-time measure, *i.e.* for any $\mathsf{A} \in \mathcal{B}(\mathcal{C})$ we have $\mathbb{P}^R(\mathsf{A}) = \mathbb{P}(\mathsf{A}^R)$ where $\mathsf{A}^R = \{t \mapsto \omega(T-t) \,:\, \omega \in \mathsf{A}\}$. The following result is the continuous counterpart of Proposition 2 and states that each IPF iteration is associated with a diffusion, showing that DSB can be seen as a discretization of the continuous IPF.

Proposition 6. *Assume* A1 *and that there exist* $\mathbb{M} \in \mathscr{P}(\mathcal{C})$, $U \in \mathrm{C}^1(\mathbb{R}^d, \mathbb{R})$, $C \geq 0$ *such that for any* $n \in \mathbb{N}$, $x \in \mathbb{R}^d$, $\mathrm{KL}(\Pi^n|\mathbb{M}) < +\infty$, $\langle x, \nabla U(x)\rangle \geq -C(1 + \|x\|^2)$ *and* $\mathbb{M}$ *is associated with*

$$\mathrm{d}\mathbf{X}_t = -\nabla U(\mathbf{X}_t)\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t, \tag{15}$$

with $\mathbf{X}_0$ *distributed according to the invariant distribution of* (15). *Then, for any* $n \in \mathbb{N}$ *we have:*

(a) $(\Pi^{2n+1})^R$ *is associated with* $\mathrm{d}\mathbf{Y}_t^{2n+1} = b_{T-t}^n(\mathbf{Y}_t^{2n+1})\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t$ *with* $\mathbf{Y}_0^{2n+1} \sim p_{\text{prior}}$;

(b) Π^{2n+2} *is associated with* $\mathrm{d}\mathbf{X}_t^{2n+2} = f_t^{n+1}(\mathbf{X}_t^{2n+2})\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t$ *with* $\mathbf{X}_0^{2n+2} \sim p_{\text{data}}$;
where for any $n \in \mathbb{N}$, $t \in [0,T]$ *and* $x \in \mathbb{R}^d$, $b_t^n(x) = -f_t^n(x) + 2\nabla \log p_t^n(x)$, $f_t^{n+1}(x) = -b_t^n(x) + 2\nabla \log q_t^n(x)$, *with* $f_t^0(x) = f(x)$, *see* (6), *and* p_t^n, q_t^n *the densities of* Π_t^{2n} *and* Π_t^{2n+1}.

4 Experiments

Gaussian example. We first confirm that our algorithm recovers the true SB in a Gaussian setting where the ground truth is available. Let $p_{\text{prior}} = \mathcal{N}(-a, \mathbf{I})$, $p_{\text{data}} = \mathcal{N}(a, \mathbf{I})$ with $a \in \mathbb{R}^d$ and consider a Brownian motion as reference dynamics. The analytic expression for the static SB is $\mathcal{N}((-a, a), \Sigma)$ with $\Sigma \in \mathbb{R}^{2d\times 2d}$ given in Appendix G.2. We let $a = 0.1 \times \mathbf{1}$ with $d = 50$ or $d = 5$. In Figure 2, we illustrate the convergence of DSB. We train each DSB with a batch size of 128, $N = 20$ and $\gamma = 1/40$. We compare two network configurations: "small" where the network is given by Figure 9 (30k parameters) whereas "large" corresponds to the same network but with twice as many latent dimensions (240k parameters). The small network recovers the statistics of SB in the low-dimensional setting ($d = 5$) but is unable to recover the variance and covariance for $d = 50$. Increasing the size of the network solves this problem.

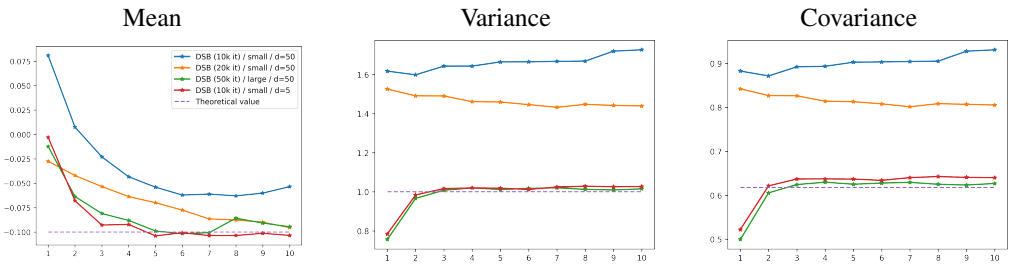


Figure 2: Convergence of DSB to ground-truth. From left to right: estimated mean, variance and covariance (first component) after each DSB iteration. The ground-truth value is given by the dashed line in each scenario.

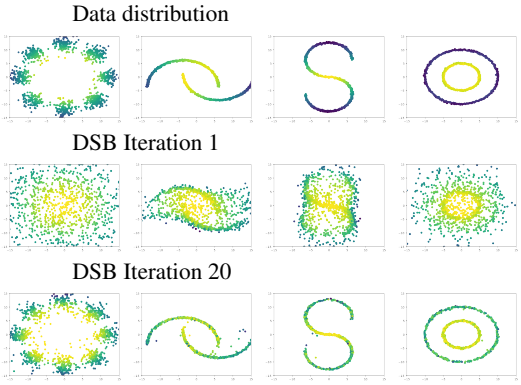
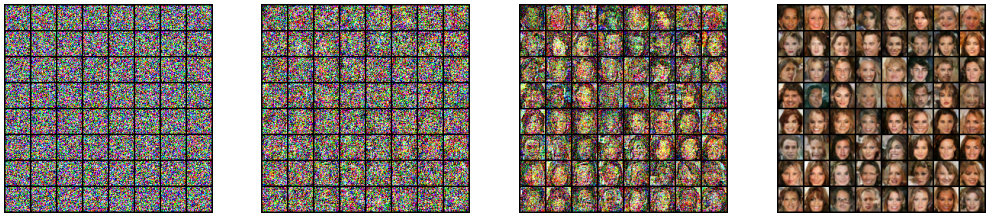


Figure 3: Data distributions p_{data} vs distribution at $t = 0$ for $T = 0.2$ after 1 and 20 DSB iterations.

Two dimensional toy experiments. We evaluate the validity of our approach on toy two dimensional examples. Contrary to existing SGM approaches we do *not* require that the number of steps is large enough for $p_N \approx p_{\text{prior}}$ to hold. We use a fully connected network with positional encoding (Vaswani et al., 2017) to approximate B_k^n and F_k^n, see Appendix J.1 and our code[3] for implementation details. Animated plots of the DSB iterations may be found online on our project webpage[4]. In Figure 3, we illustrate the benefits of DSB over classical SGM. We fix $f(x) = -\alpha x$ and choose $p_{\text{prior}} = \mathcal{N}(0, \sigma^2_{\text{data}}\mathbf{I})$, hence $\alpha = 1/\sigma^2_{\text{data}}$ where σ^2_{data} is the variance of the dataset. We let $N = 20$ and $\gamma_k = 0.01$, *i.e.* $T = 0.2$. Since T is small, we do not have $p_N \approx p_{\text{prior}}$ and the reverse-time process obtained after the first DSB iteration (corresponding to original SGM methods) does not yield a satisfactory generative model. However, multiple iterations of DSB improve the quality of the synthesis.

Generative modeling. DSB is the first practical algorithm for approximating the solution to the SB problem in high dimension ($d = 3072$ for CelebA). Whilst our implementation does not yet compete with state-of-the-art methods, we show promising results with fewer diffusion steps compared to initial SGMs (Song and Ermon, 2019) and demonstrate its performance on MNIST (LeCun and Cortes, 2010) and CelebA (Liu et al., 2015).



$t = 0$ $t = 0.31$ $t = 0.60$ $t = 0.63$

Figure 4: Generative model for CelebA 32×32 after 10 DSB iterations with $N = 50$ ($T = 0.63$)

A reduced U-net architecture based on Nichol and Dhariwal (2021) is used to approximate B_k^n and F_k^n. Further details are given in Appendix J.2. Our method is validated on downscaled CelebA in Figure 4. Figure 5 illustrates qualitative improvement over 8 DSB iterations with as few as $N = 12$ diffusion steps. Note, as shown in Appendix J.2, we obtain better results with higher N yet still significantly fewer steps than in the original SGM procedures (Song and Ermon, 2020, 2019) which

[3] Code is available here https://github.com/JTT94/diffusion_schrodinger_bridge
[4] https://vdeborto.github.io/publication/schrodinger_bridge/

use $N = 100$. Figure 6 illustrates how the sample quality, measured quantitatively in terms of Fréchet Inception Distance (FID) (Heusel et al., 2017), improves with the number of DSB iterations for various numbers of steps N.

DSB 1 DSB 8

Figure 5: Generated samples ($N = 12$)

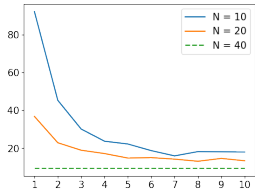


Figure 6: FID vs DSB Iterations.

Dataset interpolation. Schrödinger bridges not only reduce the number of steps in SGM methods but also enable flexibility in the choice of the prior density p_{prior}. Our approach is still valid for non-Gaussian p_{prior}, contrary to previous SGM works, and can be set as any other data distribution p'_{data}. In this case DSB converges towards a bridge between p_{data} and p'_{data}, see Figure 7. These experiments pave the way towards high-dimensional optimal transport between arbitrary data distributions.

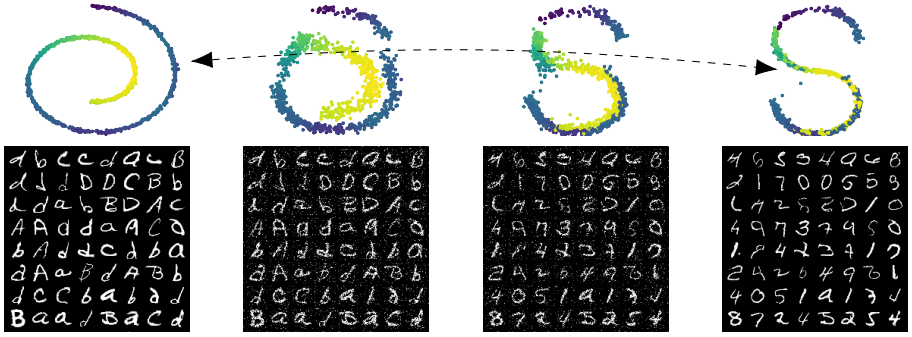


Figure 7: First row: Swiss-roll to S-curve (2D). Iteration 9 of DSB with $T = 1$ ($N = 50$). From left to right: $t = 0, 0.4, 0.6, 1$. Second row: EMNIST (Cohen et al., 2017) to MNIST. Iteration 10 of DSB with $T = 1.5$ ($N = 30$). From left to right: $t = 0, 0.4, 1.25, 1.5$.

5 Discussion

Score-based generative modeling (SGM) may be viewed as the first stage of solving a Schrödinger bridge problem. Building on this interpretation, we develop novel methodology, the Diffusion Schrödinger Bridge (DSB), that extends SGM approaches and allows one to perform generative modeling with fewer diffusion steps. DSB complements recent techniques to speed up existing SGM methods that rely on either different noise schedules (Nichol and Dhariwal, 2021; San-Roman et al., 2021; Watson et al., 2021), alternative discretizations (Jolicoeur-Martineau et al., 2021a) or knowledge distillation (Luhman and Luhman, 2021). Additionally, as the solution of the Schrödinger problem is a diffusion, it is possible as in Song et al. (2021, Section 4.3) to obtain an equivalent neural ordinary differential equation that admits the same marginals as the diffusion but enables exact likelihood computation, see Appendix H.3. Even though the final time $T > 0$ within DSB can be arbitrarily small, we observed that this has limits as choosing T too close to 0 decreases the quality of the generative models. One reason for this behavior is that if the endpoint of the original forward process is too far from the target distribution p_{prior}, then learning the score around the support of p_{prior} is challenging even for DSB. From a theoretical point of view, we have provided quantitative convergence results for SGM methods and derived new state-of-the-art convergence bounds for IPF as well as novel monotonicity results. We have demonstrated DSB on generative modeling and data interpolation tasks. Finally, although this work was motivated by generative modeling, DSB is much more widely applicable as it can be thought of as the continuous state-space counterpart of the celebrated Sinkhorn algorithm (Cuturi, 2013; Peyré and Cuturi, 2019). For example, DSB could be used to solve multi-marginal Schrödinger bridges problems (Di Marino and Gerolin, 2020), compute Wasserstein barycenters, find the minimizers of entropy-regularized Gromov–Wasserstein problems (Mémoli, 2011) or perform domain adaptation in continuous state-spaces.

Acknowledgments and Disclosure of Funding

Valentin De Bortoli and Arnaud Doucet are supported by the EPSRC CoSInES (COmputational Statistical INference for Engineering and Security) grant EP/R034710/1, James Thornton by the OxWaSP CDT through grant EP/L016710/1 and Jeremy Heng by the CY Initiative of Excellence (grant "Investissements d'Avenir" ANR-16-IDEX-0008). Computing resources were provided through the Google Cloud research credits programme. Arnaud Doucet also acknowledges support from the UK Defence Science and Technology Laboratory (DSTL) and EPSRC under grant EP/R013616/1. This is part of the collaboration between US DOD, UK MOD and UK EPSRC under the Multidisciplinary University Research Initiative.

References

Ambrosio, L., Gigli, N., and Savaré, G. (2008). *Gradient Flows in Metric Spaces and in the Space of Probability Measures*. Lectures in Mathematics ETH Zürich. Birkhäuser Verlag, Basel, second edition.

Bakry, D., Gentil, I., and Ledoux, M. (2014). *Analysis and Geometry of Markov Diffusion Operators*, volume 348. Springer.

Bernton, E., Heng, J., Doucet, A., and Jacob, P. E. (2019). Schrödinger bridge samplers. *arXiv preprint arXiv:1912.13170*.

Beurling, A. (1960). An automorphism of product measures. *Annals of Mathematics*, 72(1):189–200.

Block, A., Mroueh, Y., and Rakhlin, A. (2020). Generative modeling with denoising auto-encoders and Langevin sampling. *arXiv preprint arXiv:2002.00107*.

Cai, R., Yang, G., Averbuch-Elor, H., Hao, Z., Belongie, S., Snavely, N., and Hariharan, B. (2020). Learning gradient fields for shape generation. *European Conference on Computer Vision*.

Cattiaux, P., Conforti, G., Gentil, I., and Léonard, C. (2021). Time reversal of diffusion processes under a finite entropy condition. *arXiv preprint arXiv:2104.07708*.

Chen, N., Zhang, Y., Zen, H., Weiss, R. J., Norouzi, M., and Chan, W. (2021a). Wavegrad: Estimating gradients for waveform generation. *International Conference on Learning Representations*.

Chen, R. T., Rubanova, Y., Bettencourt, J., and Duvenaud, D. (2018). Neural ordinary differential equations. *arXiv preprint arXiv:1806.07366*.

Chen, Y., Georgiou, T., and Pavon, M. (2016). Entropic and displacement interpolation: a computational approach using the Hilbert metric. *SIAM Journal on Applied Mathematics*, 76(6):2375–2396.

Chen, Y., Georgiou, T. T., and Pavon, M. (2021b). Optimal transport in systems and control. *Annual Review of Control, Robotics, and Autonomous Systems*, 4.

Cohen, G., Afshar, S., Tapson, J., and van Schaik, A. (2017). EMNIST: an extension of MNIST to handwritten letters. *arXiv preprint arXiv:1702.05373*.

Constantine, G. M. and Savits, T. H. (1996). A multivariate Faà di Bruno formula with applications. *Transactions of the American Mathematical Society*, 348(2):503–520.

Csiszár, I. (1975). I-divergence geometry of probability distributions and minimization problems. *The Annals of Probability*, 3(1):146–158.

Cuturi, M. (2013). Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*.

Deligiannidis, G., De Bortoli, V., and Doucet, A. (2021). Quantitative uniform stability of the iterative proportional fitting procedure. *arXiv preprint arXiv:2108.08129*.

Dellacherie, C. and Meyer, P.-A. (1988). *Probabilities and Potential. C*, volume 151 of *North-Holland Mathematics Studies*. North-Holland Publishing Co., Amsterdam. Potential theory for discrete and continuous semigroups, Translated from the French by J. Norris.

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

Dessein, A., Papadakis, N., and Deledalle, C.-A. (2017). Parameter estimation in finite mixture models by regularized optimal transport: A unified framework for hard and soft clustering. *arXiv preprint arXiv:1711.04366*.

Dhariwal, P. and Nichol, A. (2021). Diffusion models beat GAN on image synthesis. *arXiv preprint arXiv:2105.05233*.

Di Marino, S. and Gerolin, A. (2020). An optimal transport approach for the Schrödinger bridge problem and convergence of Sinkhorn algorithm. *Journal of Scientific Computing*, 85(2):1–28.

Douc, R., Moulines, E., Priouret, P., and Soulier, P. (2019). *Markov Chains*. Springer.

Durkan, C. and Song, Y. (2021). On maximum likelihood training of score-based generative models. *arXiv preprint arXiv:2101.09258*.

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

Enderton, H. B. (1977). *Elements of Set Theory*. Academic Pres, New York-London.

Ethier, S. N. and Kurtz, T. G. (1986). *Markov Processes*. Wiley Series in Probability and Mathematical Statistics: Probability and Mathematical Statistics. John Wiley & Sons, Inc., New York. Characterization and convergence.

Finlay, C., Gerolin, A., Oberman, A. M., and Pooladian, A.-A. (2020). Learning normalizing flows from entropy-Kantorovich potentials. *arXiv preprint arXiv:2006.06033*.

Föllmer, H. (1985). An entropy approach to the time reversal of diffusion processes. In *Stochastic Differential Systems: Filtering and Control*, pages 156–163. Springer.

Föllmer, H. (1988). Random fields and diffusion processes. In *École d'Été de Probabilités de Saint-Flour XV–XVII, 1985–87*, pages 101–203. Springer.

Fortet, R. (1940). Résolution d'un système d'équations de M. Schrödinger. *Journal de Mathématiques Pures et Appliqués*, 1:83–105.

Franklin, J. and Lorenz, J. (1989). On the scaling of multidimensional matrices. *Linear Algebra and Its Applications*, 114:717–735.

Gao, R., Song, Y., Poole, B., Wu, Y. N., and Kingma, D. P. (2020). Learning energy-based models by diffusion recovery likelihood. *arXiv preprint arXiv:2012.08125*.

Genevay, A., Peyré, G., and Cuturi, M. (2018). Learning generative models with Sinkhorn divergences. In *International Conference on Artificial Intelligence and Statistics*.

Haussmann, U. G. and Pardoux, E. (1986). Time reversal of diffusions. *The Annals of Probability*, 14(4):1188–1205.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). GANs trained by a two time-scale update rule converge to a local Nash equilibrium. *arXiv preprint arXiv:1706.08500*.

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

Hoogeboom, E., Nielsen, D., Jaini, P., Forré, P., and Welling, M. (2021). Argmax flows and multinomial diffusion: Towards non-autoregressive language models. *arXiv preprint arXiv:2102.05379*.

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

Hyvärinen, A. and Dayan, P. (2005). Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4).

Ikeda, N. and Watanabe, S. (1989). *Stochastic Differential Equations and Diffusion Processes*, volume 24 of *North-Holland Mathematical Library*. North-Holland Publishing Co., Amsterdam; Kodansha, Ltd., Tokyo, second edition.

Jolicoeur-Martineau, A., Li, K., Piché-Taillefer, R., Kachman, T., and Mitliagkas, I. (2021a). Gotta go fast when generating data with score-based models. *arXiv preprint arXiv:2105.14080*.

Jolicoeur-Martineau, A., Piché-Taillefer, R., Tachet des Combes, R., and Mitliagkas, I. (2021b). Adversarial score matching and improved sampling for image generation. *International Conference on Learning Representations*.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kober, H. (1939). A theorem on Banach spaces. *Compositio Mathematica*, 7:135–140.

Kong, Z., Ping, W., Huang, J., Zhao, K., and Catanzaro, B. (2021). Diffwave: A versatile diffusion model for audio synthesis. *International Conference on Learning Representations*.

Kruithof, J. (1937). Telefoonverkeersrekening. *De Ingenieur*, 52:15–25.

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

Kullback, S. (1997). *Information Theory and Statistics*. Dover Publications, Inc., Mineola, NY. Reprint of the second (1968) edition.

Laumont, R., De Bortoli, V., Almansa, A., Delon, J., Durmus, A., and Pereyra, M. (2021). Bayesian imaging using plug & play priors: when Langevin meets Tweedie. *arXiv preprint arXiv:2103.04715*.

LeCun, Y. and Cortes, C. (2010). MNIST handwritten digit database.

Léger, F. (2020). A gradient descent perspective on Sinkhorn. *Applied Mathematics & Optimization*, pages 1–13.

Leha, G. and Ritter, G. (1984). On diffusion processes and their semigroups in Hilbert spaces with an application to interacting stochastic systems. *The Annals of Probability*, 12(4):1077–1112.

Lemmens, B. and Nussbaum, R. D. (2014). Birkhoff's version of Hilbert's metric and its applications in analysis. *Handbook of Hilbert Geometry*, pages 275–303.

Léonard, C. (2011). Stochastic derivatives and generalized h-transforms of markov processes. *arXiv preprint arXiv:1102.3172*.

Léonard, C. (2014a). Some properties of path measures. In *Séminaire de Probabilités XLVI*, pages 207–230. Springer.

Léonard, C. (2014b). A survey of the Schrödinger problem and some of its connections with optimal transport. *Discrete & Continuous Dynamical Systems-A*, 34(4):1533–1574.

Léonard, C. (2019). Revisiting Fortet's proof of existence of a solution to the Schrödinger system. *arXiv preprint arXiv:1904.13211*.

Liptser, R. S. and Shiryaev, A. N. (2001). *Statistics of Random Processes. I*, volume 5 of *Applications of Mathematics (New York)*. Springer-Verlag, Berlin, expanded edition. General theory, Translated from the 1974 Russian original by A. B. Aries, Stochastic Modelling and Applied Probability.

Liu, Z., Luo, P., Wang, X., and Tang, X. (2015). Deep learning face attributes in the wild. In *International Conference on Computer Vision*.

Luhman, E. and Luhman, T. (2021). Knowledge distillation in iterative generative models for improved sampling speed. *arXiv preprint arXiv:2101.02388*.

Luhman, T. and Luhman, E. (2020). Diffusion models for handwriting generation. *arXiv preprint arXiv:2011.06704*.

Mémoli, F. (2011). Gromov–Wasserstein distances and the metric approach to object matching. *Foundations of Computational Mathematics*, 11(4):417–487.

Meyn, S. P. and Tweedie, R. L. (1993). Stability of Markovian processes. III. Foster-Lyapunov criteria for continuous-time processes. *Advances in Applied Probability*, 25(3):518–548.

Mikami, T. (2004). Monge's problem with a quadratic cost by the zero-noise limit of h-path processes. *Probability Theory and Related Fields*, 129(2):245–260.

Nichol, A. and Dhariwal, P. (2021). Improved denoising diffusion probabilistic models. *arXiv preprint arXiv:2102.09672*.

Niu, C., Song, Y., Song, J., Zhao, S., Grover, A., and Ermon, S. (2020). Permutation invariant graph generation via score-based generative modeling. In *International Conference on Artificial Intelligence and Statistics*.

Pavon, M., Trigila, G., and Tabak, E. G. (2021). The data-driven Schrödinger bridge. *Communications on Pure and Applied Mathematics*, 74:1545–1573.

Petersen, K. B., Pedersen, M. S., et al. (2008). The matrix cookbook. *Technical University of Denmark*, 7(15):510.

Peyré, G. and Cuturi, M. (2019). Computational optimal transport. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607.

Popov, V., Vovk, I., Gogoryan, V., Sadekova, T., and Kudinov, M. (2021). Grad-tts: A diffusion probabilistic model for text-to-speech. *arXiv preprint arXiV:2105.06337*.

Reich, S. (2019). Data assimilation: the Schrödinger perspective. *Acta Numerica*, 28:635–711.

Revuz, D. and Yor, M. (1999). *Continuous Martingales and Brownian Motion*, volume 293 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, Berlin, third edition.

Rogers, L. C. G. and Williams, D. (2000). *Diffusions, Markov processes, and martingales. Vol. 2*. Cambridge Mathematical Library. Cambridge University Press, Cambridge. Itô calculus, Reprint of the second (1994) edition.

Ruschendorf, L. et al. (1995). Convergence of the iterative proportional fitting procedure. *The Annals of Statistics*, 23(4):1160–1174.

Rüschendorf, L. and Thomsen, W. (1993). Note on the Schrödinger equation and i-projections. *Statistics & Probability letters*, 17(5):369–375.

Saharia, C., Ho, J., Chan, W., Salimans, T., Fleet, D. J., and Norouzi, M. (2021). Image super-resolution via iterative refinement. *arXiv preprint arXiv:2104.07636*.

San-Roman, R., Nachmani, E., and Wolf, L. (2021). Noise estimation for generative diffusion models. *arXiv preprint arXiv:2104.02600*.

Schrödinger, E. (1932). Sur la théorie relativiste de l'électron et l'interprétation de la mécanique quantique. *Annales de l'Institut Henri Poincaré*, 2(4):269–310.

Sinkhorn, R. and Knopp, P. (1967). Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348.

Skilling, J. (1989). The eigenvalues of mega-dimensional matrices. In *Maximum Entropy and Bayesian Methods*, pages 455–466. Springer.

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

Song, J., Meng, C., and Ermon, S. (2020). Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

Song, Y. and Ermon, S. (2019). Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*.

Song, Y. and Ermon, S. (2020). Improved techniques for training score-based generative models. In *Advances in Neural Information Processing Systems*.

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

Tzen, B. and Raginsky, M. (2019). Theoretical guarantees for sampling and inference in generative models with latent diffusions. *Conference on Learning Theory*, 99:3084–3114.

Vargas, F., Thodoroff, P., Lawrence, N. D., and Lamacraft, A. (2021). Solving Schrödinger bridges via maximum likelihood. *arXiv preprint arXiv:2106.02081*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*.

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

Wang, G., Jiao, Y., Xu, Q., Wang, Y., and Yang, C. (2021). Deep generative learning via Schrödinger bridge. In *International Conference on Machine Learning*.

Watson, D., Ho, J., Norouzi, M., and Chan, W. (2021). Learning to efficiently sample from diffusion probabilistic models. *arXiv preprint arXiv:2106.03802*.

A Organization of the supplementary

The supplementary is organized as follows. We define our notation in Appendix B. In Appendix C, we prove Theorem 1 and draw links between our approach of SGM and existing works. We recall the classical formulation of IPF, prove Proposition 2 and draw links with autoencoders in Appendix D. In Appendix E we present alternative variational formulas for Algorithm 1 and prove Proposition 3. We gather the proofs of our theoretical study of Schrödinger bridges (Proposition 4 and Proposition 5) in Appendix F. A quantitative study of IPF with Gaussian targets and reference measure is presented in Appendix G. In particular, we show that the convergence rate of IPF is geometric in this case. In Appendix H we study the links between continuous-time and discrete-time IPF and prove Proposition 6. We also provide details on the likelihood computation of generative models obtained with Schrödinger bridges. We detail training techniques to improve training times in Appendix I then present architecture details and additional experiments in Appendix J.

B Notation

For ease of reading in this section we recall and detail some of the notation introduced in Section 1. For any measurable space $(\mathsf{E}, \mathcal{E})$, we denote by $\mathscr{P}(\mathsf{E})$ the space of probability measures over E. For any $\ell \in \mathbb{N}$, we also denote $\mathscr{P}_\ell = \mathscr{P}((\mathbb{R}^d)^\ell)$. For any $\pi \in \mathscr{P}(\mathsf{E})$ and Markov kernel $\mathrm{K} : \mathsf{E} \times \mathcal{F} \to [0,1]$ where $(\mathsf{F}, \mathcal{F})$ is a measurable space, we define $\pi\mathrm{K} \in \mathscr{P}(\mathsf{F})$ such that for any $\mathsf{A} \in \mathcal{F}$ we have $\pi\mathrm{K}(\mathsf{A}) = \int_{\mathsf{E}} \mathrm{K}(x, \mathsf{A}) \mathrm{d}\pi(x)$. If $\mathsf{E} = \mathcal{C}$ then for any $\mathbb{P} \in \mathscr{P}(\mathsf{E})$ and $s, t \in [0,T]$, we denote by $\mathbb{P}_{s,t}$ the marginals of $\mathbb{P}$ at time s and t. In addition, we denote by $\mathbb{P}_{|s,t}$ the disintegration Markov kernel given by the mapping $\omega \mapsto (\omega(s), \omega(t))$, see Appendix D.1 for a definition. In particular, we have $\mathbb{P} = \mathbb{P}_{s,t}\mathbb{P}_{|s,t}$. All defined mappings are considered to be measurable unless stated otherwise.

For any $\mathbb{P} \in \mathscr{P}(\mathcal{C})$ we define $\mathbb{P}^R$ the reverse-time measure, *i.e.* for any $\mathsf{A} \in \mathcal{B}(\mathcal{C})$ we have $\mathbb{P}^R(\mathsf{A}) = \mathbb{P}(\mathsf{A}^R)$ where $\mathsf{A}^R = \{t \mapsto \omega(T-t) \ : \ \omega \in \mathsf{A}\}$. We say that $\mathbb{P} \in \mathscr{P}(\mathcal{C})$ is *associated with a diffusion* if it solves the corresponding martingale problem. More precisely, $\mathbb{P} \in \mathscr{P}(\mathcal{C})$ is associated with $\mathrm{d}\mathbf{X}_t = b(t, \mathbf{X}_t)\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t$ for $b : [0,T] \times \mathbb{R}^d \to \mathbb{R}^d$ measurable if for any $v \in \mathrm{C}^2_c(\mathbb{R}^d, \mathbb{R})$, $(M^v_t)_{t\in[0,T]}$ is a $\mathbb{P}$-local martingale, where for any $t \in [0,T]$

$$M^v_t = v(\mathbf{X}_t) - \int_0^t \mathcal{A}_s(v)(\mathbf{X}_s)\mathrm{d}s \tag{16}$$

with for any $v \in \mathrm{C}^2(\mathbb{R}^d, \mathbb{R})$, $t \in [0,t]$ and $x \in \mathbb{R}^d$

$$\mathcal{A}_t(v)(x) = \langle b(t,x), \nabla v(x) \rangle + \Delta v(x).$$

We refer to Revuz and Yor (1999) for a rigorous treatment of local martingales. Note that (16) uniquely defines $\mathbb{P}_{t|s}$ for any $s, t \in [0,T]$ with $t \geq s$. Hence $\mathbb{P}$ is uniquely defined up to $\mathbb{P}_0$.

In some cases, we say that $\mathbb{P} \in \mathscr{P}(\mathcal{C})$ is *associated with a diffusion* if it solves the corresponding martingale problem with initial condition. More precisely, $\mathbb{P} \in \mathscr{P}(\mathcal{C})$ is associated with $\mathrm{d}\mathbf{X}_t = b(t, \mathbf{X}_t)\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t$ and $\mathbf{X}_0 \sim \mu_0 \in \mathscr{P}(\mathbb{R}^d)$ if it solves the martingale problem and $\mathbb{P}_0 = \mu_0$. Note that in this case $\mathbb{P}$ is uniquely defined.

Finally, for any measurable space $(\mathsf{E}, \mathcal{E})$ and $\mu, \nu \in \mathscr{P}(\mathsf{E})$ we recall that the Jeffrey's divergence is given by $\mathrm{J}(\mu, \nu) = \mathrm{KL}(\mu|\nu) + \mathrm{KL}(\nu|\mu)$.

C Time-reversal and existing work

Before giving the proof of Theorem 1 we start by deriving estimates on the logarithmic derivatives of the density of the Ornstein-Ulhenbeck process given growth conditions on the initial density in Appendix C.1. Note that our estimates are uniform w.r.t. the time variable. We give the proof of Theorem 1 in Appendix C.2. Finally, we draw links with existing works in Appendix C.3.

C.1 Estimates for logarithmic derivatives

We start by recalling the following multivariate Faa di Bruno's formula and a useful technical lemma. Then in Appendix C.1.1 we derive bounds for the logarithmic derivatives which are non-vacuous

for small times. In Appendix C.1.2 we derive bounds for the logarithmic derivatives which are non-vacuous for large times. We combine them in Appendix C.1.3.

For any $\alpha \in \mathbb{N}^d$ we denote $|\alpha| = \sum_{i=1}^d \alpha_i$ and $\alpha! = \prod_{i=1}^d \alpha_i!$. If $f : \mathbb{R}^d \to \mathbb{R}$ is m-differentiable with $m \in \mathbb{N}$, then for any $\lambda \in \mathbb{N}^d$ with $|\lambda| \leq m$ we denote for any $x \in \mathbb{R}^d$, $\partial_\lambda f(x) = \partial_1^{\lambda_1} \dots \partial_d^{\lambda_d} f(x)$. Similarly to Constantine and Savits (1996), we define $\prec$ the order on $\mathbb{N}^d$ such that for any $\lambda^1, \lambda^2 \in \mathbb{N}^d$, $\lambda^1 \prec \lambda^2$ if $|\lambda^1| < |\lambda^2|$ or $|\lambda^1| = |\lambda^2|$ and there exists $j \in \{1, \dots, d\}$ such that $\lambda_j^1 < \lambda_j^2$ and for any $i \in \{1, \dots, j\}$, $\lambda_i^1 = \lambda_i^2$.

Proposition 7. *Let $\mathsf{U} \subset \mathbb{R}$ open, $N \in \mathbb{N}$, $f \in \mathrm{C}^N(\mathsf{U}, \mathbb{R})$, $g \in \mathrm{C}^N(\mathbb{R}^d, \mathsf{U})$ and $h = f \circ g$. Then for any $\lambda \in \mathbb{N}^d$ with $|\lambda| \leq N$ and $x \in \mathbb{R}^d$ we have*

$$\partial_\lambda h(x) = \sum_{k,s=1}^{|\lambda|} \sum_{p_s(\lambda,k)} f^{(k)}(g(x)) \lambda! \prod_{j=1}^s \partial_{\ell_j} g(x)^{m_j} / (m_j! \ell_j!^{m_j}),$$

with

$$p_s(\lambda, k) = \{\{\ell_i\}_{i=1}^s \in (\mathbb{N}^d)^s,\ \{m_i\}_{i=1}^s \in \mathbb{N}^s\ :\ \ell_1 \prec \dots \prec \ell_s,\ \sum_{i=1}^s m_i = k,\ \sum_{i=1}^s m_i \ell_i = \lambda\}.$$

Proof. The proposition is a direct application of Constantine and Savits (1996). □

From this multivariate Faa di Bruno formula we derive the following lemma drawing links between exponential and logarithmic derivatives.

Lemma 8. *Let $N \in \mathbb{N}$, $g_1 \in \mathrm{C}^N(\mathbb{R}^d, \mathbb{R})$, $g_2 \in \mathrm{C}^N(\mathbb{R}^d, (0, +\infty))$, $h_1 = \exp[g_1]$ and $h_2 = \log(g_2)$. Then for any $\lambda \in \mathbb{N}^d$ with $|\lambda| \leq N$ let $c_{d,\lambda} = \sum_{k=1}^{|\lambda|} d^k$ and the following hold:*

(a) There exists $\mathrm{P}_{\lambda,\exp}$ a real polynomial with $c_{d,\lambda}$ variables such that for any $x \in \mathbb{R}^d$

$$\partial_\lambda h_1(x) = \mathrm{P}_{\lambda,\exp}((\partial_\ell g_1(x))_{|\ell| \leq |\lambda|}) h_1(x).$$

(b) There exists $\mathrm{P}_{\lambda,\log}$ a real polynomial with $c_{d,\lambda}$ variables such that for any $x \in \mathbb{R}^d$

$$\partial_\lambda h_2(x) = \mathrm{P}_{\lambda,\log}((\partial_\ell g_2(x)/g_2(x))_{|\ell| \leq |\lambda|}).$$

Proof. The proof of (a) is a direct application of Proposition 7 upon noting that for any $k \in \mathbb{N}$, $f^{(k)} = \exp$ if $f = \exp$. Similarly, the proof of (b) is a direct application of Proposition 7 upon noting that, in the case where $f = \log$, for any $k \in \mathbb{N}$ and $x > 0$, $f^{(k)}(x) = (-1)^{k-1}(k-1)! x^{-k}$ and that for any $s \in \{1, \dots, |\lambda|\}$ and $(\ell_1, \dots, \ell_s, m_1, \dots, m_s) \in p_s(\lambda, k)$ we have $\sum_{i=1}^s m_i = k$. □

We will also make use of the following technical lemma.

Lemma 9. *Let $p \in \mathbb{N}$. Then for any $a \geq 0$, $b > 0$ and $x \in \mathbb{R}^d$ we have*

$$-b\|x\|^{2p} + a\|x\|^{2p-1} \leq -(b/2)\|x\|^{2p} + a(2a/b)^{2p-1}, \tag{17}$$

$$-b\|x\|^{2p} + a\|x\|^{2p-2} \leq -(b/2)\|x\|^{2p} + a(2a/b)^{p-1}. \tag{18}$$

In addition for any $a \geq 0$, $b > 0$ and $x \in \mathbb{R}^d$ we have

$$-b\|x\|^{2p} + a\|x\|^{2p-1} \leq (2p-1)^{2p-1}(2p)^{-2p} a^{2p} b^{1-2p}.$$

Proof. For the first part of the proof, we only prove (17). The proof of (18) is similar. Let $a \geq 0$, $b > 0$. For any $x \in \mathbb{R}^d$ with $\|x\| \leq (b/2a)^{-1}$ we have $a\|x\|^{2p-1} \leq a(b/2a)^{-2p+1}$. For any $x \in \mathbb{R}^d$ with $\|x\| \geq (b/2a)^{-1}$ we have $a\|x\|^{2p-1} \leq (b/2)\|x\|^{2p}$. Hence, we get that for any $x \in \mathbb{R}^d$ we have

$$a\|x\|^{2p-1} - b\|x\|^{2p} \leq a(b/2a)^{-2p+1} - (b/2)\|x\|^{2p},$$

which concludes the first part of the proof. For the second part of the proof, remark that the maximum of $\mathrm{h} : t \mapsto -bt^{2p} + at^{2p-1}$ is attained for $t^\star = (2p-1)/(2p)(a/b)$. We conclude upon noting that $h(t^\star) = (2p-1)^{2p-1}(2p)^{-2p} a^{2p} b^{1-2p}$. □

C.1.1 Small times estimates

Lemma 8 is key in the following proposition which establishes upper bounds on the logarithmic derivatives of the density of the Ornstein-Ulhenbeck process. In what follows, we define $(p_t)_{t\in[0,T]}$ the density w.r.t. the Lebesgue measure of $\mathbf{X}_t$ satisfying

$$\mathrm{d}\mathbf{X}_t = -\alpha \mathbf{X}_t \mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t, \qquad \mathbf{X}_0 \sim p_{\text{data}},$$

with $\alpha \geq 0$. In the rest of this section, α is fixed.

Proposition 10. *Let $N \in \mathbb{N}$. Assume that $p_{\text{data}} \in \mathrm{C}^N(\mathbb{R}^d, (0,+\infty))$ is bounded and that for any $\ell \in \{1,\dots,N\}$ there exist $A_\ell \geq 0$ and $\alpha_\ell \in \mathbb{N}$ such that for any $x \in \mathbb{R}^d$*

$$\|\nabla^\ell \log p_{\text{data}}(x)\| \leq A_\ell(1 + \|x\|^{\alpha_\ell}). \tag{19}$$

Then for any $t \geq 0$, $p_t \in \mathrm{C}^N(\mathbb{R}^d, (0,+\infty))$ and for any $\ell \in \{1,\dots,N\}$, there exist $B_\ell \geq 0$ and $\beta_\ell \in \mathbb{N}$ such that for any $t \geq 0$

$$\|\nabla^\ell \log p_t(x)\| \leq c_t^{-2\beta_\ell} B_\ell(1 + \textstyle\int_{\mathbb{R}^d} \|x_0\|^{\beta_\ell} p_{0|t}(x_0|x_t)\mathrm{d}x_0),$$

with $c_t^2 = \exp[-2\alpha t]$.

Proof. First note that for any $t \geq 0$ and $x_t \in \mathbb{R}^d$ we have

$$p_t(x_t) = \textstyle\int_{\mathbb{R}^d} p_{\text{data}}(x_0)\mathrm{g}(x_t - c_t x_0)\mathrm{d}x_0, \tag{20}$$

with for any $\tilde{x} \in \mathbb{R}^d$

$$c_t = \exp[-\alpha t], \qquad \mathrm{g}(\tilde{x}) = (2\pi\sigma_t^2)^{-d/2}\exp[-\|\tilde{x}\|^2/(2\sigma_t^2)], \quad \sigma_t^2 = (1-\exp[-2\alpha t])/\alpha.$$

Let $t \geq 0$. We have that $p_t \in \mathrm{C}^N(\mathbb{R}^d, (0,+\infty))$ upon combining the fact that p_{data} is bounded, (20) and the dominated convergence theorem. Let $\ell \in \{1,\dots,N\}$ and $\lambda \in \mathbb{N}^d$ such that $|\lambda| \leq \ell$. Using Lemma 8-(b) we have for any $x_t \in \mathbb{R}^d$

$$\partial_\lambda \log p_t(x_t) = \mathrm{P}_{\lambda,\log}((\partial_m p_t(x_t)/p_t(x_t))_{|m|\leq|\lambda|}). \tag{21}$$

Using (20) and the change of variable $z = x_t - c_t x_0$, we have for any $x_t \in \mathbb{R}^d$

$$p_t(x_t) = c_t^{-1}\textstyle\int_{\mathbb{R}^d} p_{\text{data}}((x_t - z)/c_t)\mathrm{g}(z)\mathrm{d}z.$$

Hence, combining this result, the dominated convergence theorem and Lemma 8-(a) we get that for any $x_t \in \mathbb{R}^d$ and $m \in \mathbb{N}^d$ with $|m| \leq \ell$

$$\begin{aligned}\partial_m p_t(x_t) &= c_t^{-|m|}\textstyle\int_{\mathbb{R}^d} \partial_m p_{\text{data}}(x_0)\mathrm{g}(x_t - c_t x_0)\mathrm{d}x_0 \\ &= c_t^{-|m|}\textstyle\int_{\mathbb{R}^d} \mathrm{P}_{m,\exp}((\partial_j \log p_{\text{data}}(x_0))_{|j|\leq|m|}) p_{\text{data}}(x_0)\mathrm{g}(x_t - c_t x_0)\mathrm{d}x_0.\end{aligned}$$

We conclude the proof upon combining this result, (19), (21) and the fact that $c_t \leq 1$. □

For any $t \geq 0$ and $x_t \in \mathbb{R}^d$ we introduce the infinitesimal generator $\mathcal{A}_{t,x_t} : \mathrm{C}_2(\mathbb{R}^d,\mathbb{R}) \to \mathrm{C}_2(\mathbb{R}^d,\mathbb{R})$ given for any $\varphi \in \mathrm{C}^2(\mathbb{R}^d,\mathbb{R})$ and $x_0 \in \mathbb{R}^d$ by

$$\begin{aligned}\mathcal{A}_{t,x_t}(\varphi)(x_0) &= \langle \nabla_{x_0} \log p_{0|t}(x_0|x_t), \nabla\varphi(x_0)\rangle + \Delta\varphi(x_0) \\ &= \langle \nabla \log p_{\text{data}}(x_0), \nabla\varphi(x_0)\rangle + (c_t/\sigma_t^2)\langle x_t - c_t x_0, \nabla\varphi(x_0)\rangle + \Delta\varphi(x_0).\end{aligned} \tag{22}$$

Establishing Foster-Lyapunov drift condition for this infinitesimal generator will allow us to derive moment bounds for $x_0 \mapsto p_{0|t}(x_0|x_t)$. We now introduce the Lyapunov functional which will allow us to control these moments. For any $p \in \mathbb{N}$, $t > 0$ and $x_t \in \mathbb{R}^d$, let $V_{p,t,x_t} : \mathbb{R}^d \to [1,+\infty)$ given for any $x_0 \in \mathbb{R}^d$ by

$$V_{p,t,x_t}(x_0) = 1 + \|x_0 - x_t/c_t\|^{2p}, \qquad c_t = \exp[-\alpha t].$$

Proposition 11. *Assume $p_{\text{data}} \in \mathrm{C}^1(\mathbb{R}^d,\mathbb{R})$ and that there exist $\mathtt{m}_0 > 0$, $d_0, C_0 \geq 0$ such that for any $x_0 \in \mathbb{R}^d$ we have*

$$\langle x_0, \nabla \log p_{\text{data}}(x_0)\rangle \leq -\mathtt{m}_0\|x_0\|^2 + d_0\|x_0\|, \quad \|\nabla \log p_{\text{data}}(x_0)\| \leq C_0(1+\|x_0\|). \tag{23}$$

Then for any $t > 0$, $x_t \in \mathbb{R}^d$ and $p \in \mathbb{N}$ there exist $\beta_p \in \mathbb{N}$, $a_p > 0$ and $b_p \geq 0$ (independent of t and x_t) such that for any $x_0 \in \mathbb{R}^d$ we have

$$\mathcal{A}_{t,x_t}(V_{p,t,x_t})(x_0) \leq -a_p V_{p,t,x_t}(x_0) + b_p(1 + \|x_t/c_t\|^{\beta_p}),$$

with $\beta_p = 2p$.

Proof. Let $t \geq 0$, $x_0, x_t \in \mathbb{R}^d$ and $p \in \mathbb{N}$. First, we have for any $x_0 \in \mathbb{R}^d$

$$V_{p,t,x_t}(x_0) = \|x_0 - x_t/c_t\|^{2p}, \quad \nabla V_{p,t,x_t}(x_0) = 2p(x_0 - x_t/c_t)\|x_0 - x_t/c_t\|^{2(p-1)}, \quad (24)$$
$$\Delta V_{p,t,x_t}(x_0) = 2p(2p-1)\|x_0 - x_t/c_t\|^{2(p-1)}.$$

Second, using Lemma 9, the Cauchy-Schwarz inequality and (23), we have for any $x_0 \in \mathbb{R}^d$

$$\begin{aligned}
\langle \nabla \log p_{\text{data}}(x_0), x_0 - x_t/c_t \rangle &\leq -\mathtt{m}_0\|x_0\|^2 + d_0\|x_0\| + \|\nabla \log p_{\text{data}}(x_0)\|\|x_t/c_t\| \\
&\leq -\mathtt{m}_0\|x_0 - x_t/c_t\|^2 + 2\mathtt{m}_0\|x_0\|\|x_t\|/c_t + C_0(1 + \|x_0\|)\|x_t\|/c_t \\
&\quad + d_0\|x_0 - x_t/c_t\| + d_0\|x_t\|/c_t + \mathtt{m}_0\|x_t\|^2/c_t^2 \\
&\leq -\mathtt{m}_0\|x_0 - x_t/c_t\|^2 + \{(2\mathtt{m}_0 + C_0)\|x_t\|/c_t + d_0\}\|x_0 - x_t/c_t\| \\
&\quad + (3\mathtt{m}_0 + C_0)\|x_t\|^2/c_t^2 + (C_0 + d_0)\|x_t\|/c_t.
\end{aligned}$$

Combining this result and (24), we have for any $x_0 \in \mathbb{R}^d$

$$\begin{aligned}
&\langle \nabla \log p_{\text{data}}(x_0), \nabla V_{p,t,x_t}(x_0) \rangle \\
&\quad \leq -2p\mathtt{m}_0\|x_0 - x_t/c_t\|^{2p} + 2p\{(2\mathtt{m}_0 + C_0)\|x_t\|/c_t + d_0\}\|x_0 - x_t/c_t\|^{2p-1} \\
&\quad\quad + 2p((3\mathtt{m}_0 + C_0)\|x_t\|^2/c_t^2 + (C_0 + d_0)\|x_t\|/c_t)\|x_0 - x_t/c_t\|^{2p-2}.
\end{aligned}$$

Combining this result with (22) and the fact that for any $x_0 \in \mathbb{R}^d$, $(c_t/\sigma_t^2)\langle x_t - c_t x_0, \nabla V_{p,t,x_t}(x_0)\rangle \leq 0$, we get that for any $x_0 \in \mathbb{R}^d$

$$\begin{aligned}
\mathcal{A}_{t,x_t}(V_{p,t,x_t})(x_0) &\leq -2p\mathtt{m}_0\|x_0 - x_t/c_t\|^{2p} + 2p\{(2\mathtt{m}_0 + C_0)\|x_t\|/c_t + d_0\}\|x_0 - x_t/c_t\|^{2p-1} \\
&\quad + 2p((3\mathtt{m}_0 + C_0)\|x_t\|^2/c_t^2 + (C_0 + d_0)\|x_t\|/c_t)\|x_0 - x_t/c_t\|^{2p-2}.
\end{aligned}$$

Using Lemma 9 there exist $\beta_p \in \mathbb{N}$, $a_p > 0$ and $b_p \geq 0$ (independent of x_t and t) such that for any $x_0 \in \mathbb{R}^d$ we have

$$\mathcal{A}_{t,x_t}(V_{p,t,x_t})(x_0) \leq -a_p V_{p,t,x_t}(x_0) + b_p(1 + (\|x_t\|/c_t)^{\beta_p}),$$

which concludes the proof. □

Using this Foster-Lyapunov drift we are now ready to bound the moments of $x_0 \mapsto p_{0|t}(x_0|x_t)$.

Proposition 12. *Assume that $p_{\text{data}} \in \mathrm{C}^2(\mathbb{R}^d, \mathbb{R})$ and that there exist $\mathtt{m}_0 > 0$, $d_0, C_0 \geq 0$ such that for any $x_0 \in \mathbb{R}^d$ we have*

$$\langle x_0, \nabla \log p_{\text{data}}(x_0)\rangle \leq -\mathtt{m}_0\|x_0\|^2 + d_0\|x_0\|, \quad \|\nabla \log p_{\text{data}}(x_0)\| \leq C_0(1 + \|x_0\|).$$

Then, for any $p \in \mathbb{N}$ there exist $C_p \geq 0$ and $\beta_p \in \mathbb{N}$ such that for any $t \geq 0$ and $x_t \in \mathbb{R}^d$

$$\int_{\mathbb{R}^d} \|x_0\|^p \, p(x_0|x_t)\mathrm{d}x_0 \leq C_p c_t^{-2\beta_p}(1 + \|x_t\|^{\beta_p}), \quad (25)$$

with $c_t^2 = \exp[-2\alpha t]$ and $\beta_p = p$.

Proof. Let $t \geq 0$ and $x_t \in \mathbb{R}^d$. Using (Ikeda and Watanabe, 1989, Theorem 2.3, Theorem 3.1), Proposition 11 and (Meyn and Tweedie, 1993, Theorem 2.1) for any $x \in \mathbb{R}^d$, there exists a unique strong solution $(\mathbf{X}_u^x)_{u \geq 0}$ such that $\mathbf{X}_0^x \sim \delta_x$ and

$$\mathrm{d}\mathbf{X}_u^x = \nabla \log p_{0|t}(\mathbf{X}_u^x|x_t)\mathrm{d}u + \sqrt{2}\mathrm{d}\mathbf{B}_u.$$

Using (Leha and Ritter, 1984, Theorem 5.19) we get that $\{(\mathbf{X}_u^x)_{u\geq 0} : x \in \mathbb{R}^d\}$ is associated with a Feller semi-group. In addition, we have that for any $f \in \mathrm{C}_c^2(\mathbb{R}^d)$, $\int_{\mathbb{R}^d} \mathcal{A}_{t,x_t}(f)(x_0)p_{0|t}(x_0|x_t)\mathrm{d}x_0 = 0$. Therefore, using (Revuz and Yor, 1999, Proposition 1.5) and (Ethier and Kurtz, 1986, Theorem 9.17) we get that the probability distribution with density $x_0 \mapsto p_{0|t}(x_0|x_t)$ is an invariant distribution for the semi-group associated with $\{(\mathbf{X}_u^x)_{u\geq 0} : x \in \mathbb{R}^d\}$. Therefore, using Proposition 11 and (Meyn and Tweedie, 1993, Theorem 4.6) we get that for any $p \in \mathbb{N}$

$$\int_{\mathbb{R}^d}(1 + \|x_0 - c_t^{-1}x_t\|^{2p})p_{0|t}(x_0|x_t)\mathrm{d}x_0 \leq b_p(1 + \|x_t/c_t\|^{\beta_p})/a_p$$

which concludes the proof upon using that $c_t \leq 1$ and Jensen's inequality. □

C.1.2 Large times estimates

In Proposition 12, the bound in (25) goes to $+\infty$ as $t \to +\infty$ since $\lim_{t\to+\infty} c_t^{-1} = +\infty$ (if $\alpha > 0$). This does not yield any degeneracy in our setting since we consider a fixed time horizon $T > 0$. However, we can improve the result by deriving another bound which is bounded at $t \to +\infty$ but explodes as $t \to 0$. In this section we assume that $h : u \mapsto (\exp[u] - 1)/u$ is extended to 0 by continuity with $h(0) = 1$.

The following proposition is the equivalent of Proposition 10 with a bound which explodes for $t \to 0$ instead of $t \to +\infty$. Note that contrary to Proposition 10 we do not require any differentiability condition the initial distribution p_{data}.

Proposition 13. *Let $N \in \mathbb{N}$. Assume that $p_{\text{data}} \in \mathrm{C}^0(\mathbb{R}^d, (0, +\infty))$ is bounded. Then for any $t \geq 0$, $p_t \in \mathrm{C}^N(\mathbb{R}^d, (0, +\infty))$ and for any $\ell \in \{1, \dots, N\}$, there exist $B_\ell \geq 0$ and $\beta_\ell \in \mathbb{N}$ such that for any $t \geq 0$*

$$\begin{aligned}\|\nabla^\ell \log p_t(x)\| &\leq \sigma_t^{-\beta_\ell} B_\ell (1 + \textstyle\int_{\mathbb{R}^d} \|x_t - c_t x_0\|^{\beta_\ell} \, p_{0|t}(x_0|x_t) \mathrm{d}x_0) \\ &\leq \sigma_t^{-\beta_\ell} B_\ell (1 + \textstyle\int_{\mathbb{R}^d} \|x_t - x_0\|^{\beta_\ell} \, q_{0|t}(x_0|x_t) \mathrm{d}x_0).\end{aligned}$$

with $\sigma_t^2 = (1 - \exp[-2\alpha t])/\alpha$ and for any $\tilde{x} \in \mathbb{R}^d$

$$\begin{aligned}q_{0|t}(x_0|x_t) &= p_{\text{data}}(x_0/c_t)\mathrm{g}(x_t - x_0) / \textstyle\int_{\mathbb{R}^d} p_{\text{data}}(x_0/c_t)\mathrm{g}(x_t - x_0)\mathrm{d}x_0, \\ \mathrm{g}(\tilde{x}) &= (2\pi\sigma_t^2) \exp[-\|\tilde{x}\|^2 / (2\sigma_t^2)].\end{aligned}$$

Proof. First note that for any $t \geq 0$ and $x_t \in \mathbb{R}^d$ we have

$$p_t(x_t) = \textstyle\int_{\mathbb{R}^d} p_{\text{data}}(x_0)\mathrm{g}(x_t - c_t x_0)\mathrm{d}x_0, \tag{26}$$

with

$$c_t = \exp[-\alpha t], \qquad \mathrm{g}(\tilde{x}) = (2\pi\sigma_t^2)^{-d/2} \exp[-\|\tilde{x}\|^2 / (2\sigma_t^2)], \quad \sigma_t^2 = (1 - \exp[-2\alpha t])/\alpha.$$

Let $t \geq 0$. We have $p_t \in \mathrm{C}^N(\mathbb{R}^d, (0, +\infty))$ upon combining the fact that p_{data} is bounded, (26) and the dominated convergence theorem. Let $\ell \in \{0, \dots, N\}$ and $\lambda \in \mathbb{N}^d$ such that $|\lambda| \leq \ell$. Using Lemma 8-(b) we have for any $x_t \in \mathbb{R}^d$

$$\partial_\lambda \log p_t(x_t) = \mathrm{P}_{\lambda,\log}((\partial_m p_t(x_t)/p_t(x_t))_{|m| \leq |\lambda|}).$$

For any $m \in \mathbb{N}^d$ with $|m| \leq |\lambda|$, using the dominated convergence theorem, there exist $C_m \geq 0$ and $\beta_m \in \mathbb{N}$ such that for any $x_t \in \mathbb{R}^d$ we have

$$|\partial_m p_t(x_t)| \leq C_m \sigma_t^{-2\beta_m} \textstyle\int_{\mathbb{R}^d} (1 + \|x_t - c_t x_0\|^{\beta_m}) p_{\text{data}}(x_0)\mathrm{g}(x_t - c_t x_0)\mathrm{d}x_0,$$

which concludes the proof. □

For any $t \geq 0$ and $x_t \in \mathbb{R}^d$ we introduce the infinitesimal generator $\tilde{\mathcal{A}}_{t,x_t} : \mathrm{C}_2(\mathbb{R}^d, \mathbb{R}) \to \mathrm{C}_2(\mathbb{R}^d, \mathbb{R})$ given for any $\varphi \in \mathrm{C}^2(\mathbb{R}^d, \mathbb{R})$ and $x_0 \in \mathbb{R}^d$ by

$$\begin{aligned}\tilde{\mathcal{A}}_{t,x_t}(f)(x_0) &= \langle \nabla \log q_{0|t}(x_0|x_t), \nabla\varphi(x_0)\rangle + \Delta\varphi(x_0) \\ &= c_t^{-1}\langle \nabla \log p_{\text{data}}(x_0/c_t), \nabla\varphi(x_0)\rangle + \sigma_t^{-2}\langle x_t - x_0, \nabla\varphi(x_0)\rangle + \Delta\varphi(x_0).\end{aligned}$$

For any $p \in \mathbb{N}$, let $V_p : \mathbb{R}^d \to [1, +\infty)$ given for any $x_0 \in \mathbb{R}^d$ by

$$V_p(x_0) = 1 + \|x_0\|^{2p}.$$

The following proposition is the counterpart to Proposition 11.

Proposition 14. *Assume that $p_{\text{data}} \in \mathrm{C}^1(\mathbb{R}^d, \mathbb{R})$ and that there exist $\mathtt{m}_0 > 0$, $d_0 \geq 0$ such that for any $x_0 \in \mathbb{R}^d$ we have*

$$\langle x_0, \nabla \log p_{\text{data}}(x_0)\rangle \leq -\mathtt{m}_0\|x_0\|^2 + d_0 \|x_0\| . \tag{27}$$

Then for any $t > 0$, $x_t \in \mathbb{R}^d$ and $p \in \mathbb{N}$ there exist $\beta_p \in \mathbb{N}$, $a_p > 0$ and $b_p \geq 0$ (independent of t and x_t) such that for any $x_0 \in \mathbb{R}^d$ we have

$$\tilde{\mathcal{A}}_{t,x_t}(V_p)(x_0) \leq -a_p\sigma_t^{-2}V_p(x_0) + b_p(1 + \|x_t/\sigma_t^2\|^{\beta_p}),$$

with $\beta_p = 2p$.

Proof. Let $t \geq 0$, $x_0, x_t \in \mathbb{R}^d$ and $p \in \mathbb{N}$. First, we have for any $x_0 \in \mathbb{R}^d$

$$V_p(x_0) = 1 + \|x_0\|^{2p}, \qquad \nabla V_p(x_0) = 2p \|x_0\|^{2(p-1)} x_0, \qquad \Delta V_p(x_0) = 2p(2p-1) \|x_0\|^{2(p-1)}.$$

Using this result, (27) and Lemma 9, we get that for any $x_0 \in \mathbb{R}^d$

$$\begin{aligned} 2p\langle \nabla \log p_{\text{data}}(x_0/c_t), x_0/c_t \rangle \|x_0\|^{2(p-1)} &\leq 2pc_t^{-1}(-\mathtt{m}_0 \|x_0\|^{2p}/c_t + d_0 \|x_0\|^{2p-1}) \\ &\leq c_t^{-1}(2p-1)^{2p-1}(2p)^{1-2p}(\mathtt{m}_0/c_t)^{1-2p} d_0^{2p}. \end{aligned}$$

Combining this result and the fact that $c_t \leq 1$, there exists $d_p \geq 0$ (independent from t and x_t) such that for any $x_0 \in \mathbb{R}^d$

$$2p\langle \nabla \log p_{\text{data}}(x_0/c_t), x_0/c_t \rangle \|x_0\|^{2(p-1)} \leq d_p. \tag{28}$$

In addition, we have for any $x_0 \in \mathbb{R}^d$

$$\begin{aligned} &(2p/\sigma_t^2)\langle x_0, x_t - x_0 \rangle \|x_0\|^{2(p-1)} + 2p(2p-1)\|x_0\|^{2(p-1)} \\ &\quad \leq -(2p/\sigma_t^2)\|x_0\|^{2p} + (2p/\sigma_t^2)\|x_0\|^{2p-1}\|x_t\| + 2p(2p-1)\|x_0\|^{2p-1} + 2p(2p-1). \end{aligned}$$

Combining this result and (28) we have for any $x_0 \in \mathbb{R}^d$

$$\begin{aligned} &\tilde{\mathcal{A}}_{t,x_t}(V_p)(x_0) \\ &\quad \leq -(2p/\sigma_t^2)\|x_0\|^{2p} + (2p/\sigma_t^2)\|x_0\|^{2p-1}\|x_t\| + 2p(2p-1)\|x_0\|^{2p-1} + 2p(2p-1) + d_p. \end{aligned}$$

We conclude upon using Lemma 9. □

The next proposition is the counterpart of Proposition 12.

Proposition 15. *Assume that $p_{\text{data}} \in \mathrm{C}^2(\mathbb{R}^d, \mathbb{R})$ and that there exist $\mathtt{m}_0 > 0$, $d_0 \geq 0$ such that for any $x_0 \in \mathbb{R}^d$ we have*

$$\langle x_0, \nabla \log p_{\text{data}}(x_0) \rangle \leq -\mathtt{m}_0 \|x_0\|^2 + d_0 \|x_0\|.$$

Then, for any $p \in \mathbb{N}$ there exist $C_p \geq 0$ and $\beta_p \in \mathbb{N}$ such that for any $t \in \geq 0$ and $x_t \in \mathbb{R}^d$

$$\int_{\mathbb{R}^d} \|x_t - x_0\|^p \, q_{0|t}(x_0|x_t) \mathrm{d}x_0 \leq C_p \sigma_t^{-2\beta_p}(1 + \|x_t\|^{\beta_p}),$$

with $\sigma_t^2 = (1 - \exp[-2\alpha t])/\alpha$ and $\beta_p = p$.

Proof. The proof is similar to the one of Proposition 12. □

C.1.3 Uniform in time logarithmic derivatives estimates

In this section we combine the results of Appendix C.1.2 and Appendix C.1.1 to establish uniform in time estimates for the logarithmic derivatives of the density of the Ornstein-Ulhenbeck diffusion.

Theorem 16. *Let $N \in \mathbb{N}$ with $N \geq 2$. Assume that $p_{\text{data}} \in \mathrm{C}^N(\mathbb{R}^d, \mathbb{R})$ and that there exist $\mathtt{m}_0 > 0$, $d_0, C_0 \geq 0$ such that for any $x_0 \in \mathbb{R}^d$ we have*

$$\langle x_0, \nabla \log p_{\text{data}}(x_0) \rangle \leq -\mathtt{m}_0 \|x_0\|^2 + d_0 \|x_0\|, \quad \|\nabla \log p_{\text{data}}(x_0)\| \leq C_0(1 + \|x_0\|).$$

In addition, assume that p_{data} is bounded and that for any $\ell \in \{1, \ldots, N\}$ there exist $A_\ell \geq 0$ and $\alpha_\ell \in \mathbb{N}$ such that for any $x_0 \in \mathbb{R}^d$

$$\|\nabla^\ell \log p_{\text{data}}(x_0)\| \leq A_\ell(1 + \|x_0\|^{\alpha_\ell}). \tag{29}$$

Then for any $t \geq 0$, $p_t \in \mathrm{C}^N(\mathbb{R}^d, (0, +\infty))$ and for any $\ell \in \{1, \ldots, N\}$, there exist $D_\ell \geq 0$ and $\beta_\ell \in \mathbb{N}$ such that for any $t \geq 0$

$$\|\nabla^\ell \log p_t(x_t)\| \leq D_\ell(1 + \|x_t\|^{\beta_\ell}).$$

In particular if $\alpha_1 = 1$ then $\beta_1 = 1$.

Proof. Let $t \geq 0$ and $\ell \in \{1, \dots, N\}$. Using Proposition 10 and Proposition 12 there exist $D_\ell^1 \geq 0$ and $\beta_\ell^1 \in \mathbb{N}$ such that for any $x_t \in \mathbb{R}^d$ we have

$$\|\nabla^\ell \log p_t(x_t)\| \leq D_\ell^1 c_t^{-2\beta_\ell^1}(1 + \|x_t\|^{\beta_\ell^1}).$$

Similarly, using Proposition 13 and Proposition 15 there exist $D_\ell^2 \geq 0$ and $\beta_\ell^2 \in \mathbb{N}$ such that for any $x_t \in \mathbb{R}^d$ we have

$$\|\nabla^\ell \log p_t(x_t)\| \leq D_\ell^2 (\alpha^{1/2}\sigma_t)^{-2\beta_\ell^2}(1 + \|x_t\|^{\beta_\ell^2}).$$

Therefore, there exist $\tilde{D}_\ell \geq 0$ and $\beta_\ell \in \mathbb{N}$ such that for any $x_t \in \mathbb{R}^d$ we have

$$\|\nabla^\ell \log p_t(x_t)\| \leq \tilde{D}_\ell \min(\alpha^{-1}\sigma_t^{-2}, c_t^{-2})^{\beta_\ell}(1 + \|x_t\|^{\beta_\ell}).$$

Since for any $c_t^{-2} = \exp[2\alpha t]$ and $\alpha^{-1}\sigma_t^{-2} = (1 - \exp[-2\alpha t])^{-1}$. Hence we have

$$\min(\alpha^{-1}\sigma_t^{-2}, c_t^{-2})^{\beta_\ell} \leq \max\{\min(1/u, 1/(1-u)) \ : \ u \in [0,1]\} \leq 2^{\beta_\ell},$$

which concludes the first part proof. We now show that if $\alpha_1 = 1$ then $\beta_1 = 1$. Recall that for any $t \geq 0$ and $x_t \in \mathbb{R}^d$ we have

$$p_t(x_t) = \textstyle\int_{\mathbb{R}^d} p_{\text{data}}(x_0)\mathrm{g}(x_t - c_t x_0)\mathrm{d}x_0,$$

with for any $\tilde{x} \in \mathbb{R}^d$

$$c_t = \exp[-\alpha t], \qquad \mathrm{g}(\tilde{x}) = (2\pi\sigma_t^2)^{-d/2}\exp[-\|\tilde{x}\|^2/(2\sigma_t^2)], \quad \sigma_t^2 = (1 - \exp[-2\alpha t])/\alpha.$$

Therefore, using the dominated convergence theorem we get that for any $x_t \in \mathbb{R}^d$

$$\nabla \log p_t(x_t) = \sigma_t^{-2}\textstyle\int_{\mathbb{R}^d}(x_t - c_t x_0)p_{0|t}(x_0|x_t)\mathrm{d}x_0 = \sigma_t^{-2}\int_{\mathbb{R}^d}(x_t - c_t x_0)q_{0|t}(x_0|x_t)\mathrm{d}x_0. \quad (30)$$

Similarly, using the dominate convergence theorem and change of variable $z = x_t - c_t x_0$, we have for any $x_t \in \mathbb{R}^d$

$$\nabla \log p_t(x_t) = c_t^{-1}\textstyle\int_{\mathbb{R}^d} \nabla \log p_{\text{data}}(x_0)p_{0|t}(x_0|x_t)\mathrm{d}x_0.$$

We conclude the proof upon combining this result, (30), (29) with $\alpha_1 = 1$, Proposition 15 and Proposition 12. In particular, we use that $\beta_1 = 1$. □

C.2 Proof of Theorem 1

We start by recalling the following basic lemma.

Lemma 17. *Let $(\mathsf{E}, \mathcal{E})$ and $(\mathsf{F}, \mathcal{F})$ be two measurable spaces and $\mathrm{K} : \mathsf{E} \times \mathcal{F} \to [0,1]$ be a Markov kernel. Then for any $\mu_0, \mu_1 \in \mathscr{P}(\mathsf{E})$ we have*

$$\|\mu_0\mathrm{K} - \mu_1\mathrm{K}\|_{\mathrm{TV}} \leq \|\mu_0 - \mu_1\|_{\mathrm{TV}}.$$

In addition, for any $\varphi : \mathsf{E} \to \mathsf{F}$ measurable we get that

$$\|\varphi_\#\mu_0 - \varphi_\#\mu_1\|_{\mathrm{TV}} \leq \|\mu_0 - \mu_1\|_{\mathrm{TV}},$$

with equality if φ is injective.

Proof. We divide the proof into two parts.

(a) Note that for any $f : \mathsf{F} \to \mathbb{R}$ such that $\|f\|_\infty \leq 1$ we have $\|\mathrm{K}f\|_\infty \leq 1$. Using this result we get

$$\begin{aligned}\|\mu_0\mathrm{K} - \mu_1\mathrm{K}\|_{\mathrm{TV}} &= \sup\{\textstyle\int_{\mathsf{F}} f(y)\mathrm{d}(\mu_0\mathrm{K})(y) - \int_{\mathsf{F}} f(y)\mathrm{d}(\mu_1\mathrm{K})(y) \ : \ \|f\|_\infty \leq 1\} \\ &= \sup\{\textstyle\int_{\mathsf{E}} \mathrm{K}f(x)\mathrm{d}\mu_0(x) - \int_{\mathsf{E}} \mathrm{K}f(x)\mathrm{d}\mu_0(x) \ : \ \|f\|_\infty \leq 1\} \leq \|\mu_0 - \mu_1\|_{\mathrm{TV}}.\end{aligned}$$

(b) We have

$$\begin{aligned}\|\varphi_\#\mu_0 - \varphi_\#\mu_1\|_{\mathrm{TV}} &= \sup\{\textstyle\int_{\mathsf{E}} f(\varphi(x))\mathrm{d}\mu_0(x) - \int_{\mathsf{E}} f(\varphi(x))\mathrm{d}\mu_1(x) \ : \ \|f\|_\infty \leq 1\} \\ &\leq \sup\{\textstyle\int_{\mathsf{E}} f(x)\mathrm{d}\mu_0(x) - \int_{\mathsf{E}} f(x)\mathrm{d}\mu_1(x) \ : \ \|f\|_\infty \leq 1\} \leq \|\mu_0 - \mu_1\|_{\mathrm{TV}}.\end{aligned}$$

If φ is injective then there exists $\varphi^{-1} : \mathsf{F} \to \mathsf{F}$ (measurable) such that $\varphi^{-1} \circ \varphi = \mathrm{Id}$. Therefore, for any $f : \mathsf{E} \to \mathbb{R}$ with $\|f\|_\infty \leq 1$ we have $f = (f \circ \varphi^{-1}) \circ \varphi$ and $\|f \circ \varphi^{-1}\|_\infty \leq 1$. Hence we have

$$\begin{aligned}\|\mu_0 - \mu_1\|_{\mathrm{TV}} &= \sup\{\textstyle\int_{\mathsf{E}} f(x)\mathrm{d}\mu_0(x) - \int_{\mathsf{E}} f(x)\mathrm{d}\mu_1(x) \ : \ \|f\|_\infty \leq 1\} \\ &\leq \sup\{\textstyle\int_{\mathsf{E}} f(\varphi(x))\mathrm{d}\mu_0(x) - \int_{\mathsf{E}} f(\varphi(x))\mathrm{d}\mu_1(x) \ : \ \|f\|_\infty \leq 1\} \leq \|\varphi_\#\mu_0 - \varphi_\#\mu_1\|_{\mathrm{TV}},\end{aligned}$$

which concludes the proof.

□

We will also make use of the following inequality.

Lemma 18. *Let $\varepsilon > 0$, $x, y \in \mathbb{R}^d$, $t > 2/\varepsilon$ and $\varphi : [0,1] \to \mathbb{R}$ such that for any $s \in [0,1]$, $\varphi(s) = \exp[-\|x - sy\|^2/(4t)]$. Then $\varphi \in \mathrm{C}^1([0,1], \mathbb{R})$ and we have for any $s \in [0,1]$*

$$|\varphi'(s)| \leq 2(1 + \varepsilon^{-1})(1 + \|x\|) \exp[-\|x\|^2/(8t)] \exp[\varepsilon \|y\|^2]/t.$$

Proof. Let $s \in [0,1]$, we have

$$\varphi'(s) = (\langle x, y \rangle - s\|y\|^2) \exp[-\|x - sy\|^2/(4t)]/(2t).$$

Using the Cauchy-Schwarz inequality and that for any $a, b \in \mathbb{R}^d$, $-\|a+b\|^2 \leq -\|a\|^2/2 + \|b\|^2$ we get

$$|\varphi'(s)| \leq (\|x\| \|y\| + \|y\|^2) \exp[-\|x\|^2/(8t) + \|y\|^2/(4t)]/(2t). \tag{31}$$

In addition, we have

$$\|y\| \exp[\|y\|^2/(4t)] \leq \|y\| \exp[\varepsilon \|y\|^2/2] \leq (1 + \|y\|^2) \exp[\varepsilon \|y\|^2/2] \leq 2(1 + \varepsilon^{-1}) \exp[\varepsilon \|y\|^2]. \tag{32}$$

Finally we also have $\|y\|^2 \exp[\|y\|^2/(4t)] \leq (1 + \varepsilon^{-1}) \exp[\varepsilon \|y\|^2]$. Combining this result, (31) and (32) concludes the proof. □

Finally we show the following lemma which is a straightforward consequence of Girsanov's theorem (Liptser and Shiryaev, 2001, Theorem 7.7). A similar version of this lemma can be found in the proof of (Durmus and Moulines, 2017, Proposition 2) and in (Laumont et al., 2021, Lemma 26) (version where the dependence of the drift in $w \in \mathrm{C}([0,T], \mathbb{R}^d)$ is replaced by a (simpler) dependence in $x \in \mathbb{R}^d$). We refer to (Liptser and Shiryaev, 2001, Section 4) for the definitions of semi-group, non-anticipative processes and diffusion type processes.

Lemma 19. *Let $T > 0$, $b_1, b_2 : [0, +\infty) \times \mathrm{C}([0,T], \mathbb{R}^d) \to \mathbb{R}^d$ measurable such that for any $i \in \{1,2\}$ and $x \in \mathbb{R}^d$, $\mathrm{d}\mathbf{X}_t^{(i)} = b_i(t, (\mathbf{X}_s^{(i)})_{s \in [0,T]})\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t$ admits a unique strong solution with $\mathbf{X}_0^{(i)} = x$ and $(b_i(t, (\mathbf{X}_s^{(i)}))_{t \in [0,T]}$ is non-anticipative, with Markov semi-group $(\mathrm{P}_t^{(i)})_{t \geq 0}$. In addition, assume that for any $x \in \mathbb{R}^d$ and $i \in \{1,2\}$, $\mathbb{P}(\int_0^T \{\|b_i(t, (\mathbf{X}_s^{(i)})_{s \in [0,T]})\|^2 + \|b_i(t, (\mathbf{B}_s)_{s \in [0,T]})\|^2\}\mathrm{d}t < +\infty) = 1$. Then for any $x \in \mathbb{R}^d$ we have*

$$\|\delta_x \mathrm{P}_T^{(1)} - \delta_x \mathrm{P}_T^{(2)}\|_{\mathrm{TV}}^2 \leq (1/2) \int_0^T \mathbb{E}[\|b_1(t, (\mathbf{X}_s^{(1)})_{s \in [0,T]}) - b_2(t, (\mathbf{X}_s^{(1)})_{s \in [0,T]})\|^2]\mathrm{d}t.$$

Proof. Let $T > 0$ and $x \in \mathbb{R}^d$. For any $i \in \{1,2\}$, denote $\mu_{(i)}^x$ the distribution of $(\mathbf{X}_t^{(i)})_{t \in [0,T]}$ on the Wiener space $(\mathcal{C}, \mathcal{B}(\mathcal{C}))$ with $\mathbf{X}_0^{(i)} = x$. Similarly denote μ_B^x the distribution of $(\mathbf{B}_t)_{t \in [0,T]}$ with $\mathbf{B}_0 = x$, where we recall that $(\mathbf{B}_t)_{t \in [0,T]}$ is a d-dimensional Brownian motion. Using Pinsker's inequality (Bakry et al., 2014, Equation 5.2.2) and the transfer theorem (Kullback, 1997, Theorem 4.1) we get that

$$\|\delta_x \mathrm{P}_T^{(1)} - \delta_x \mathrm{P}_T^{(2)}\|_{\mathrm{TV}}^2 \leq 2\,\mathrm{KL}(\mu_{(1)} | \mu_{(2)}).$$

Since for any $i \in \{1,2\}$, $\mathbb{P}(\int_0^T \{\|b_i(t, (\mathbf{X}_s^{(i)})_{s \in [0,T]})\|^2 + \|b_i(t, (\mathbf{B}_s)_{s \in [0,T]})\|^2\}\mathrm{d}t < +\infty) = 1$ and the processes $(\mathbf{X}_t^{(i)})_{t \in [0,T]}$ are of diffusion type for $i \in \{1,2\}$ we can apply Girsanov's theorem (Liptser and Shiryaev, 2001, Theorem 7.7) and μ_B-almost surely for any $w \in \mathrm{C}([0,T], \mathbb{R})$ we get

$$\begin{aligned}
&(\mathrm{d}\mu_{(1)}^x / \mathrm{d}\mu_B^x)((w_t)_{t \in [0,T]}) \\
&\quad = \exp[(1/2) \int_0^T \langle b_1(t, (w_s)_{s \in [0,T]}), \mathrm{d}w_t \rangle - (1/4) \int_0^T \|b_1(t, (w_s)_{s \in [0,T]})\|^2 \mathrm{d}t] \\
&(\mathrm{d}\mu_B^x / \mathrm{d}\mu_{(2)}^x)((w_t)_{t \in [0,T]}) \\
&\quad = \exp[-(1/2) \int_0^T \langle b_2(t, (w_s)_{s \in [0,T]})), \mathrm{d}w_t \rangle + (1/4) \int_0^T \|b_2(t, (w_s)_{s \in [0,T]}))\|^2 \mathrm{d}t].
\end{aligned}$$

Hence, we obtain that

$$\mathrm{KL}(\mu_{(1)}^x | \mu_{(2)}^x) = \mathbb{E}[\log((\mathrm{d}\mu_{(1)}^x / \mathrm{d}\mu_{(2)}^x)((\mathbf{X}_t^{(1)})_{t \in [0,T]}))]$$

$$= (1/4) \int_0^T \mathbb{E}[\|b_1(t, (\mathbf{X}_s^{(1)})_{s\in[0,T]}) - b_2(t, (\mathbf{X}_s^{(1)})_{s\in[0,T]})\|^2]\mathrm{d}t$$

which concludes the proof. □

We study distributions satisfying some curvature assumption and show that they are sub-Gaussian. More precisely, we show the following proposition.

Lemma 20. *Let $q \in \mathrm{C}^1(\mathbb{R}^d, (0, +\infty))$ and $\mathtt{m} > 0$ and $c \geq 0$ such that for any $x \in \mathbb{R}^d$ we have $\langle \nabla \log q(x), x\rangle \leq -\mathtt{m} \|x\|^2 + c\|x\|$. Then for any $\varepsilon \in [0, \mathtt{m}/2)$ we have*

$$\int_{\mathbb{R}^d} \exp[\varepsilon \|x\|^2] q(x)\mathrm{d}x < +\infty.$$

Proof. For any $x \in \mathbb{R}^d$ we have

$$\begin{aligned}\log q(x) &= \log q(0) + \int_0^1 \langle \nabla \log q(tx), x\rangle \mathrm{d}t \\ &\leq \log q(0) - \mathtt{m} \int_0^1 t \|x\|^2 \mathrm{d}t + c\|x\| \leq \log q(0) + c\|x\| - \mathtt{m}\|x\|^2.\end{aligned}$$

which concludes the proof. □

Finally, we will use the following basic lemma.

Lemma 21. *Let $\mu \in \mathscr{P}(\mathbb{R}^d)$, $\alpha_1 \in \mathbb{R}$, $\beta_1 > 0$ and $(\mathbf{X}_t)_{t\geq 0}$ such that $\mathbf{X}_0$ has distribution μ and*

$$\mathrm{d}\mathbf{X}_t = \alpha_1 \mathbf{X}_t \mathrm{d}t + \beta_1^{1/2} \mathrm{d}\mathbf{B}_t,$$

where $(\mathbf{B}_t)_{t\geq 0}$ is a Brownian motion. Then for any $\alpha_2 \in \mathbb{R}$ and $\beta_2 > 0$ we have that $(\mathbf{Y}_t)_{t\geq 0}$ given for any $t \geq 0$ by $\mathbf{Y}_t = \alpha_2 \mathbf{X}_{\beta_2 t}$ satisfies

$$\mathrm{d}\mathbf{Y}_t = \beta_2 \alpha_1 \mathbf{Y}_t \mathrm{d}t + \alpha_2 (\beta_2 \beta_1)^{1/2} \mathrm{d}\tilde{\mathbf{B}}_t,$$

where $(\tilde{\mathbf{B}}_t)_{t\geq 0}$ is a Brownian motion, and $\mathbf{Y}_0$ has distribution $(\tau_{\alpha_2})_{\#}\mu$, where for any $x \in \mathbb{R}^d$, $\tau_{\alpha_2}(x) = \alpha_2 x$.

Proof. Let $t \geq 0$. Using the change of variable $u \mapsto \beta_2 u$ the following equalities hold in distribution

$$\begin{aligned}\mathbf{Y}_t &= \alpha_2 \alpha_1 \int_0^{\beta_2 t} \mathbf{X}_s \mathrm{d}s + \alpha_2 \beta_1^{1/2} \mathbf{B}_{\beta_2 t} \\ &= \beta_2 \alpha_2 \alpha_1 \int_0^t \mathbf{X}_{\beta_2 s} \mathrm{d}s + \alpha_2 (\beta_1 \beta_2)^{1/2} \mathbf{B}_t = \beta_2 \alpha_1 \int_0^t \mathbf{Y}_s \mathrm{d}s + \alpha_2 (\beta_1 \beta_2)^{1/2} \mathbf{B}_t,\end{aligned}$$

which concludes the proof. □

We now turn to the proof of Theorem 1

Proof. Let $\alpha \geq 0$. For any $k \in \{1, \dots, N\}$, denote R_k the Markov kernel such that for any $x \in \mathbb{R}^d$, $\mathsf{A} \in \mathcal{B}(\mathbb{R}^d)$ and $k \in \{0, \dots, N-1\}$ we have

$$\mathrm{R}_{k+1}(x, \mathsf{A}) = (4\pi\gamma_{k+1})^{-1/2} \int_{\mathsf{A}} \exp[-\|\tilde{x} - \mathcal{T}_{k+1}(x)\|^2 / (4\gamma_{k+1})] \mathrm{d}\tilde{x},$$

where for any $x \in \mathbb{R}^d$, $\mathcal{T}_{k+1}(x) = x + \gamma_{k+1}\{\alpha x + 2 s_\theta(t_k, x)\}$, where $t_k = \sum_{\ell=0}^{k-1} \gamma_\ell$. Define for any $k_0, k_1 \in \{1, \dots, N\}$ with $k_1 \geq k_0$ $\mathrm{Q}_{k_0, k_1} = \prod_{\ell=k_0}^{k_1} \mathrm{R}_\ell$. Finally, for ease of notation, we also define for any $k \in \{1, \dots, N\}$, $\mathrm{Q}_k = \mathrm{Q}_{1,k}$. Note that for any $k \in \{1, \dots, N\}$, Y_k has distribution $\pi_\infty \mathrm{Q}_k$, where $\pi_\infty \in \mathscr{P}(\mathbb{R}^d)$ with density w.r.t. the Lebesgue measure p_{data}. Let $\mathbb{P} \in \mathscr{P}(\mathcal{C})$ be the probability measure associated with the diffusion

$$\mathrm{d}\mathbf{X}_t = -\alpha \mathbf{X}_t \mathrm{d}t + \sqrt{2} \mathrm{d}\mathbf{B}_t, \quad \mathbf{X}_0 \sim \pi_0,$$

where $\pi_0 \in \mathscr{P}(\mathbb{R}^d)$ admits a density w.r.t. the Lebesgue measure given by p_{data}. First note that using that $\mathbb{P}_0 = \pi_0$ we have for any $\mathsf{A} \in \mathcal{B}(\mathbb{R}^d)$

$$\pi_0 \mathbb{P}_{T|0}(\mathbb{P}^R)_{T|0}(\mathsf{A}) = \mathbb{P}_T(\mathbb{P}^R)_{T|0}(\mathsf{A}) = (\mathbb{P}^R)_0(\mathbb{P}^R)_{T|0}(\mathsf{A}) = (\mathbb{P}^R)_T(\mathsf{A}) = \pi_0(\mathsf{A}).$$

Hence $\pi_0 = \pi_0 \mathbb{P}_{T|0}(\mathbb{P}^R)_{T|0}$. Using this result and Lemma 17, we have

$$\|\pi_0 - \pi_\infty \mathrm{Q}_N\|_{\mathrm{TV}} = \|\pi_0 \mathbb{P}_{T|0}(\mathbb{P}^R)_{T|0} - \pi_\infty \mathrm{Q}_N\|_{\mathrm{TV}}$$

$$\leq \|\pi_0 \mathbb{P}_{T|0}(\mathbb{P}^R)_{T|0} - \pi_\infty(\mathbb{P}^R)_{T|0}\|_{\mathrm{TV}} + \|\pi_\infty(\mathbb{P}^R)_{T|0} - \pi_\infty \mathrm{Q}_N\|_{\mathrm{TV}}$$
$$\leq \|\pi_0 \mathbb{P}_{T|0} - \pi_\infty\|_{\mathrm{TV}} + \|\pi_\infty(\mathbb{P}^R)_{T|0} - \pi_\infty \mathrm{Q}_N\|_{\mathrm{TV}}.$$

Note that $\mathcal{L}(X_0) = \mathcal{L}(Y_N) = \pi_\infty \mathrm{Q}_N$ and therefore

$$\|\mathcal{L}(X_0) - \pi_0\|_{\mathrm{TV}} \leq \|\pi_0 \mathbb{P}_{T|0} - \pi_\infty\|_{\mathrm{TV}} + \|\pi_\infty(\mathbb{P}^R)_{T|0} - \pi_\infty \mathrm{Q}_N\|_{\mathrm{TV}}.$$

We now bound each one of these terms.

(a) First, assume that $\alpha > 0$. Let $T_\alpha = \alpha T$ and $\tilde{\mathbb{P}} \in \mathscr{P}(\mathrm{C}([0, T_\alpha], \mathbb{R}^d))$ be associated with $(\mathbf{Z}_t)_{t\in[0,T_\alpha]}$ the classical Ornstein-Ulhenbeck process with $\mathbf{Z}_0 \sim (\tau_\alpha)_\# \pi_0$, where for any $x \in \mathbb{R}^d$ we have $\tau_\alpha(x) = \alpha^{1/2} x$, satisfying the following SDE: $\mathrm{d}\mathbf{Z}_t = -\mathbf{Z}_t \mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t$. We denote $\pi_0^\alpha = (\tau_\alpha)_\# \pi_0$, $\mu = (\tau_\alpha)_\# \pi_\infty$. Note that since p_{prior} is the Gaussian density with zero mean and covariance matrix $(1/\alpha)\,\mathrm{Id}$, μ is the Gaussian distribution with zero mean and identity covariance matrix.

First, using (Bakry et al., 2014, Proposition 4.1.1, Proposition 4.3.1, Theorem 4.2.5), we get that for any $t \in [0, T_\alpha]$, $f \in \mathrm{L}^1(\mu)$ and $x \in \mathbb{R}^d$

$$\int_{\mathbb{R}^d} (\tilde{\mathbb{P}}_{t|0} g(x))^2 \mathrm{d}\mu(x) \leq \exp[-2t] \int_{\mathbb{R}^d} g^2(x) \mathrm{d}\mu(x), \quad \text{with } g(x) = f(x) - \int_{\mathbb{R}^d} f(\tilde{x}) \mathrm{d}\mu(\tilde{x}). \tag{33}$$

Recall that $(\mathbf{X}_t)_{t\geq 0}$ satisfies $\mathrm{d}\mathbf{X}_t = -\alpha \mathbf{X}_t + \mathrm{d}\mathbf{B}_t$. Using Lemma 21 we have that for any $t \in [0, T]$, $\mathbf{Z}_t$ and $\alpha^{1/2} \mathbf{X}_{\alpha^{-1}t}$ have the same distribution. Hence for any $t \in [0, T]$ we have $\mathbb{P}_t = (\tau_\alpha^{-1})_\# \tilde{\mathbb{P}}_{\alpha t}$. Therefore, using that $(\tau_\alpha)_\# \pi_\infty = \mu$, that $\tilde{\mathbb{P}}$ is Markov and Lemma 17, we get that

$$\|\pi_0 \mathbb{P}_{t|0} - \pi_\infty\|_{\mathrm{TV}} = \|\mathbb{P}_t - \pi_\infty\|_{\mathrm{TV}} = \|(\tau_\alpha)_\# \mathbb{P}_t - (\tau_\alpha)_\# \pi_\infty\|_{\mathrm{TV}}$$
$$= \|\tilde{\mathbb{P}}_{\alpha t} - \mu\|_{\mathrm{TV}} = \|\tilde{\mathbb{P}}_{\alpha t_0} \tilde{\mathbb{P}}_{\alpha(t-t_0)|0} - \mu\|_{\mathrm{TV}}.$$

Finally, note that we have for any $t \geq t_0 \in [0, T]$ and $x \in \mathbb{R}^d$

$$(\mathrm{d}(\tilde{\mathbb{P}}_{\alpha t_0} \tilde{\mathbb{P}}_{\alpha(t-t_0)|0})/\mathrm{d}\mu)(x) = \tilde{\mathbb{P}}_{\alpha(t-t_0)|0} f(x), \quad \text{with } f(x) = (\mathrm{d}\tilde{\mathbb{P}}_{\alpha t_0}/\mathrm{d}\mu)(x). \tag{34}$$

Let $g = f - 1$. Using (34), (33) and that $(\tau_\alpha)_\# \pi_\infty = \mu$, we get that for any $t \geq t_0$ with $t \in [0, T]$

$$\|\pi_0 \mathbb{P}_{t|0} - \pi_\infty\|_{\mathrm{TV}} \leq \|\tilde{\mathbb{P}}_{\alpha t_0} \tilde{\mathbb{P}}_{\alpha(t-t_0)|0} - \mu\|_{\mathrm{TV}} \tag{35}$$
$$\leq \int_{\mathbb{R}^d} |\tilde{\mathbb{P}}_{\alpha(t-t_0)|0} f(x) - 1| \mathrm{d}\mu(x)$$
$$\leq (\int_{\mathbb{R}^d} (\tilde{\mathbb{P}}_{\alpha(t-t_0)|0} g(x))^2 \mathrm{d}\mu(x))^{1/2}$$
$$\leq \exp[-\alpha(t-t_0)] (\int_{\mathbb{R}^d} g^2(x) \mathrm{d}\mu(x))^{1/2}$$
$$\leq \exp[-\alpha(t-t_0)] (\int_{\mathbb{R}^d} g^2(\alpha^{1/2} x) \mathrm{d}\pi_\infty(x))^{1/2}.$$

In addition, we have for any $\varphi \in \mathrm{C}_c(\mathbb{R}^d, \mathbb{R})$

$$\int_{\mathbb{R}^d} \varphi(x) f(\alpha^{1/2} x) \mathrm{d}\pi_\infty(x) = \int_{\mathbb{R}^d} \varphi(\alpha^{-1/2} x) f(x) \mathrm{d}\mu(x)$$
$$= \int_{\mathbb{R}^d} \varphi(\alpha^{-1/2} x) \mathrm{d}\tilde{\mathbb{P}}_{\alpha t_0}(x) = \int_{\mathbb{R}^d} \varphi(x) \mathrm{d}\mathbb{P}_{t_0}(x).$$

Hence, for any $x \in \mathbb{R}^d$, $g(\alpha^{1/2} x) = (\mathrm{d}\mathbb{P}_{t_0}/\mathrm{d}\pi_\infty)(x) - 1$. Combining this result and (35) we get that for any $t \geq t_0$ with $t \in [0, T]$

$$\|\pi_0 \mathbb{P}_{t|0} - \pi_\infty\|_{\mathrm{TV}} \leq \sqrt{2} \exp[-\alpha(t-t_0)] \left(1 + \int_{\mathbb{R}^d} (\mathrm{d}\mathbb{P}_{t_0}/\mathrm{d}\pi_\infty)(x)^2 \mathrm{d}\pi_\infty(x)\right)^{1/2}. \tag{36}$$

Let $t_0 \in [0, T]$. We now derive an upper bound for $\int_{\mathbb{R}^d} (\mathrm{d}\mathbb{P}_{t_0}/\mathrm{d}\pi_\infty)(x)^2 \mathrm{d}\pi_\infty(x)$. We recall that $\mathbb{P}_{t_0}$ and π_∞ admit density w.r.t. the Lebesgue measure denoted p_{t_0} and p_∞ such that for any $x \in \mathbb{R}^d$

$$p_{t_0}(x) = \int_{\mathbb{R}^d} \mathrm{G}_{t_0}(x, \tilde{x}) \mathrm{d}\pi_0(\tilde{x}), \quad p_\infty(x) = (2\pi/\alpha)^{-d/2} \exp[-\alpha \|x\|^2 / 2],$$

where for any $x, \tilde{x} \in \mathbb{R}^d$

$$\mathrm{G}_{t_0}(x, \tilde{x}) = (2\pi\sigma_{t_0}^2)^{-d/2} \exp[-\|x - m_{t_0}(\tilde{x})\|^2 / (2\sigma_{t_0}^2)],$$
$$\sigma_{t_0}^2 = (1 - \exp[-2\alpha t_0])/\alpha, \qquad m_{t_0}(\tilde{x}) = \exp[-\alpha t_0] \tilde{x}.$$

Combining this result and Jensen's inequality we get

$$\int_{\mathbb{R}^d} p_{t_0}^2(x) p_\infty^{-1}(x) \mathrm{d}x \leq \alpha^{-d/2} (2\pi)^{-d/2} \sigma_{t_0}^{-2d} \int_{\mathbb{R}^d} \exp[-\left\|x - m_{t_0}(\tilde{x})\right\|^2 / \sigma_{t_0}^2 + \alpha \left\|x\right\|^2 / 2] \mathrm{d}x \mathrm{d}\pi_0(\tilde{x}). \tag{37}$$

For any $x, \tilde{x} \in \mathbb{R}^d$ we have

$$\left\|x - m_{t_0}(\tilde{x})\right\|^2 / \sigma_{t_0}^2 - \alpha \left\|x\right\|^2 / 2 = \left\|x - m_{t_0}(\tilde{x})(2\tilde{\sigma}_{t_0}^2 / \sigma_{t_0}^2)\right\|^2 / (2\tilde{\sigma}_{t_0}^2) - \left\|\tilde{x}\right\|^2 \phi(\alpha, t_0) / \sigma_{t_0}^2,$$

with $\tilde{\sigma}_{t_0}^2 = (\sigma_{t_0}^2/2)(1 - \alpha\sigma_{t_0}^2/2)^{-1}$ and $\phi(\alpha, t_0) = \alpha\sigma_{t_0}^2(1 - \sigma_{t_0}^2\alpha)/(2 - \sigma_{t_0}^2\alpha)$. Using this result, we get that

$$\int_{\mathbb{R}^d} \exp[-\|x - m_{t_0}(\tilde{x})\|^2/\sigma_{t_0}^2 + \alpha\|x\|^2/2] \mathrm{d}x \mathrm{d}\pi_0(\tilde{x}) \leq (2\pi\tilde{\sigma}_{t_0}^2)^{d/2} \int_{\mathbb{R}^d} \exp[\phi(\alpha, t_0)\|\tilde{x}\|^2] \mathrm{d}\pi_0(\tilde{x}),$$

Let $\varepsilon = \mathtt{m}/4$ and $t_0 \geq 0$ such that $\phi(\alpha, t_0) \leq \varepsilon$. Using Lemma 20, we get that

$$\int_{\mathbb{R}^d} \exp[-\left\|x - m_{t_0}(\tilde{x})\right\|^2 / \sigma_{t_0}^2 + \alpha \left\|x\right\|^2 / 2] \mathrm{d}x \mathrm{d}\pi_0(\tilde{x}) \leq (2\pi\tilde{\sigma}_{t_0}^2)^{d/2} \int_{\mathbb{R}^d} \exp[\varepsilon \left\|\tilde{x}\right\|^2] \mathrm{d}\pi_0(\tilde{x}).$$

Combining this result, the fact that $\sigma_{t_0}^2 \leq \alpha^{-1}$, (37) and that for any $t \geq 0$, $(1 - \mathrm{e}^{-t})^{-1} \leq 1 + 1/t$, we obtain

$$\begin{aligned}
\int_{\mathbb{R}^d} p_{t_0}^2(x) p_\infty^{-1}(x) \mathrm{d}x &\leq (\alpha^{-1}\tilde{\sigma}_{t_0}^2 \sigma_{t_0}^{-4})^{d/2} \int_{\mathbb{R}^d} \exp[\varepsilon \left\|\tilde{x}\right\|^2] \mathrm{d}\pi_0(\tilde{x}) \\
&\leq (1 - \exp[-2\alpha t_0])^{-d/2} \int_{\mathbb{R}^d} \exp[\varepsilon \left\|\tilde{x}\right\|^2] \mathrm{d}\pi_0(\tilde{x}) \\
&\leq (1 + 1/(2\alpha t_0))^{d/2} \int_{\mathbb{R}^d} \exp[\varepsilon \left\|\tilde{x}\right\|^2] \mathrm{d}\pi_0(\tilde{x}).
\end{aligned}$$

Combining this result and (36), we get that for any $t > t_0$

$$\left\|\pi_0 \mathbb{P}_{t|0} - \pi_\infty\right\|_{\mathrm{TV}} \leq C_1^a \exp[-\alpha t],$$

with

$$C_1^a = \sqrt{2}(1 + 1/(2\alpha t_0))^{d/2} (1 + (\textstyle\int_{\mathbb{R}^d} \exp[\varepsilon \|\tilde{x}\|^2] \mathrm{d}\pi_0(\tilde{x}))^{1/2}) \exp[\alpha t_0].$$

For $t \leq t_0$, using that $\left\|\pi_0 \mathbb{P}_{t|0} - \pi_\infty\right\|_{\mathrm{TV}} \leq 1$ we have

$$\left\|\pi_0 \mathbb{P}_{t|0} - \pi_\infty\right\|_{\mathrm{TV}} \leq C_1^b \exp[-\alpha t], \qquad \text{with } C_1^b = \exp[\alpha t_0].$$

Let $C_1 = C_1^a + C_1^b$ and we have that for any $t \in [0, T]$

$$\left\|\pi_0 \mathbb{P}_{t|0} - \pi_\infty\right\|_{\mathrm{TV}} \leq C_1 \exp[-\alpha t]. \tag{38}$$

(b) Second assume that $\alpha = 0$.

$$\left\|\pi_0 \mathbb{P}_{T|0} - \pi_\infty\right\|_{\mathrm{TV}} \leq \int_{\mathbb{R}^d} \int_{\mathbb{R}^d} (4\pi T)^{-d/2} |\exp[-\left\|x - \tilde{x}\right\|^2 / (4T)] - \exp[-\left\|x\right\|^2 / (4T)]| \mathrm{d}x \mathrm{d}\pi_0(\tilde{x}).$$

For any $x, \tilde{x} \in \mathbb{R}^d$, let $\varphi \in \mathrm{C}^1([0, 1], \mathbb{R})$ with for any $s \in [0, 1]$, $\varphi(s) = \exp[-\left\|x - s\tilde{x}\right\|^2 / (4T)]$. First, assume that $T \geq 2/\varepsilon$. Using Lemma 18, we get that for any $s \in [0, 1]$

$$|\varphi'(s)| \leq (1 + \varepsilon^{-1})(1 + \left\|x\right\|) \exp[-\left\|x\right\|^2 / (8T)] \exp[\varepsilon \left\|y\right\|^2] / T.$$

Using this result we get that

$$\begin{aligned}
\left\|\pi_0 \mathbb{P}_{T|0} - \pi_\infty\right\|_{\mathrm{TV}} &\leq \textstyle\int_{\mathbb{R}^d} \int_{\mathbb{R}^d} (4\pi T)^{-d/2} |\exp[-\left\|x - \tilde{x}\right\|^2 / (4T)] - \exp[-\left\|x\right\|^2 / (4T)]| \mathrm{d}x \mathrm{d}\pi_0(\tilde{x}) \\
&\leq \textstyle\int_{\mathbb{R}^d} \int_{\mathbb{R}^d} (4\pi T)^{-d/2} (1 + \varepsilon^{-1})(1 + \left\|x\right\|) \exp[-\left\|x\right\|^2 / (8T)] \exp[\varepsilon \left\|\tilde{x}\right\|^2] / T \mathrm{d}x \mathrm{d}\pi_0(\tilde{x}) \\
&\leq 2^{d/2}(1 + \varepsilon^{-1}) \textstyle\int_{\mathbb{R}^d} (8\pi T)^{-d/2} (1 + \left\|x\right\|) \exp[-\left\|x\right\|^2 / (8T)] \mathrm{d}x \int_{\mathbb{R}^d} \exp[\varepsilon \left\|\tilde{x}\right\|^2] / T \mathrm{d}\pi_0(\tilde{x}) \\
&\leq 2^{d/2}(1 + \varepsilon^{-1})(1 + 2\sqrt{2} d^{1/2} T^{1/2}) \textstyle\int_{\mathbb{R}^d} \exp[\varepsilon \left\|\tilde{x}\right\|^2] / T \mathrm{d}\pi_0(\tilde{x}).
\end{aligned}$$

In addition, if $T \leq 2/\varepsilon$ then

$$\left\|\pi_0 \mathbb{P}_{T|0} - \pi_\infty\right\|_{\mathrm{TV}} \leq (\varepsilon/2 + (\varepsilon/2)^{1/2})^{-1} (T^{-1} + T^{-1/2}).$$

Hence, we get that there exists $C_2 \geq 0$ such that

$$\left\|\pi_0 \mathbb{P}_{T|0} - \pi_\infty\right\|_{\mathrm{TV}} \leq C_2 (T^{-1} + T^{-1/2}), \tag{39}$$

with

$$C_2 = (\varepsilon/2 + (\varepsilon/2)^{1/2})^{-1} + 2^{d/2}(1 + \varepsilon^{-1})(1 + 2\sqrt{2} d^{1/2}) \textstyle\int_{\mathbb{R}^d} \exp[\varepsilon \left\|\tilde{x}\right\|^2] \mathrm{d}\pi_0(\tilde{x}).$$

(c) Recall that $\mathbb{P}^R$ is associated with the diffusion $(\mathbf{Y}_t)_{t\geq 0}$ such that for any $t \in [0,T]$ and $x \in \mathbb{R}^d$

$$\mathrm{d}\mathbf{Y}_t = b_1(t, \mathbf{Y}_t)\mathrm{d}t + \sqrt{2}\mathbf{B}_t, \quad b_1(t,x) = \alpha x + 2\nabla \log p_{T-t}(x).$$

Similarly, for any $k \in \{1, \dots, N\}$ we have $\mathrm{Q}_k = \mathbb{Q}_{t_k}$ where $\mathbb{Q}$ is associated with the diffusion $(\bar{\mathbf{Y}}_t)_{t\in[0,T]}$ such that for any $(w_t)_{t\in[0,T]} \in \mathrm{C}([0,T], \mathbb{R}^d)$ we have

$$\begin{aligned}
&\mathrm{d}\bar{\mathbf{Y}}_t = b_2(t, (\bar{\mathbf{Y}}_s)_{s\in[0,T]})\mathrm{d}t + \sqrt{2}\mathbf{B}_t, \\
&b_2(t, (w_t)_{t\in[0,T]}) = \textstyle\sum_{k=0}^{N-1} \mathbb{1}_{[t_k, t_{k+1})}(t) \left\{2\alpha w_{t_k} + s_\theta(t_k, w_{t_k})\right\}
\end{aligned}$$

where for any $k \in \{0, \dots, N\}$, $t_k = \sum_{\ell=0}^{k-1} \gamma_{\ell+1}$. Recall that for any $i \in \{1,2,3\}$ there exist $A_i \geq 0$ and $\alpha_i \in \mathbb{N}$ such that for any $x_0 \in \mathbb{R}^d$

$$\|\nabla^i \log p_0(x)\| \leq A_i(1 + \|x_0\|^{\alpha_i}),$$

with $\alpha_1 = 1$. Using this result and Theorem 16 we get that for any $i \in \{1,2,3\}$ there exist $B_i \geq 0$ and $\beta_i \in \mathbb{N}$ with $\beta_1 = 1$ such that for any $x_t \in \mathbb{R}^d$ and $t \in [0,T]$

$$\|\nabla^i \log p_t(x_t)\| \leq B_i(1 + \|x_t\|^{\beta_i}). \tag{40}$$

In addition, for any $t \in [0,T]$ and $x \in \mathbb{R}^d$ we have

$$\partial_t p_t(x) = -\mathrm{div}(bp_t)(x) + \Delta p_t(x),$$

with $b(x) = -\alpha x$. Therefore, since $\log p \in \mathrm{C}^\infty((0,T] \times \mathbb{R}^d, \mathbb{R})$ we obtain that for any $t \in (0,T]$ and $x_t \in \mathbb{R}^d$

$$\partial_t \log p_t(x_t) = -\mathrm{div}(b \log p_t)(x_t) + \Delta \log p_t(x_t) + \|\nabla \log p_t(x_t)\|^2 .$$

Finally, we get that for any $t \in (0,T]$ and $x_t \in \mathbb{R}^d$

$$\partial_t \nabla \log p_t(x_t) = -\nabla \mathrm{div}(b \log p_t)(x_t) + \nabla \Delta \log p_t(x_t) + \nabla \|\nabla \log p_t\|^2 (x_t).$$

Therefore combining this result and (40) there exist $\tilde{A} \geq 0$ and $\beta \in \mathbb{N}$ such that for any $x_t \in \mathbb{R}^d$ and $t \in (0,T]$, $\|\partial_t \nabla \log p_t(x_t)\| \leq \tilde{A}(1 + \|x_t\|^\beta)$. Hence, for any $t_1, t_2 \in [0,T]$ and $x \in \mathbb{R}^d$

$$\|\nabla \log p_{t_2}(x) - \nabla \log p_{t_1}(x)\| \leq \tilde{A}\,|t_2 - t_1|\,(1 + \|x\|^\beta). \tag{41}$$

In addition, using (40), we have for any $t \in [0,T]$ and $x_1, x_2 \in \mathbb{R}^d$

$$\begin{aligned}
\|\nabla \log p_t(x_1) - \nabla \log p_t(x_2)\| &\leq \textstyle\int_0^1 \|\nabla^2 \log p_t((1-s)x_1 + sx_2)\|\mathrm{d}s \|x_1 - x_2\| \\
&\leq B_2(1 + \textstyle\int_0^1 \|(1-s)x_1 + sx_2\|^{\beta_2}\,\mathrm{d}s)\|x_1 - x_2\| \\
&\leq B_2(1 + \|x_1\|^{\beta_2} + \|x_2\|^{\beta_2})\|x_1 - x_2\|.
\end{aligned} \tag{42}$$

Since $s_\theta \in \mathrm{C}([0,T] \times \mathbb{R}^d, \mathbb{R}^d)$ and $\nabla \log p \in \mathrm{C}([0,T] \times \mathbb{R}^d, \mathbb{R}^d)$ we have using Lemma 19, (41), (42) and the Cauchy-Schwarz inequality

$$\begin{aligned}
\|\pi_\infty(\mathbb{P}^R)_{T|0} - \pi_\infty \mathrm{Q}_N\|_{\mathrm{TV}}^2 &\leq (1/2)\textstyle\int_0^T \mathbb{E}[\|b_1(t, \mathbf{Y}_t) - b_2(t, (\mathbf{Y}_t)_{t\in[0,T]})\|^2]\mathrm{d}t \\
&\leq 2\textstyle\sum_{k=0}^{N-1} \int_{t_k}^{t_{k+1}} \mathbb{E}[\|\nabla \log p_{T-t}(\mathbf{Y}_t) - s_\theta(\mathbf{Y}_{t_k})\|^2]\mathrm{d}t \\
&\quad + \textstyle\sum_{k=0}^{N-1} \int_{t_k}^{t_{k+1}} \alpha^2 \mathbb{E}[\|\mathbf{Y}_t - \mathbf{Y}_{t_k}\|^2]\mathrm{d}t \\
&\leq 6\textstyle\sum_{k=0}^{N-1} \int_{t_k}^{t_{k+1}} \mathbb{E}[\|\nabla \log p_{T-t}(\mathbf{Y}_t) - \nabla \log p_{T-t}(\mathbf{Y}_{t_k})\|^2]\mathrm{d}t \\
&\quad + 6\textstyle\sum_{k=0}^{N-1} \int_{t_k}^{t_{k+1}} \mathbb{E}[\|\nabla \log p_{T-t}(\mathbf{Y}_{t_k}) - \nabla \log p_{T-t_k}(\mathbf{Y}_{t_k})\|^2]\mathrm{d}t \\
&\quad + 6\textstyle\sum_{k=0}^{N-1} \int_{t_k}^{t_{k+1}} \mathbb{E}[\|\nabla \log p_{T-t_k}(\mathbf{Y}_{t_k}) - s_\theta(t_k, \mathbf{Y}_{t_k})\|^2]\mathrm{d}t \\
&\quad + \textstyle\sum_{k=0}^{N-1} \int_{t_k}^{t_{k+1}} \alpha^2 \mathbb{E}[\|\mathbf{Y}_t - \mathbf{Y}_{t_k}\|^2]\mathrm{d}t \\
&\leq 18\sqrt{2}B_2^2(1 + 2N_T(4\beta_2))^{1/2} \textstyle\sum_{k=0}^{N-1} \int_{t_k}^{t_{k+1}} \mathbb{E}[\|\mathbf{Y}_t - \mathbf{Y}_{t_k}\|^4]^{1/2}\mathrm{d}t \\
&\quad + 12\tilde{A}^2(1 + N_T(2\beta))\textstyle\sum_{k=0}^{N-1} \int_{t_k}^{t_{k+1}} (t - t_k)^2\mathrm{d}t + 6T\mathtt{M}^2
\end{aligned} \tag{43}$$

$$
\begin{aligned}
&+ \sum_{k=0}^{N-1} \int_{t_k}^{t_{k+1}} \alpha^2 \mathbb{E}[\|\mathbf{Y}_t - \mathbf{Y}_{t_k}\|^2] \mathrm{d}t \\
&\leq \{18\sqrt{2} B_2^2 (1 + 2N_T(4\beta_2))^{1/2} + \alpha^2\} \sum_{k=0}^{N-1} \int_{t_k}^{t_{k+1}} \mathbb{E}[\|\mathbf{Y}_t - \mathbf{Y}_{t_k}\|^4]^{1/2} \mathrm{d}t \\
&\quad + 4\tilde{A}^2 (1 + N_T(2\beta)) \sum_{k=0}^{N-1} (t_{k+1} - t_k)^3 + 6T\mathtt{M}^2 \\
&\leq \{18\sqrt{2} B_2^2 (1 + 2N_T(4\beta_2))^{1/2} + \alpha^2\} \sum_{k=0}^{N-1} \int_{t_k}^{t_{k+1}} \mathbb{E}[\|\mathbf{Y}_t - \mathbf{Y}_{t_k}\|^4]^{1/2} \mathrm{d}t \\
&\quad + 4\tilde{A}^2 (1 + N_T(2\beta)) T\bar{\gamma}^2 + 6T\mathtt{M}^2,
\end{aligned}
$$

where for any $\ell \in \mathbb{N}$, $N_T(\ell) = \sup_{t \in [0,T]} \mathbb{E}[\|\mathbf{Y}_t\|^\ell]$. For any $t \in [0,T]$, let $\mathcal{A}_t : \mathrm{C}^2(\mathbb{R}^d) \to \mathrm{C}^2(\mathbb{R}^d, \mathbb{R})$ the generator given for any $t \geq 0$, $\varphi \in \mathrm{C}^2(\mathbb{R}^d, \mathbb{R})$ and $x \in \mathbb{R}^d$ by

$$
\mathcal{A}_t(\varphi)(x) = \langle \alpha x + 2\nabla \log p_{T-t}(x), \nabla \varphi(x) \rangle + \Delta \varphi(x).
$$

For any $\ell \in \mathbb{N}$, let $V_\ell(x) = \|x\|^{2\ell}$. Hence, for any $\ell \in \mathbb{N}$, $x \in \mathbb{R}^d$ and $t \in [0,T]$ we have using (40)

$$
\mathcal{A}_t(V_\ell)(x) = 2\ell\alpha \|x\|^{2\ell} + 2\ell B_1 \|x\|^{2\ell-1} + 2\ell B_1 \|x\|^{2\ell} + 2\ell(2\ell-1) \|x\|^{2(\ell-1)}.
$$

Hence, for any $\ell \in \mathbb{N}$ there exist $\tilde{B}_\ell$ such that $x \in \mathbb{R}^d$ and $t \in [0,T]$

$$
|\mathcal{A}_t(V_\ell)(x)| \leq \tilde{B}_\ell (1 + V_\ell(x)). \tag{44}
$$

For any $\ell \in \mathbb{N}$, $(M_{\ell,t})_{t \in [0,T]} = (V_\ell(\mathbf{Y}_t) - V_\ell(\mathbf{Y}_0) - \int_0^t \mathcal{A}_t(V_\ell)(\mathbf{Y}_s) \mathrm{d}s)_{t \in [0,T]}$ is a local martingale. For any $\ell \in \mathbb{N}$, there exists $(\tau_{\ell,k})_{k \in \mathbb{N}}$ a sequence of stopping times such that $\lim_{k \to +\infty} \tau_{\ell,k} = T$ and $(M_{\ell,t \wedge \tau_{\ell,k}})_{t \in [0,T]}$ is a martingale. Using (44), we have for any $t \in [0,T]$, $\ell \in \mathbb{N}$ and $k \in \mathbb{N}$

$$
\mathbb{E}[V_\ell(\mathbf{Y}_{t \wedge \tau_{\ell,k}})] \leq \mathbb{E}[V_\ell(\mathbf{Y}_0)] + \tilde{B}_\ell \int_0^t (1 + \mathbb{E}[V_\ell(\mathbf{Y}_{s \wedge \tau_{\ell,k}})]) \mathrm{d}s.
$$

Hence, using Grönwall's lemma we get that for any $\ell \in \mathbb{N}$, $\sup_{k \in \mathbb{N}} \mathbb{E}[V_\ell(\mathbf{Y}_{t \wedge \tau_{\ell,k}})] < +\infty$. Therefore for any $\ell \in \mathbb{N}$, $((M_{\ell,t \wedge \tau_k})_{t \in [0,T]})_{k \in \mathbb{N}}$ is uniformly integrable and we have that for any $\ell \in \mathbb{N}$, $(M_{\ell,t})_{t \in [0,T]}$ is a martingale. Therefore we get that for any $t \in [0,T]$, $\ell \in \mathbb{N}$

$$
\mathbb{E}[V_\ell(\mathbf{Y}_t)] \leq \mathbb{E}[V_\ell(\mathbf{Y}_0)] + \tilde{B}_\ell \int_0^t (1 + \mathbb{E}[V_\ell(\mathbf{Y}_s)] \mathrm{d}s).
$$

Using Grönwall's lemma we get that for any $\ell \in \mathbb{N}$ there exist $\tilde{C}_\ell \geq 0$ such that

$$
N_T(\ell) = \sup_{t \in [0,T]} \mathbb{E}[\|\mathbf{Y}_t\|^{2\ell}] \leq \tilde{C}_\ell \exp[\tilde{B}_\ell T]. \tag{45}
$$

We have that for any $s, t \in [0,T]$

$$
\mathbf{Y}_t = \mathbf{Y}_s + \int_s^t \{\alpha \mathbf{Y}_u + 2\nabla \log p_{T-t}(\mathbf{Y}_u)\} \mathrm{d}u + \sqrt{2} \int_s^t \mathrm{d}\mathbf{B}_u.
$$

Using (41) and Cauchy-Schwarz inequality we have for any $s, t \in [0,T]$

$$
\begin{aligned}
\mathbb{E}[\|\mathbf{Y}_t - \mathbf{Y}_s\|^4] &\leq 64(t-s)^3 \int_s^t \{\alpha^4 \mathbb{E}[\|\mathbf{Y}_u\|^4] + 16\mathbb{E}[\|\nabla \log p_{T-t}(\mathbf{Y}_u)\|^4]\} \mathrm{d}u + 48\sqrt{2}(t-s)^2 \\
&\leq 64(t-s)^3 \int_s^t \{\alpha^4 \mathbb{E}[\|\mathbf{Y}_u\|^4] + 128 B_1^4 (1 + \mathbb{E}[\|\mathbf{Y}_u\|^4])\} \mathrm{d}u + 48\sqrt{2}(t-s)^2 \\
&\leq 64(\alpha^4 + 128 B_1^4)(1 + N_T(4))(t-s)^4 + 48\sqrt{2}(t-s)^2. \qquad (46)
\end{aligned}
$$

Combining (45) and (46) in (43) we get that there exist $C_3 \geq 0$ such that

$$
\|\pi_\infty(\mathbb{P}^R)_{T|0} - \pi_\infty \mathrm{Q}_N\|_{\mathrm{TV}}^2 \leq C_3 \exp[C_3 T](\bar{\gamma} + \mathtt{M}^2), \tag{47}
$$

We conclude the proof upon combining (38) and (47) if $\alpha > 0$ and (39) and (47) if $\alpha = 0$. □

C.3 General SGM and links with existing works

In this section we describe a general algorithm for SGM in Appendix C.3.1 and show that the formulation (6) encompasses the ones of (Song et al., 2021; Ho et al., 2020) in Appendix C.3.2.

C.3.1 General SGM algorithm

We first present a general algorithm to compute approximate reverse dynamics, *i.e.* to compute the reverse-time Markov chain associated with the forward process

$$\mathrm{d}\mathbf{X}_t = f_t(\mathbf{X}_t)\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t, \qquad \mathbf{X}_0 \sim p_{\text{data}}. \tag{48}$$

We use the Euler-Maruyama discretization of (48), *i.e.* let $X_0 \sim p_{\text{data}}$ and for any $k \in \{0, \ldots, N-1\}$

$$X_{k+1} = X_k + \gamma_{k+1} f_k(X_k) + \sqrt{2\gamma_{k+1}} Z_{k+1}.$$

In general, we do not have that $p(x_k|x_0)$ is a Gaussian density contrary to Song and Ermon (2019); Ho et al. (2020). However, in this case, we obtain that for any $x \in \mathbb{R}^d$,

$$p_{k+1}(x) = (4\pi\gamma_{k+1})^{-d/2} \int_{\mathbb{R}^d} p_k(\tilde{x}) \exp[-\left\|\mathcal{T}_{k+1}(\tilde{x}) - x\right\|^2 / (4\gamma_{k+1})]\mathrm{d}\tilde{x},$$

with $\mathcal{T}_{k+1}(x) = \tilde{x} + \gamma_{k+1} f_k(\tilde{x})$. Therefore, we get that for any $x \in \mathbb{R}^d$

$$(2\gamma_{k+1} p_{k+1}(x)) \nabla \log p_{k+1}(x) = \int_{\mathbb{R}^d} (\mathcal{T}_{k+1}(\tilde{x}) - x) p_k(\tilde{x}) \exp[-\left\|\mathcal{T}_{k+1}(\tilde{x}) - x\right\|^2 / (4\gamma_{k+1})]\mathrm{d}\tilde{x}.$$

Hence, we get that for any $x \in \mathbb{R}^d$

$$\nabla \log p_{k+1}(x) = \mathbb{E}[\mathcal{T}_{k+1}(X_k) - X_{k+1} | X_{k+1} = x]/(2\gamma_{k+1}) = -(2\gamma_{k+1})^{1/2} \mathbb{E}[Z_{k+1} | X_{k+1} = x]. \tag{49}$$

From this formula we derive a regression problem similar to the one of Section 2.1. We obtain Algorithm 2. We highlight a few differences between our approach and the ones of Song and Ermon (2019); Ho et al. (2020):

(a) As emphasized in (49), the regression problem in Algorithm 2 is different from the one usually considered in SGM which restrict themselves to the setting $f_k(x) = \alpha x$ with $\alpha = 0$ (Song and Ermon, 2019) or $\alpha > 0$ (Ho et al., 2020).

(b) In the present algorithm we do not use any corrector step (Song et al., 2021) at sampling time. Note that the use of a corrector step is only justified in the context of classical SGM algorithms and not the DSB method introduced in Section 3.3. This is because, we do not have access to the marginal of the time-reverse density during the IPF iterations contrary to classical SGMs.

(c) Finally, we do not present the Exponential Moving Average (EMA) procedure Song and Ermon (2020) which is key to prevent the network from oscillating. Contrary to the corrector step, this technique can easily be incorporated in Algorithm 2.

Further comments and additional techniques are presented in Appendix I.

C.3.2 Links with existing work

In this section, we show that we can recover the training and sampling algorithm of Song and Ermon (2019) and Ho et al. (2020) by reversing homogeneous diffusions. Note that Song et al. (2021) identified links with non-homogeneous SDEs. We explicitly characterize the fundamental difference between the approaches of Song and Ermon (2019); Ho et al. (2020) by identifying the two corresponding forward homogeneous processes (Brownian motion or Ornstein-Ulhenbeck).

Brownian motion First, we show that we can recover the sampling procedure and the loss function of Song and Ermon (2019) by reversing a Brownian motion. Assume that we have

$$\mathrm{d}\mathbf{X}_t = \sqrt{2}\mathrm{d}\mathbf{B}_t, \qquad \mathbf{X}_0 \sim p_{\text{data}}. \tag{50}$$

In what follows we define $\{Y_k\}_{k=0}^{N-1}$ such that $\{Y_k\}_{k=0}^{N-1}$ approximates $\{\mathbf{X}_{T-t_k}\}_{k=0}^{N-1}$ for a specific sequence of times $\{t_k\}_{k=0}^{N-1} \in [0,T]^N$. We recall that the time-reversal of (50) is associated with the following SDE

$$\mathrm{d}\mathbf{Y}_t = 2\nabla \log p_{T-t}(\mathbf{Y}_t) + \sqrt{2}\mathrm{d}\mathbf{B}_t. \tag{51}$$

The Euler-Maruyama discretization of (51) yields for any $k \in \{0, \ldots, N-1\}$

$$\tilde{Y}_{k+1} = \tilde{Y}_k + 2\gamma_{k+1} \nabla \log p_{T-t_k}(\tilde{Y}_k) + \sqrt{2\gamma_{k+1}} Z_{k+1}.$$

Algorithm 2 Generalized score-matching

Inputs: $(b_k)_{k\in\{0,\dots,N-1\}}$, $N \in \mathbb{N}$ (nb. of iterations), $M \in \mathbb{N}$ (batch size), N_{epochs} (nb. of epochs), $(\gamma_k)_{k\in\{0,\dots,N-1\}}$ (stepsizes), $\{s_\theta : \theta \in \Theta\}$ (neural network), opt (optimizer), p_{prior} (prior distribution), $\lambda(k)$ (weights)
for $n_{\text{epoch}} = 0, \dots, N_{\text{epoch}} - 1$ **do**
  for $j \in \{1, \dots, M\}$ **do**
    $X_0^j \sim p_{\text{data}}$
    for $k \in \{0, \dots, N-1\}$ **do**
      $X_{k+1}^j = X_k^j + \gamma_{k+1} f_k(X_k^j) + \sqrt{2\gamma_{k+1}} Z_{k+1}^j$
    end for
  end for
  $\widehat{\ell}(\theta) = M^{-1} \sum_{j=1}^{M} \sum_{k=0}^{N-1} \lambda(k)/(2\gamma_{k+1}) \sum_{j=1}^{M} \| \sqrt{2\gamma_{k+1}} s_\theta(k+1, X_{k+1}^j) + Z_{k+1}^j \|^2$
  $\theta_{n_{\text{epoch}}+1} = \text{opt}(\ell, \theta_{n_{\text{epoch}}})$
end for
$X_N \sim p_{\text{prior}}$
for $k \in \{N-1, \dots, 0\}$ **do**
  $X_k = X_{k+1} + \gamma_{k+1}\{-f_k(X_{k+1}) + 2 s_{\theta_{N_{\text{epoch}}}}(k+1, X_{k+1})\} + \sqrt{2\gamma_{k+1}} Z_{k+1}$
end for
Output: X_0

where $\{\gamma_{k+1}\}_{k=0}^{N-1}$ is a sequence of stepsizes and for any $k \in \{0, \dots, N\}$, $t_k = \sum_{j=0}^{k-1} \gamma_{j+1}$. A close form for $\{\nabla \log p_{T-t_k}\}_{k=0}^{N-1}$ is not available and in practice we consider

$$Y_{k+1} = Y_k + 2\gamma_{k+1} s_{\theta^\star}(T - t_k, Y_k) + \sqrt{2\gamma_{k+1}} Z_{k+1}, \tag{52}$$

where for any $k \in \{0, \dots, N-1\}$, $s_{\theta^\star}(T - t_k, \cdot)$ is an approximation of $\nabla \log p_{T-t_k}$. The sampling procedure (52) is similar to the one of Song and Ermon (2019) upon setting (with the notations of Song and Ermon (2019)) $T \leftarrow 1$ in (Song and Ermon, 2019, Algorithm 1) (no corrector step), $\alpha_k/2 \leftarrow \gamma_k$ and $\mathbf{s}_{\boldsymbol{\theta}}(\cdot, \sigma_{k+1}) \leftarrow 2 s_{\theta^\star}(T - t_k, \cdot)$. It remains to show that $2 s_{\theta^\star}$ is the solution to the same regression problem as $\mathbf{s}_{\boldsymbol{\theta}}$ in (Song and Ermon, 2019, Equation 6). First, note that for any $t > 0$ and $x_t \in \mathbb{R}^d$ we have

$$p_t(x_t) = (4\pi t)^{-d/2} \int_{\mathbb{R}^d} p_{\text{data}}(x_0) \exp[-\|x_t - x_0\|^2 / (4t)] \mathrm{d}x_0.$$

Therefore, we get that for any $t > 0$ and $x_t \in \mathbb{R}^d$

$$\nabla \log p_t(x_t) = \int_{\mathbb{R}^d} (x_0 - x_t)/(2t) p_{0|t}(x_0|x_t) \mathrm{d}x_0 = \mathbb{E}[\mathbf{X}_0 - \mathbf{X}_t | \mathbf{X}_t = x_t]/(2t).$$

Hence, we have that $\theta^\star$ satisfies the following regression problem

$$\theta^\star = \arg\min_\theta \sum_{k=0}^{N-1} \lambda(k) \mathbb{E}[\|(\mathbf{X}_0 - \mathbf{X}_{T-t_k})/(T - t_k) - 2 s_\theta(T - t_k, \mathbf{X}_{T-t_k})\|].$$

Note that this loss function is similar to the one of (Song and Ermon, 2019, Equation 6) upon letting $\sigma_{k+1}^2 \leftarrow 2(T - t_k)$ and $L \leftarrow N$. Hence, the two recursions approximately define the same scheme if for any $k \in \{0, \dots, N-1\}$, $\sigma_1^2 - \sigma_{k+1}^2 \approx (1/2) \sum_{j=0}^{k-1} \alpha_{j+1}$ since $t_0 = 0$ implies $T = (1/2)\sigma_1^2$. In Song and Ermon (2019) we have for any $k \in \{0, \dots, N-1\}$, $\sigma_k^2 = \kappa^{N-k} \sigma_N^2$ (recall that $N = L$) with $\kappa > 1$. In addition, we have for any $k \in \{0, \dots, N-1\}$, $\alpha_k = \varepsilon \sigma_k^2 / \sigma_N^2$ for some $\varepsilon > 0$. We get that

$$\begin{aligned}
(1/2) \textstyle\sum_{j=0}^{k-1} \alpha_{j+1} &= (\varepsilon/2) \kappa^{N-1} \textstyle\sum_{j=0}^{k-1} \kappa^{-j} \\
&= (\varepsilon/2)(\kappa^{N-1} - \kappa^{N-k-1})/(1 - \kappa^{-1}) \\
&= \varepsilon/(2(1 - \kappa^{-1})\sigma_N^2)(\sigma_1^2 - \sigma_{k+1}^2).
\end{aligned}$$

Hence, the two schemes are identical if $\varepsilon = 2(1 - \kappa^{-1})\sigma_N^2$. In practice in Song and Ermon (2019) the authors choose $N = 10$, $\sigma_N = 10^{-2}$, $\sigma_1 = 1$ (hence $\kappa = 10^{4/9}$) and $\varepsilon = 2 \times 10^{-5}$. We have $2(1 - \kappa^{-1})\sigma_N^2 \approx 1.3 \times 10^{-4}$ which has one order of difference with ε.

Ornstein-Ulhenbeck Second, we show that we can recover the sampling procedure and the loss function of Ho et al. (2020) by reversing an Ornstein-Ulhenbeck process. Contrary to the previous analysis we do not show a strict equivalence between the two recursions but instead that our algorithm can be seen as a first order approximation of the one of Ho et al. (2020).

In this section, we consider the following diffusion

$$\mathrm{d}\mathbf{X}_t = -\alpha \mathbf{X}_t \mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t, \qquad \mathbf{X}_0 \sim p_{\text{data}}. \tag{53}$$

In what follows we define $\{Y_k\}_{k=0}^{N-1}$ such that $\{Y_k\}_{k=0}^{N-1}$ approximates $\{\mathbf{X}_{T-t_k}\}_{k=0}^{N-1}$ for a specific sequence of times $\{t_k\}_{k=0}^{N-1} \in [0,T]^N$. We recall that the time-reversal of (53) is associated with the following SDE

$$\mathrm{d}\mathbf{Y}_t = \{\alpha \mathbf{Y}_t + 2\nabla \log p_{T-t}(\mathbf{Y}_t)\}\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t. \tag{54}$$

In what follows, we fix $\alpha = 1$. The Euler-Maruyama discretization of (54) yields for any $k \in \{0, \ldots, N-1\}$

$$\tilde{Y}_{k+1} = (1+\gamma_{k+1})\tilde{Y}_k + 2\gamma_{k+1}\nabla \log p_{T-t_k}(\tilde{Y}_k) + \sqrt{2\gamma_{k+1}} Z_{k+1}.$$

where $\{\gamma_{k+1}\}_{k=0}^{N-1}$ is a sequence of stepsizes and for any $k \in \{0, \ldots, N-1\}$, $t_k = \sum_{j=0}^{k-1} \gamma_{j+1}$. A close form for $\{\nabla \log p_{T-t_k}\}_{k=0}^{N-1}$ is not available and in practice we consider

$$Y_{k+1} = (1+\gamma_{k+1})Y_k + 2\gamma_{k+1} s_{\theta^\star}(T-t_k, Y_k)\mathrm{d}t + \sqrt{2\gamma_{k+1}} Z_{k+1}. \tag{55}$$

In (Ho et al., 2020, Equation 11) the backward recursion is given for any $k \in \{0, \ldots, N-1\}$

$$Y_{k+1} = \alpha_{N-k}^{-1/2}(Y_k - \beta_{N-k}/(1-\bar{\alpha}_{N-k})^{1/2}\boldsymbol{\epsilon}_\theta(Y_k, T-t_k)) + \sigma_{N-k} Z_{k+1}. \tag{56}$$

In (56) we set $\sigma_k^2 = \beta_k$ as suggested in Ho et al. (2020) where for any $k \in \{0, \ldots, N-1\}$

$$\sigma_{k+1}^2 = \beta_{k+1}, \quad \alpha_{k+1} = 1 - \beta_{k+1}, \quad \bar{\alpha}_{k+1} = \textstyle\prod_{i=1}^{k+1} \alpha_i.$$

We consider a first-order expansion of (56) with respect to $\{\beta_{k+1}\}_{k=0}^{N-1}$. We obtain the following recursion for any $k \in \{0, \ldots, N-1\}$

$$Y_{k+1} = (1+\beta_{N-k}/2)Y_k - \beta_{N-k}/(1-\bar{\alpha}_{N-k})^{1/2}\boldsymbol{\epsilon}_\theta(Y_k, T-t_k) + \sqrt{\beta_{N-k}} Z_{k+1}.$$

This last recursion is equivalent to (55) upon setting $\beta_{N-k} \leftarrow 2\gamma_{k+1}$ and $-\boldsymbol{\epsilon}_\theta(\cdot, T-t_k)/(1-\bar{\alpha}_{N-k})^{1/2} \leftarrow -s_{\theta^\star}(T-t_k, \cdot)$. It remains to show that $s_{\theta^\star}$ is the solution to the same regression problem as $\boldsymbol{\epsilon}_\theta/(1-\bar{\alpha}_{N-\cdot})$ in (Ho et al., 2020, Equation 12). First, note that for any $t > 0$ and $x_t \in \mathbb{R}^d$ we have

$$p_t(x_t) = (2\pi\bar{\sigma}_t^2)^{-d/2} \textstyle\int_{\mathbb{R}^d} p_{\text{data}}(x_0) \exp[-\left\|x_t - c_t x_0\right\|^2 / (2\bar{\sigma}_t^2)]\mathrm{d}x_0,$$

with

$$c_t^2 = \exp[-2t], \qquad \bar{\sigma}_t^2 = 1 - \exp[-2t].$$

Therefore we get that for any $t \in [0,T]$ and $x_t \in \mathbb{R}^d$

$$\begin{aligned}\nabla \log p_t(x_t) &= \textstyle\int_{\mathbb{R}^d} (c_t x_0 - x_t) p_{\text{data}}(x_0) \exp[-\left\|x_t - c_t x_0\right\|^2 / (2\bar{\sigma}_t^2)]\mathrm{d}x_0 \\ &= \mathbb{E}\left[c_t \mathbf{X}_0 - \mathbf{X}_t | \mathbf{X}_t = x_t\right] / \bar{\sigma}_t^2 = -\mathbb{E}\left[\mathbf{Z} | \mathbf{X}_t = x_t\right] / \bar{\sigma}_t,\end{aligned}$$

where we recall that $\mathbf{X}_t$ has the same distribution as $c_t \mathbf{X}_0 + \bar{\sigma}_t \mathbf{Z}$, with $\mathbf{Z}$ a d-dimensional Gaussian random variable with zero mean and identity covariance matrix. Hence, we have that $\theta^\star$ satisfies the following regression problem

$$\theta^\star = \arg\min_\theta \textstyle\sum_{k=0}^{N-1} \lambda(k) \mathbb{E}[\|\mathbf{Z}/\sigma_{T-t_k} + s_\theta(T-t_k, \mathbf{X}_{T-t_k})\|].$$

Note that we have

$$\textstyle\sum_{i=1}^{N-k} \beta_i = \sum_{i=k}^{N-1} \beta_{N-i} = 2\sum_{i=k}^{N-1} \gamma_{i+1} = 2(T-t_k).$$

Using this result we have for any $k \in \{0, \ldots, N-1\}$

$$1 - \bar{\alpha}_{N-k} = 1 - \exp[-\textstyle\sum_{i=1}^{N-k} \log(1-\beta_i)] \approx 1 - \exp[-\sum_{i=1}^{N-k} \beta_i] \approx \bar{\sigma}_{T-t_k}^2.$$

Let $\tilde{\theta}^\star$ the minimizer of (Ho et al., 2020, Equation 12) we have

$$\begin{aligned}\tilde{\theta}^\star &\approx \arg\min_\theta \textstyle\sum_{k=0}^{N-1} (2\alpha_{N-k}(1-\alpha_{N-k}))^{-1} \mathbb{E}[\|\mathbf{Z} - \boldsymbol{\epsilon}_\theta(\mathbf{X}_{T-t_k}, T-t_k)\|^2] \\ &\approx \arg\min_\theta \textstyle\sum_{k=0}^{N-1} (2\alpha_{N-k})^{-1} \mathbb{E}[\|\mathbf{Z}/(1-\alpha_{N-k})^{1/2} - \boldsymbol{\epsilon}_\theta(\mathbf{X}_{T-t_k}, T-t_k)/(1-\alpha_{N-k})^{1/2}\|^2] \\ &\approx \arg\min_\theta \textstyle\sum_{k=0}^{N-1} (2\alpha_{N-k})^{-1} \mathbb{E}[\|\mathbf{Z}/\bar{\sigma}_{T-t_k} + s_\theta(T-t_k, \mathbf{X}_{T-t_k})\|^2].\end{aligned}$$

Hence the two regression problems are approximately the same (for small values of $\{\beta_{k+1}\}_{k=0}^{N-1}$) if we set $\lambda(k) = (2\alpha_{N-k})^{-1}$.

D Schrödinger bridges with potentials and DSB recursion

In this section, we start by proving an additive formula for the Kullback–Leibler divergence in Appendix D.1 following Léonard (2014a). We recall the classical IPF formulation using potentials in Appendix D.2. Then, Proposition 2 is proved in Appendix D.3. Finally, we highlight a link between our formulation and autoencoders in Appendix D.4.

D.1 Additive formula for the Kullback–Leibler divergence

In this section, we prove a formula for the Kullback–Leibler divergence following the proof of Léonard (2014a) which extends the result to unbounded measures defined on the space of right-continuous left-limited functions from $[0, T]$. We recall that a Polish space is a complete metric separable space.

We start with the following disintegration theorem for probability measures.

Theorem 22. *Let $(\mathsf{X}, \mathcal{X})$ and $(\mathsf{Y}, \mathcal{Y})$ be two Polish spaces. Let $\pi \in \mathscr{P}(\mathsf{X})$ and $\varphi : \mathsf{X} \to \mathsf{Y}$ measurable. Then there exists a Markov kernel $\mathrm{K}_\varphi^\pi : \mathsf{Y} \times \mathcal{X} \to [0, 1]$ such that the following hold:*

(a) For any $y \in \mathsf{Y}$, $\mathrm{K}_\varphi^\pi(y, \varphi^{-1}(\{y\})) = 1$.

(b) For any $f : \mathsf{X} \to [0, +\infty)$ measurable we have $\int_\mathsf{X} f(x)\mathrm{d}\pi(x) = \int_\mathsf{Y} \mathrm{K}_\varphi^\pi(y, f)\mathrm{d}\pi_\varphi(y)$,

where $\pi_\varphi = \varphi_\# \pi$.

Proof. See (Dellacherie and Meyer, 1988, III-70) for instance. □

K_φ^π is called the disintegration of π w.r.t. φ and is unique, see (Dellacherie and Meyer, 1988, III-70). In particular, for any X-valued random variable X with distribution π we have $\mathbb{E}[f(X)|\varphi(X)] = \mathrm{K}_\varphi^\pi(\varphi(X), f)$. Next we prove the following proposition, see (Léonard, 2014a, Proposition A.13) for an extension to unbounded measures. In what follows, for any $\varphi : \mathsf{X} to \mathbb{R}$ measurable we denote $\pi_\varphi = \varphi_\# \pi$.

Proposition 23. *Let $(\mathsf{X}, \mathcal{X})$ and $(\mathsf{Y}, \mathcal{Y})$ be two Polish spaces. Let $\pi, \mu \in \mathscr{P}(\mathsf{X})$ and $\varphi : \mathsf{X} \to \mathsf{Y}$ measurable. Assume that $\pi \ll \mu$. Then the following holds:*

(a) $\pi_\varphi \ll \mu_\varphi$

(b) There exists $\mathsf{A} \in \mathcal{Y}$ with $\pi_\varphi(\mathsf{A}) = 1$ such that for any $y \in \mathsf{A}$, $\mathrm{K}_\varphi^\pi(y, \cdot) \ll \mathrm{K}_\varphi^\mu(y, \cdot)$.

In addition, we have for any $y \in \mathsf{Y}$, $y' \in \mathsf{A}$ and $x \in \mathsf{X}$

$$(\mathrm{d}\pi_\varphi/\mathrm{d}\mu_\varphi)(y) = \mathrm{K}_\varphi^\mu(y, (\mathrm{d}\pi/\mathrm{d}\mu)), \quad (\mathrm{dK}_\varphi^\pi(y', \cdot)/\mathrm{dK}_\varphi^\mu(y', \cdot))(x) = (\mathrm{d}\pi/\mathrm{d}\mu)(x)/(\mathrm{d}\pi_\varphi/\mathrm{d}\mu_\varphi)(y').$$

Finally, there exists $\mathsf{C} \in \mathcal{X}$ with $\pi(\mathsf{C}) = 1$ such that for any $x \in \mathsf{C}$ we have

$$(\mathrm{d}\pi/\mathrm{d}\mu)(x) = (\mathrm{d}\pi_\varphi/\mathrm{d}\mu_\varphi)(\varphi(x))(\mathrm{dK}_\varphi^\pi(\varphi(x), \cdot)/\mathrm{dK}_\varphi^\mu(\varphi(x), \cdot))(x).$$

Proof. Let $f : \mathsf{X} \to [0, +\infty)$ measurable. Using Theorem 22 we have

$$\pi_\varphi[f] = \int_\mathsf{X} f(\varphi(x))\mathrm{d}\pi(x) = \int_\mathsf{X} f(\varphi(x))(\mathrm{d}\pi/\mathrm{d}\mu)(x)\mathrm{d}\mu(x) = \int_\mathsf{X} f(y)\mathrm{K}_\varphi^\mu(y, (\mathrm{d}\pi/\mathrm{d}\mu))\mathrm{d}\mu_\varphi(y),$$

which concludes the first part of the proof. For the second part of the proof, let $\mathsf{B} = \{y \in \mathsf{Y} : (\mathrm{d}\pi_\varphi/\mathrm{d}\mu_\varphi)(y) = 0\}$. We have

$$0 = \int_\mathsf{Y} \mathbb{1}_\mathsf{B}(y)(\mathrm{d}\pi_\varphi/\mathrm{d}\mu_\varphi)(y)\mathrm{d}\mu_\varphi(y) = \pi_\varphi(\mathsf{B}).$$

Therefore, there exists $\mathsf{A}_1 \in \mathcal{Y}$ such that $\pi_\varphi(\mathsf{A}_1) = 1$ and for any $y \in \mathsf{A}_1$, $(\mathrm{d}\pi_\varphi/\mathrm{d}\mu_\varphi)(y) > 0$. Let $g : \mathsf{Y} \to [0, +\infty)$. Using Theorem 22 we have

$$\int_\mathsf{X} g(\varphi(x))f(x)\mathrm{d}\pi(x) = \int_\mathsf{X} g(\varphi(x))f(x)(\mathrm{d}\pi/\mathrm{d}\mu)(x)\mathrm{d}\mu(x) = \int_\mathsf{Y} g(y)\mathrm{K}_\varphi^\mu(y, f \times (\mathrm{d}\pi/\mathrm{d}\mu))\mathrm{d}\mu_\varphi(y).$$

Similarly, using Theorem 22 we have

$$\int_\mathsf{X} g(\varphi(x))f(x)\mathrm{d}\pi(x) = \int_\mathsf{Y} g(y)\mathrm{K}_\varphi^\pi(y, f)\mathrm{d}\pi_\varphi(y) = \int_\mathsf{Y} g(y)\mathrm{K}_\varphi^\pi(y, f)(\mathrm{d}\pi_\varphi/\mathrm{d}\mu_\varphi)(y)\mathrm{d}\pi_\varphi(y).$$

Hence, we get that there exists $\mathsf{A}_2 \in \mathcal{Y}$ with $\mu_\varphi(\mathsf{A}_2) = 1$ (hence $\pi_\varphi(\mathsf{A}_2) = 1$) such that for any $y \in \mathsf{A}_2$ we have

$$\mathrm{K}^\pi_\varphi(y, f)(\mathrm{d}\pi_\varphi/\mathrm{d}\mu_\varphi)(y) = \mathrm{K}^\mu_\varphi(y, f \times (\mathrm{d}\pi/\mathrm{d}\mu)).$$

We conclude upon letting $\mathsf{A} = \mathsf{A}_1 \cap \mathsf{A}_2$ and using the fact that for any $y \in \mathsf{A}$, $(\mathrm{d}\pi_\varphi/\mathrm{d}\mu_\varphi)(y) > 0$. Finally, since $\pi_\varphi(\mathsf{A}) = 1$ if and only if $\pi(\varphi^{-1}(\mathsf{A})) = 1$, we have for any $x \in \varphi^{-1}(\mathsf{A})$

$$(\mathrm{d}\pi/\mathrm{d}\mu)(x) = (\mathrm{d}\pi_\varphi/\mathrm{d}\mu_\varphi)(\varphi(x))(\mathrm{dK}^\pi_\varphi(\varphi(x), \cdot)/\mathrm{dK}^\mu_\varphi(\varphi(x), \cdot))(x),$$

which concludes the proof. □

We are now ready to state the additive formula.

Proposition 24. *Let $(\mathsf{X}, \mathcal{X})$ and $(\mathsf{Y}, \mathcal{Y})$ be two Polish spaces and $\pi, \mu \in \mathscr{P}(\mathsf{X})$ with $\pi \ll \mu$. Then for any $\varphi : \mathsf{X} \to \mathsf{Y}$ we have*

$$\mathrm{KL}(\pi|\mu) = \mathrm{KL}(\pi_\varphi|\mu_\varphi) + \textstyle\int_\mathsf{Y} \mathrm{KL}(\mathrm{K}^\pi_\varphi(y, \cdot)|\mathrm{K}^\mu_\varphi(y, \cdot))\mathrm{d}\pi_\varphi(y).$$

Proof. First assume that $\int_\mathsf{X} |\log((\mathrm{d}\pi/\mathrm{d}\mu)(x))| \, \mathrm{d}\pi(x) = +\infty$. Then, using Proposition 23 we have $\int_\mathsf{X} |\log((\mathrm{d}\pi_\varphi/\mathrm{d}\mu_\varphi)(\varphi(x)))| \, \mathrm{d}\pi(x) = +\infty$ or $\int_\mathsf{X} \left|\log((\mathrm{dK}^\pi_\varphi(\varphi(x), \cdot)/\mathrm{dK}^\mu_\varphi(\varphi(x), \cdot))(x))\right| \mathrm{d}\pi(x) = +\infty$, *i.e.* either $\mathrm{KL}(\pi_\varphi|\mu_\varphi) = +\infty$ or $\int_\mathsf{X} \mathrm{KL}(\mathrm{K}^\pi_\varphi(\varphi(x), \cdot)|\mathrm{K}^\mu_\varphi(\varphi(x), \cdot))\mathrm{d}\pi(x) = +\infty$ using Theorem 22, which concludes the first part of the proof. Second, assume that $\int_\mathsf{X} |\log((\mathrm{d}\pi/\mathrm{d}\mu)(x))| \, \mathrm{d}\pi(x) < +\infty$. Using Pinsker's inequality (Bakry et al., 2014, Equation 5.2.2) we get that $\mathrm{KL}(\pi_\varphi|\mu_\varphi) < +\infty$, *i.e.* $\int_\mathsf{X} |\log((\mathrm{d}\pi_\varphi/\mathrm{d}\mu_\varphi)(\varphi(x)))| \, \mathrm{d}\pi(x) < +\infty$. Hence, we get that $\int_\mathsf{X} \left|\log((\mathrm{dK}^\pi_\varphi(\varphi(x), \cdot)/\mathrm{dK}^\mu_\varphi(\varphi(x), \cdot))(x))\right| \mathrm{d}\pi(x) < +\infty$. Therefore we have

$$\mathrm{KL}(\pi|\mu) = \mathrm{KL}(\pi_\varphi|\mu_\varphi) + \textstyle\int_\mathsf{Y} \mathrm{KL}(\mathrm{K}^\pi_\varphi(y, \cdot)|\mathrm{K}^\mu_\varphi(y, \cdot))\mathrm{d}\pi_\varphi(y)$$

which concludes the proof □

We emphasize that in the case where $\mathsf{X} = \mathbb{R}^d \times \mathbb{R}^d$, $\varphi = \mathrm{proj}_0$ the projection on the first variable and π, μ admit densities w.r.t. the Lebesgue measure denoted p and q such that for any $x, y \in \mathbb{R}^d$, $p(x, y) = p_0(x)p_{1|0}(y|x)$ and $q(x, y) = q_0(x)q_{1|0}(y|x)$ then one can avoid using disintegration theory and Proposition 24 can be proved directly.

D.2 Iterative Proportional Fitting via potentials

In this section, before recalling the usual definition of the IPF via potentials we provide a condition under which the IPF sequence is well-defined which is used throughout Section 3.2.

Proposition 25. *Assume that there exists $\tilde{\pi} \in \mathscr{P}_{N+1}$ such that $\tilde{\pi}_0 = p_{\text{data}}$, $\tilde{\pi}_N = p_{\text{prior}}$ and $\mathrm{KL}(\tilde{\pi}|\pi^0) < +\infty$. Then the IPF sequence is well-defined.*

Proof. We prove the existence of the IPF sequence by recursion. First, note that π^1 is well-defined since $\tilde{\pi} \in \mathscr{P}_{N+1}$ with $\tilde{\pi}_N = p_{\text{prior}}$ and $\mathrm{KL}(\tilde{\pi}|\pi^0) < +\infty$. Second, assume that the sequence is well-defined up to n with $n \in \mathbb{N}$. Using (Csiszár, 1975, Theorem 2.2) we have

$$\mathrm{KL}(\tilde{\pi}|\pi^0) = \mathrm{KL}(\tilde{\pi}|\pi^n) + \textstyle\sum_{j=0}^{n-1} \mathrm{KL}(\pi^{j+1}|\pi^j).$$

Hence $\mathrm{KL}(\tilde{\pi}|\pi^n) < +\infty$. Using that $\tilde{\pi}_0 = p_{\text{data}}$ if n is odd and that $\tilde{\pi}_N = p_{\text{prior}}$ if n is even, we get that π^{n+1} is well-defined, which concludes the proof. □

We now introduce the IPF using potentials. This construction is not new and can be found in Bernton et al. (2019); Chen et al. (2016, 2021b); Pavon et al. (2021); Peyré and Cuturi (2019) for instance (in continuous state spaces). In discrete settings the recursion can be found in the following earlier works Kruithof (1937); Deming and Stephan (1940); Fortet (1940); Sinkhorn and Knopp (1967); Kullback (1968); Ruschendorf et al. (1995). The IPF is defined by the following recursion $\pi^0 = p$ given in (1) and for $n \geq 0$

$$\pi^{2n+1} = \arg\min \left\{ \mathrm{KL}(\pi|\pi^{2n}) \, : \, \pi \in \mathscr{P}_{N+1}, \; \pi_N = p_{\text{prior}} \right\},$$
$$\pi^{2n+2} = \arg\min \left\{ \mathrm{KL}(\pi|\pi^{2n+1}) \, : \, \pi \in \mathscr{P}_{N+1}, \; \pi_0 = p_{\text{data}} \right\}.$$

In the classical IPF presentation we obtain under mild assumptions that π^{2n+1} admits a density q^n w.r.t the Lebesgue measure and that π^{2n} admits a density p^n w.r.t the Lebesgue measure, given by the following expressions

$$q^n(x_{0:N}) = p^n_{\text{data}}(x_0) \prod_{k=0}^{N-1} p^{n+1}_{k+1|k}(x_{k+1}|x_k), \tag{57}$$
$$p^{n+1}(x_{0:N}) = p_{\text{data}}(x_0) \prod_{k=0}^{N-1} p^{n+1}_{k+1|k}(x_{k+1}|x_k),$$

where $(p^n_{\text{data}}(x_0))_{n\in\mathbb{N}}$ and $(p^n_{k+1|k}(x_{k+1}|x_k))_{n\in\mathbb{N}}$ are densities which are iteratively computed, with $p^0_{k+1|k} = p_{k+1|k}$.

In the context of generative modelling the derivation (57) is not useful because it does not provide a generative model, *i.e.* a probabilistic transition from p_{prior} to p_{data} but instead defines a transition from p_{data} to p_{prior}. Therefore, in this section only, we reverse the roles of p_{prior} and p_{data} and consider a reference density $\bar{p}$ such that for any $x_{0:N} \in \mathcal{X}$ we have

$$\bar{p}(x_{0:N}) = p_{\text{prior}}(x_0) \prod_{k=0}^{N-1} \bar{p}_{k+1|k}(x_{k+1}|x_k). \tag{58}$$

Then, we consider the following recursion $\pi^0 = \bar{p}$ given in (58) and for $n \in \mathbb{N}$

$$\pi^{2n+1} = \arg\min \left\{ \mathrm{KL}(\pi|\pi^{2n}) \,:\, \pi \in \mathscr{P}_{N+1},\ \pi_N = p_{\text{data}} \right\}, \tag{59}$$
$$\pi^{2n+2} = \arg\min \left\{ \mathrm{KL}(\pi|\pi^{2n+1}) \,:\, \pi \in \mathscr{P}_{N+1},\ \pi_0 = p_{\text{prior}} \right\}.$$

Again, we emphasize that the roles of p_{prior} and p_{data} are exchanged in this formulation. Using the classical IPF presentation we obtain the following expressions under mild assumptions

$$\bar{q}^n(x_{0:N}) = p^n_{\text{prior}}(x_0) \prod_{k=0}^{N-1} \bar{p}^{n+1}(x_{k+1}|x_k), \tag{60}$$
$$\bar{p}^{n+1}(x_{0:N}) = p_{\text{prior}}(x_0) \prod_{k=0}^{N-1} \bar{p}^{n+1}(x_{k+1}|x_k).$$

In this case, we get that π^{2n+1} (approximately) defines a generative model for large values of $n \in \mathbb{N}$ since it provides a transition from to p_{prior} to (approximately) p_{data}. In the following proposition we give the precise statement corresponding to (60). We assume that $\bar{p}^0 = \bar{p}$.

Proposition 26. *Assume that* $\mathrm{KL}(p_{\text{prior}} \otimes p_{\text{data}}|\bar{p}_{0,N}) < +\infty$. *Then* $(\pi^n)_{n\in\mathbb{N}}$ *given by* (59) *is well-defined and for any* $n \in \mathbb{N}$ *we have that* π^{2n+1} *and* π^{2n+2} *admit a density w.r.t. the Lebesgue measures denoted* $\bar{q}^n$ *and* $\bar{p}^{n+1}$. *In addition, we have for any* $n \in \mathbb{N}$ *and* $x_{0:N} \in \mathcal{X}$

$$\bar{q}^n(x_{0:N}) = p^n_{\text{prior}}(x_0) \prod_{k=0}^{N-1} \bar{p}^{n+1}(x_{k+1}|x_k),$$
$$\bar{p}^{n+1}(x_{0:N}) = p_{\text{prior}}(x_0) \prod_{k=0}^{N-1} \bar{p}^{n+1}(x_{k+1}|x_k),$$

where for any $n \in \mathbb{N}$ *we have for any* $x_{0:N} \in \mathcal{X}$ *and* $k \in \{0, \dots, N-1\}$

$$p^n_{\text{prior}}(x_0) = \psi^n_0(x_0) p_{\text{prior}}(x_0), \quad \bar{p}^{n+1}(x_{k+1}|x_k) = \bar{p}^n(x_{k+1}|x_k) \psi^n_{k+1}(x_{k+1})/\psi^n_k(x_k),$$

with

$$\psi^n_N(x_N) = p_{\text{data}}(x_N)/\bar{p}^n_N(x_N), \quad \psi^n_k(x_k) = \textstyle\int_{\mathbb{R}^d} \psi^n_{k+1}(x_{k+1}) \bar{p}^n(x_{k+1}|x_k) \mathrm{d}x_{k+1}.$$

Proof. Let $\tilde{\pi} = (p_{\text{prior}} \otimes p_{\text{data}})\bar{p}_{|0,N}$. Using Proposition 24 we get that $\mathrm{KL}(\tilde{\pi}|\bar{p}) = \mathrm{KL}(p_{\text{prior}} \otimes p_{\text{data}}|\bar{p}_{0,N}) < +\infty$. Using Proposition 25 the IPF sequence is well-defined. In addition, using (Csiszár, 1975, Theorem 3.1) for any $n \in \mathbb{N}$ there exists $\psi^n_N : \mathbb{R}^d \to [0, +\infty)$ such that for any $x_{0:N} \in \mathsf{A}$ with $\tilde{\pi}(\mathsf{A}) = 1$ we have

$$\bar{q}^n(x_{0:N}) = \bar{p}^n(x_{0:N}) \psi^n_N(x_N).$$

Since $\tilde{\pi}$ is equivalent to the Lebesgue measure we get that for any $x_{0:N} \in \mathbb{R}^d$

$$\bar{q}^n(x_{0:N}) = \bar{p}^n(x_{0:N}) \psi^n_N(x_N).$$

Let $n \in \mathbb{N}$. We have for any $x_N \in \mathbb{R}^d$, $p_{\text{data}}(x_N) = \bar{q}^n(x_N) = \bar{p}^n_N(x_N)\psi^n_N(x_N)$. Hence, we get that for any $N \in \mathbb{N}$, $\psi^n_N(x_N) = p_{\text{data}}(x_N)/\bar{p}^n_N(x_N)$. For any $x_{0:N} \in \mathcal{X}$ and $k \in \{0, \dots, N-1\}$ let

$$\psi^n_k(x_k) = \textstyle\int_{\mathbb{R}^d} \psi^n_{k+1}(x_{k+1}) \bar{p}^n(x_{k+1}|x_k) \mathrm{d}x_{k+1}.$$

We obtain that for any $x_{0:N} \in \mathcal{X}$

$$\bar{q}^n(x_{0:N}) = p_{\text{prior}}(x_0)\psi_0(x_0) \prod_{k=0}^{N-1} (\bar{p}^n(x_{k+1}|x_k)\psi_{k+1}(x_{k+1})/\psi_k(x_k)).$$

Hence, we get that for any $x_{0:N} \in \mathcal{X}$, $\bar{q}^n(x_0) = p^n_{\text{prior}}(x_0) \prod_{k=0}^{N-1} \bar{p}^{n+1}(x_{k+1}|x_k)$. Using Proposition 24 we get that for any $x_{0:N} \in \mathcal{X}$, $\bar{p}^{n+1}(x_0) = p_{\text{prior}}(x_0) \prod_{k=0}^{N-1} \bar{p}^{n+1}(x_{k+1}|x_k)$, which concludes the proof. □

The previous expression is not symmetric and the IPF iterations appear as a policy refinement of the original forward dynamic $\bar{p}$. In the next proposition we present another potential formulation of the IPF iterations which is symmetric.

Proposition 27. *Assume that* $\mathrm{KL}(p_{\text{prior}} \otimes p_{\text{data}}|q_{0,N}) < +\infty$. *Then* $(\pi^n)_{n\in\mathbb{N}}$ *given by* (59) *is well-defined and for any* $n \in \mathbb{N}$ *we have that* π^{2n+1} *and* π^{2n+2} *admit a density w.r.t. the Lebesgue measures denoted* $\bar{q}^n$ *and* $\bar{p}^{n+1}$. *In addition, we have for any* $n \in \mathbb{N}$ *and* $x_{0:N} \in \mathcal{X}$

$$\bar{q}^n(x_{0:N}) = \varphi_0^n(x_0) \prod_{k=0}^{N-1} \bar{p}(x_{k+1}|x_k)\psi_N^n(x_N),$$
$$\bar{p}^{n+1}(x_{0:N}) = \varphi_0^{n+1}(x_0) \prod_{k=0}^{N-1} \bar{p}(x_{k+1}|x_k)\psi_N^n(x_N),$$

where for any $n \in \mathbb{N}$ *we have for any* $x_{0:N} \in \mathcal{X}$ *and* $k \in \{0, \dots, N-1\}$

$$\psi_N^n(x_N) = p_{\text{data}}(x_N)/\varphi_N^n(x_N), \quad \psi_k^n(x_k) = \textstyle\int_{\mathbb{R}^d} \psi_{k+1}^n(x_{k+1})\bar{p}(x_{k+1}|x_k)\mathrm{d}x_{k+1},$$
$$\varphi_0^{n+1}(x_0) = p_{\text{prior}}(x_0)/\psi_0^n(x_0), \quad \varphi_{k+1}^{n+1}(x_{k+1}) = \textstyle\int_{\mathbb{R}^d} \varphi_k^{n+1}(x_k)\bar{p}(x_{k+1}|x_k)\mathrm{d}x_k,$$

and $\varphi_0^0 = p_{\text{prior}}$ *and* $\psi_N^{-1} = 1$.

Proof. Let $\tilde{\pi} = (p_{\text{prior}} \otimes p_{\text{data}})q_{|0,N}$. Using Proposition 24 we get that $\mathrm{KL}(\tilde{\pi}|q) = \mathrm{KL}(p_{\text{prior}} \otimes p_{\text{data}}|p_{0,N}) < +\infty$. Using Proposition 25 the IPF sequence is well-defined. In addition, using (Csiszár, 1975, Theorem 3.1) for any $n \in \mathbb{N}$ there exists $\tilde{\psi}_N^n : \mathbb{R}^d \to [0, +\infty)$ such that for any $x_{0:N} \in \mathsf{A}$ with $\tilde{\pi}(\mathsf{A}) = 1$ we have

$$\bar{q}^n(x_{0:N}) = \bar{p}^n(x_{0:N})\tilde{\psi}_N^n(x_N), \qquad \bar{p}^{n+1}(x_{0:N}) = \bar{q}^n(x_{0:N})\tilde{\varphi}_0^n(x_0).$$

Since $\tilde{\pi}$ is equivalent to the Lebesgue measure we get that for any $x_{0:N} \in \mathbb{R}^d$

$$\bar{q}^n(x_{0:N}) = \bar{p}^n(x_{0:N})\tilde{\psi}_N^n(x_N), \qquad \bar{p}^{n+1}(x_{0:N}) = \bar{q}^n(x_{0:N})\tilde{\varphi}_0^n(x_0).$$

For any $n \in \mathbb{N}$, let $\psi_N^n = \psi_N^{n-1}\tilde{\psi}_N^n$ and $\varphi_0^{n+1} = \varphi_0^n\tilde{\varphi}_0^n$. By recursion, we get that for any $n \in \mathbb{N}$ and $x_{0:N} \in \mathcal{X}$

$$\bar{q}^n(x_{0:N}) = \varphi_0^n(x_0) \prod_{k=0}^{N-1} \bar{p}(x_{k+1}|x_k)\psi_N^n(x_N),$$
$$\bar{p}^{n+1}(x_{0:N}) = \varphi_0^{n+1}(x_0) \prod_{k=0}^{N-1} \bar{p}(x_{k+1}|x_k)\psi_N^n(x_N).$$

Let $n \in \mathbb{N}$. For any $x_N \in \mathbb{R}^d$ we have

$$\bar{q}_N^n(x_N) = p_{\text{data}}(x_N) = \bar{p}_N^n(x_N)\tilde{\psi}_N^n(x_n). \tag{61}$$

In addition, for any $k \in \{0, \dots, N-1\}$ and $x_{0:N} \in \mathcal{X}$ we define $\varphi_{k+1}^{n+1}(x_{k+1}) = \int_{\mathbb{R}^d} \varphi_k^{n+1}(x_k)\bar{p}(x_{k+1}|x_k)\mathrm{d}x_k$. We have for any $x_N \in \mathbb{R}^d$, $\bar{p}_N^n(x_N) = \varphi_N^n(x_N)\psi_N^{n-1}(x_n)$. Combining this result with (61) we get that for any $x_N \in \mathbb{R}^d$

$$\psi_N^n(x_N) = p_{\text{data}}(x_N)/\varphi_N^n(x_N).$$

Similarly, we get that for any $x_0 \in \mathbb{R}^d$, $\varphi_0^{n+1}(x_0) = p_{\text{prior}}(x_0)/\psi_0^n(x_0)$, which concludes the proof. □

D.3 Proof of Proposition 2

Let $\tilde{\pi} = (p_{\text{prior}} \otimes p_{\text{data}})p_{|0,N}$. Using Proposition 24 we get that $\mathrm{KL}(\tilde{\pi}|p) = \mathrm{KL}(p_{\text{prior}} \otimes p_{\text{data}}|p_{0,N}) < +\infty$. Using Proposition 25 the IPF sequence is well-defined. Note that π^0 admits a density w.r.t. the Lebesgue measure given by $p > 0$. Let $n \in \mathbb{N}$ and assume that $p^n > 0$ is given for any $x_{0:N} \in \mathcal{X}$ by

$$p^n(x_{0:N}) = p_{\text{data}}(x_0) \prod_{k=0}^{N-1} q^{n-1}(x_{k+1}|x_k). \tag{62}$$

Using Proposition 24 we get that for any $\pi \in \mathscr{P}_{N+1}$ such that $\pi_N = p_{\text{prior}}$ we have

$$\mathrm{KL}(\pi|\pi^{2n}) = \mathrm{KL}(p_{\text{prior}}|\pi_0^{2n}) + \textstyle\int_{\mathbb{R}^d} \mathrm{KL}(\pi_{|N}|\pi_{|N}^{2n})p_{\text{prior}}(x_N)\mathrm{d}x_N.$$

Hence, we have that $\pi^{2n+1} = p_{\text{prior}}\pi_{|N}^{2n}$. Since $p^n > 0$ we get that for any $\pi_{|N}^{2n}$ satisfies for any $\mathsf{A} \in \mathcal{B}(\mathcal{X})$ and $x_N \in \mathbb{R}^d$

$$\pi_{|N}^{2n}(\mathsf{A}|x_N) = \textstyle\int_{\mathsf{A}} p^n(x_{0:N})/p^n(x_N)\mathrm{d}x_{0:N}\delta_{x_N}(\mathsf{A}_N).$$

Therefore, π^{2n+1} admits a density w.r.t. the Lebesgue measure denoted q^n and given for any $x_{0:N} \in \mathcal{X}$ by

$$q^n(x_{0:N}) = p^n(x_{0:N}) p_{\text{prior}}(x_N)/p^n(x_N)$$
$$= p_{\text{prior}}(x_N) \prod_{k=0}^{N-1} p^n(x_{k+1}|x_k) p^n(x_k)/p^n(x_{k+1}) = p_{\text{prior}}(x_N) \prod_{k=0}^{N-1} p^n(x_k|x_{k+1}),$$

where we have used (62). Note that $q^n > 0$. Similarly, we get that for any $x_{0:N} \in \mathcal{X}$

$$p^{n+1}(x_{0:N}) = p_{\text{data}}(x_0) \prod_{k=0}^{N-1} q^n(x_{k+1}|x_k).$$

Note that again that $p^{n+1} > 0$. We conclude by recursion.

D.4 Link with autoencoders

Consider the maximum likelihood problem

$$q^\star = \arg\max\{\mathbb{E}_{p_{\text{data}}}[\log q_0(X_0)] \,:\, q \in \mathscr{P}_d(\mathcal{X}),\ q_N = p_{\text{prior}}\},$$

where $\mathscr{P}_d(\mathcal{X})$ is the subset of the probability distribution over $\mathcal{X}$ which admit a density w.r.t. the Lebesgue measure. Using Jensen's inequality we have for any $q \in \mathscr{P}_d(\mathcal{X})$

$$\mathbb{E}_{p_{\text{data}}}[\log q_0(X_0)] = \int_{\mathbb{R}^d} \log\Big(\int_{(\mathbb{R}^d)^{N-1}} q(x_{0:N}) p(x_{1:N}|x_0)/p(x_{1:N}|x_0) \mathrm{d}x_{1:N}\Big) p_0(x_0) \mathrm{d}x_0$$
$$\geq \int_{\mathcal{X}} \log(q(x_{0:N})/p(x_{1:N}|x_0)) p(x_{0:N}) \mathrm{d}x_{0:N} \geq -\mathrm{KL}(p|q) - \mathrm{H}(p_0).$$

This Evidence Lower Bound (ELBO) is similar to the one identified in Ho et al. (2020). Maximizing this ELBO is equivalent to solving the following problem

$$q^0 = \arg\min\{\mathrm{KL}(q|p) \,:\, q \in \mathscr{P}_d(\mathcal{X}),\ q_N = p_{\text{prior}}\},$$

which is the first step of IPF. Hence subsequent steps can be obtained by maximizing ELBOs associated with the following maximum likelihood problems for any $n \in \mathbb{N}$

$$q^\star = \arg\max\{\mathbb{E}_{p_{\text{data}}}[\log q_0(X_0)] \,:\, q \in \mathscr{P}_d(\mathcal{X}),\ q_N = p_{\text{prior}}\},$$
$$p^\star = \arg\max\{\mathbb{E}_{p_{\text{prior}}}[\log p_N(X_N)] \,:\, p \in \mathscr{P}_d(\mathcal{X}),\ p_0 = p_{\text{data}}\}.$$

E Alternative variational formulations

In this section, we draw links between IPF and score-matching techniques. We start by proving Proposition 3 in Appendix E.1. We then present alternative variational formulations in Appendix E.2.

E.1 Proof of Proposition 3

We only prove (12) since the proof (13) is similar. Let $n \in \mathbb{N}$ and $k \in \{0, \dots, N-1\}$. For any $x_{k+1} \in \mathbb{R}^d$ we have

$$p^n_{k+1}(x_{k+1}) = (4\pi\gamma_{k+1})^{-d/2} \int_{\mathbb{R}^d} p^n(x_k) \exp[-\|F^n_k(x_k) - x_{k+1}\|^2/(4\gamma_{k+1})] \mathrm{d}x_k,$$

with $F^n_k(x_k) = x_k + \gamma_{k+1} f^n_k(x_k)$. Since $p^n_k > 0$ is bounded using the dominated convergence theorem we have for any $x_{k+1} \in \mathbb{R}^d$

$$\nabla \log p^n_{k+1}(x_{k+1}) = \int_{\mathbb{R}^d} (F^n_k(x_k) - x_{k+1})/(2\gamma_{k+1}) p_{k|k+1}(x_k|x_{k+1}) \mathrm{d}x_k.$$

Therefore we get that for any $x_{k+1} \in \mathbb{R}^d$

$$b^n_{k+1}(x_{k+1}) = \int_{\mathbb{R}^d} (F^n_k(x_k) - F^n_k(x_{k+1}))/\gamma_{k+1} p_{k|k+1}(x_k|x_{k+1}) \mathrm{d}x_k.$$

This is equivalent to

$$B^n_{k+1}(x_{k+1}) = \mathbb{E}[X_{k+1} + F^n_k(X_k) - F^n_k(X_{k+1}) | X_{k+1} = x_{k+1}],$$

with $(X_k, X_{k+1}) \sim p_{k,k+1}(x_k, x_{k+1})$. Hence, we get that

$$B^n_{k+1} = \arg\min_{\mathrm{B} \in \mathrm{L}^2(\mathbb{R}^d, \mathbb{R}^d)} \mathbb{E}_{p^n_{k,k+1}}[\|\mathrm{B}(X_{k+1}) - (X_{k+1} + F^n_k(X_k) - F^n_k(X_{k+1}))\|^2],$$

which concludes the proof.

E.2 Variational formulas

In Proposition 3 and Section 3.3 we present a variational formula for B^n_{k+1} and F^{n+1}_k for any $n \in \mathbb{N}$ and $k \in \{0, \dots, N-1\}$, where we recall that for any $x \in \mathbb{R}^d$ we have

$$B^n_{k+1}(x) = x + \gamma_{k+1} b^n_{k+1}(x), \qquad F^{n+1}_k = x + \gamma_{k+1} f^{n+1}_k(x),$$

where we have

$$b^n_{k+1}(x) = -f^n_k(x) + 2\nabla \log p^n_{k+1}(x), \qquad f^{n+1}_k(x) = -b^n_{k+1}(x) + 2\nabla \log q^n_k(x). \tag{63}$$

In the rest of this section we assume that for any $n \in \mathbb{N}$, $k \in \{0, \dots, N-1\}$ and $x \in \mathbb{R}^d$ we have

$$q^n_{k|k+1}(x_k|x_{k+1}) = (4\pi\gamma_{k+1})^{-d/2} \exp[-\|x_k - B^n_{k+1}(x_{k+1})\|^2/(4\gamma_{k+1})],$$
$$p^{n+1}_{k+1|k}(x_{k+1}|x_k) = (4\pi\gamma_{k+1})^{-d/2} \exp[-\|x_{k+1} - F^{n+1}_k(x_k)\|^2/(4\gamma_{k+1})].$$

We recall that in this case Proposition 3 ensures that for any $n \in \mathbb{N}$ and $k \in \{0, \dots, N-1\}$

$$B^n_{k+1} = \arg\min_{\mathrm{B} \in \mathrm{L}^2(\mathbb{R}^d, \mathbb{R}^d)} \mathbb{E}_{p^n_{k,k+1}}[\|\mathrm{B}(X_{k+1}) - (X_{k+1} + F^n_k(X_k) - F^n_k(X_{k+1}))\|^2],$$
$$F^{n+1}_k = \arg\min_{\mathrm{F} \in \mathrm{L}^2(\mathbb{R}^d, \mathbb{R}^d)} \mathbb{E}_{q^n_{k,k+1}}[\|\mathrm{F}(X_k) - (X_k + B^n_{k+1}(X_{k+1}) - B^n_{k+1}(X_k))\|^2].$$

In the rest of this section we derive other variational formulas and discuss their practical limitations/advantages.

E.2.1 Score-matching formula and sum of networks

First, using (63) we have for any $n \in \mathbb{N}$, $k \in \{0, \dots, N-1\}$ and $x \in \mathbb{R}^d$

$$b^n_{k+1}(x) = \alpha x + 2\textstyle\sum_{j=0}^{n} \nabla \log p^j_{k+1}(x) - 2\sum_{j=0}^{n-1} \nabla \log q^j_k(x), \tag{64}$$
$$f^n_k(x) = -\alpha x + 2\textstyle\sum_{j=0}^{n-1} \nabla \log q^j_k(x) - 2\sum_{j=0}^{n-1} \nabla \log p^j_{k+1}(x). \tag{65}$$

In the following proposition we derive a variational formula for $\nabla \log p^n_{k+1}$ and $\nabla \log q^n_k(x)$ for any $n \in \mathbb{N}$ and $k \in \{0, \dots, N-1\}$.

Proposition 28. *For any $n \in \mathbb{N}$ and $k \in \{0, \dots, N-1\}$ we have*

$$\nabla \log p^n_{k+1} = \arg\min_{u \in \mathrm{L}^2(\mathbb{R}^d, \mathbb{R}^d)} \mathbb{E}_{p^n_{k,k+1}}[\|u(X_{k+1}) - (F^n_k(X_k) - X_{k+1})/(2\gamma_{k+1})\|^2], \tag{66}$$
$$\nabla \log q^n_k = \arg\min_{v \in \mathrm{L}^2(\mathbb{R}^d, \mathbb{R}^d)} \mathbb{E}_{q^n_{k,k+1}}[\|v(X_k) - (B^n_{k+1}(X_{k+1}) - X_k)/(2\gamma_{k+1})\|^2]. \tag{67}$$

Proof. The proof is similar to the one of Proposition 3 but is provided for completeness. We only prove (68) since the proof (69) is similar. Let $n \in \mathbb{N}$ and $k \in \{0, \dots, N-1\}$. For any $x_{k+1} \in \mathbb{R}^d$ we have

$$p^n_{k+1}(x_{k+1}) = (4\pi\gamma_{k+1})^{-d/2} \textstyle\int_{\mathbb{R}^d} p^n(x_k) \exp[-\|F^n_k(x_k) - x_{k+1}\|^2/(4\gamma_{k+1})] \mathrm{d}x_k,$$

with $F^n_k(x_k) = x_k + \gamma_{k+1} f^n_k(x_k)$. Since $p^n_k > 0$ is bounded using the dominated convergence theorem we have for any $x_{k+1} \in \mathbb{R}^d$

$$\nabla \log p^n_{k+1}(x_{k+1}) = \textstyle\int_{\mathbb{R}^d} (F^n_k(x_k) - x_{k+1})/(2\gamma_{k+1}) p_{k|k+1}(x_k|x_{k+1}) \mathrm{d}x_k.$$

This is equivalent to

$$\nabla \log p^n_{k+1}(x_{k+1}) = \mathbb{E}[(F^n_k(X_k) - X_{k+1})/(2\gamma_{k+1}) | X_{k+1} = x_{k+1}],$$

with $(X_k, X_{k+1}) \sim p_{k,k+1}(x_k, x_{k+1})$. Hence, we get that

$$\nabla \log p^n_{k+1} = \arg\min_{u \in \mathrm{L}^2(\mathbb{R}^d, \mathbb{R}^d)} \mathbb{E}_{p^n_{k,k+1}}[\|u(X_{k+1}) - (F^n_k(X_k) - X_{k+1})/(2\gamma_{k+1})\|^2],$$

which concludes the proof. □

Note that (68) and (69) can be simplified upon remarking that for any $n \in \mathbb{N}$ and $k \in \{0, \dots, N-1\}$

$$X_{k+1}^n = F_k^n(X_k^n) + \sqrt{2\gamma_{k+1}} Z_{k+1}^n, \quad \tilde{X}_k^n = F_k^n(\tilde{X}_{k+1}^n) + \sqrt{2\gamma_{k+1}} \tilde{Z}_{k+1}^n,$$

with $\{X_k^n\}_{k=0}^N \sim p^n$, $\{\tilde{X}_k^n\}_{k=0}^N \sim q^n$ and $\{(Z_{k+1}^n, \tilde{Z}_{k+1}^n) : n \in \mathbb{N}, k \in \{0, \dots, N-1\}\}$ a family of independent Gaussian random variables with zero mean an identity covariance matrix. Using this result we get that for any $n \in \mathbb{N}$ and $k \in \{0, \dots, N-1\}$

$$\nabla \log p_{k+1}^n = \arg\min_{u \in \mathrm{L}^2(\mathbb{R}^d, \mathbb{R}^d)} \mathbb{E}_{p_{k,k+1}^n}[\|u(X_{k+1}) - Z_{k+1}^n / \sqrt{2\gamma_{k+1}}\|^2], \tag{68}$$

$$\nabla \log q_k^n = \arg\min_{v \in \mathrm{L}^2(\mathbb{R}^d, \mathbb{R}^d)} \mathbb{E}_{q_{k,k+1}^n}[\|v(X_k) - \tilde{Z}_{k+1}^n / \sqrt{2\gamma_{k+1}}\|^2]. \tag{69}$$

In practice, neural networks $u_{\alpha^n}(k, x) \approx \nabla \log p_k^n(x)$, and $v_{\beta^n}(k, x) \approx \nabla \log q_k^n(x)$ are used. Hence, we sample approximately from q^n and p^n for any $n \in \mathbb{N}$ using the following recursion:

$$\begin{aligned} \tilde{X}_k^n &= \tilde{\tau}_{k+1} \tilde{X}_{k+1}^n + 2\gamma_{k+1} \{\textstyle\sum_{j=0}^n u_{\alpha^j}(k+1, \tilde{X}_{k+1}^n) - \sum_{j=0}^{n-1} v_{\beta^j}(k, \tilde{X}_{k+1}^n)\} + \sqrt{2\gamma_{k+1}} \tilde{Z}_{k+1}^n, \\ X_{k+1}^n &= \tau_{k+1} X_k^n + 2\gamma_{k+1} \{\textstyle\sum_{j=0}^n u_{\alpha^j}(k+1, X_k^n) - \sum_{j=0}^n v_{\beta^j}(k, X_k^n)\} + \sqrt{2\gamma_{k+1}} Z_{k+1}^n, \end{aligned} \tag{70}$$

where $\tilde{\tau}_{k+1} = 1 + \alpha\gamma_{k+1}$, $\tau_{k+1} = 1 - \alpha\gamma_{k+1}$ and $X_0^n \sim p_{\text{data}}$, $\tilde{X}_N^n \sim p_{\text{prior}}$.

E.2.2 Drift-matching formula

In Proposition 3 we have given a variational formula for B_{k+1}^n and F_k^{n+1} for any $n \in \mathbb{N}$ and $k \in \{0, \dots, N-1\}$. In Proposition 28 we have given a variational formula for $\nabla \log p_{k+1}^n$ and $\nabla \log q_k^n$ for any $n \in \mathbb{N}$ and $k \in \{0, \dots, N-1\}$. In the following proposition we give a variational formula for the drifts b_{k+1}^n and f_k^{n+1}.

Proposition 29. *For any $n \in \mathbb{N}$ and $k \in \{0, \dots, N-1\}$ we have*

$$b_{k+1}^n = \arg\min_{b \in \mathrm{L}^2(\mathbb{R}^d, \mathbb{R}^d)} \mathbb{E}_{p_{k,k+1}^n}[\|b(X_{k+1}) - (F_k^n(X_k) - F_k^n(X_{k+1}))/\gamma_{k+1}\|^2] \tag{71}$$

$$f_k^{n+1} = \arg\min_{f \in \mathrm{L}^2(\mathbb{R}^d, \mathbb{R}^d)} \mathbb{E}_{q_{k,k+1}^n}[\|f(X_k) - (B_{k+1}^n(X_{k+1}) - B_{k+1}^n(X_k))/\gamma_{k+1}\|^2] \tag{72}$$

Proof. The proof is similar to the one of Proposition 3 but is provided for completeness. We only prove (71) since the proof (72) is similar. Let $n \in \mathbb{N}$ and $k \in \{0, \dots, N-1\}$. For any $x_{k+1} \in \mathbb{R}^d$ we have

$$p_{k+1}^n(x_{k+1}) = (4\pi\gamma_{k+1})^{-d/2} \textstyle\int_{\mathbb{R}^d} p^n(x_k) \exp[-\|F_k^n(x_k) - x_{k+1}\|^2/(4\gamma_{k+1})] \mathrm{d}x_k,$$

with $F_k^n(x_k) = x_k + \gamma_{k+1} f_k^n(x_k)$. Since $p_k^n > 0$ is bounded using the dominated convergence theorem we have for any $x_{k+1} \in \mathbb{R}^d$

$$\nabla \log p_{k+1}^n(x_{k+1}) = \textstyle\int_{\mathbb{R}^d} (F_k^n(x_k) - x_{k+1})/(2\gamma_{k+1}) p_{k|k+1}(x_k|x_{k+1}) \mathrm{d}x_k.$$

Therefore we get that for any $x_{k+1} \in \mathbb{R}^d$

$$b_{k+1}^n(x_{k+1}) = \textstyle\int_{\mathbb{R}^d} (F_k^n(x_k) - F_k^n(x_{k+1}))/\gamma_{k+1} p_{k|k+1}(x_k|x_{k+1}) \mathrm{d}x_k.$$

This is equivalent to

$$b_{k+1}^n(x_{k+1}) = \mathbb{E}[(F_k^n(X_k) - F_k^n(X_{k+1}))/\gamma_{k+1} | X_{k+1} = x_{k+1}],$$

with $(X_k, X_{k+1}) \sim p(x_k, x_{k+1})$. Hence, we get that

$$b_{k+1}^n = \arg\min_{b \in \mathrm{L}^2(\mathbb{R}^d, \mathbb{R}^d)} \mathbb{E}_{p_{k,k+1}^n}[\|b(X_{k+1}) - (F_k^n(X_k) - F_k^n(X_{k+1}))/\gamma_{k+1}\|^2],$$

which concludes the proof. □

In practice, neural networks $b_{\beta^n}(k, x) \approx b_k^n(x)$, and $f_{\alpha^n}(k, x) \approx f_k^n(x)$ are used. Hence, we sample approximately from q^n and p^n for any $n \in \mathbb{N}$ using the following recursion:

$$\begin{aligned} \tilde{X}_k^n &= \tilde{X}_{k+1}^n + \gamma_{k+1} b_{\beta^n}(k+1, \tilde{X}_{k+1}^n) + \sqrt{2\gamma_{k+1}} \tilde{Z}_{k+1}^n, \\ X_{k+1}^n &= X_k^n + \gamma_{k+1} f_{\alpha^n}(k, X_k^n) + \sqrt{2\gamma_{k+1}} Z_{k+1}^n, \end{aligned}$$

with $X_0^n \sim p_{\text{data}}$, $\tilde{X}_N^n \sim p_{\text{prior}}$.

E.2.3 Discussion

We identify three variational formulas associated with Proposition 3, Proposition 28 and Proposition 29. In practice we discard the approach of Appendix E.2.1 because it requires storing $2n$ neural networks to sample from p^n, see (70). Hence the algorithm requires more memory as n increases and the sampling procedure requires $\mathcal{O}(nN)$ passes through a neural network. The approaches described in Proposition 3 and Proposition 29 yield sampling procedures which only require $\mathcal{O}(N)$ passes through a neural network and have fixed memory cost for any $n \in \mathbb{N}$. In practice we observed that the approach of Proposition 3 yields better results. We conjecture that this favorable behavior is mainly due to the architecture of the neural networks used to approximate B_{k+1}^n and F_k^{n+1} which have residual connections and therefore are better suited at representing functions of the $x \mapsto x + \Phi(x)$ where Φ is a perturbation.

F Theoretical study of Schrödinger bridges and the IPF

In this section, we explore some of the theoretical properties of Schrödinger bridges and the IPF procedure. Proposition 4 and Proposition 5 are proved in Appendix F.1 and Appendix F.2 respectively.

F.1 Proof of Proposition 4

In this section, we prove Proposition 4. First we gather novel monotonicity results for the IPF in Proposition 31, see Appendix F.1.1. Then we prove our quantitative convergence bounds in Theorem 36, see Appendix F.1.2.

F.1.1 Monotonicity results

We consider the static IPF recursion: $\pi^0 = \mu \in \mathscr{P}_2$ and

$$\pi^{2n+1} = \arg\min \left\{ \mathrm{KL}(\pi | \pi^{2n}) \, : \, \pi \in \mathscr{P}_2, \; \pi_1 = \nu_1 \right\},$$
$$\pi^{2n+2} = \arg\min \left\{ \mathrm{KL}(\pi | \pi^{2n+1}) \, : \, \pi \in \mathscr{P}_2, \; \pi_0 = \nu_0 \right\},$$

where $\nu_0, \nu_1 \in \mathscr{P}(\mathbb{R}^d)$. We also consider the following assumption.

B1. *μ is absolutely continuous w.r.t. $\mu_0 \otimes \mu_1$ and $\mathrm{KL}(\nu_0 \otimes \nu_1 | \mu) < +\infty$. In addition, ν_i and μ_i are equivalent for $i \in \{0, 1\}$.*

First we draw links between **A**1 and **B**1.

Proposition 30. **A**1 *implies* **B**1 *with* $\mu = p_{0,N}$.

Proof. Since $p_N > 0$ we get that p_N and p_{prior} are equivalent. Hence μ_1 and ν_1 are equivalent and $\mu_0 = \nu_0$. Let us show that μ is absolutely continuous w.r.t. $\mu_0 \otimes \mu_1$, *i.e.* that $p_{0,N}$ is absolutely continuous w.r.t. $p_{\text{data}} \otimes p_N$. Since $p_N > 0$ we get that $p_{0,N}$ is absolutely continuous w.r.t. $p_{\text{data}} \otimes p_N$ with density $p_{N|0}/p_N$. Finally we have

$$\begin{aligned}
&\textstyle\int_{(\mathbb{R}^d)^2} \log(p_{\text{data}}(x_0) p_{\text{prior}}(x_N) / (p_{\text{data}}(x_0) p_{N|0}(x_N|x_0))) p_{\text{data}}(x_0) p_{\text{prior}}(x_N) \mathrm{d}x_0 \mathrm{d}x_N \\
&\quad = \textstyle\int_{(\mathbb{R}^d)^2} \log(p_{\text{prior}}(x_N) / p_{N|0}(x_N|x_0)) p_{\text{data}}(x_0) p_{\text{prior}}(x_N) \mathrm{d}x_0 \mathrm{d}x_N \\
&\quad \leq |\mathrm{H}(p_{\text{prior}})| + \textstyle\int_{\mathbb{R}^d} |\log p_{N|0}(x_N|x_0)| p_{\text{data}}(x_0) p_{\text{prior}}(x_N) \mathrm{d}x_0 \mathrm{d}x_N < +\infty
\end{aligned}$$

which concludes the proof. □

In this section we prove the following proposition.

Proposition 31. *Assume* **B**1. *Then, the IPF sequence is well-defined and for any $n \in \mathbb{N}$ with $n \geq 1$ we have*

$$\mathrm{KL}(\pi^{n+1} | \pi^n) \leq \mathrm{KL}(\pi^{n-1} | \pi^n), \qquad \mathrm{KL}(\pi^n | \pi^{n+1}) \leq \mathrm{KL}(\pi^n | \pi^{n-1}). \tag{73}$$

In addition, the following results hold:

(a) $(\|\pi^{n+1} - \pi^n\|_{\mathrm{TV}})_{n \in \mathbb{N}}$ *and* $(\mathrm{J}(\pi^{n+1}, \pi^n))_{n \in \mathbb{N}}$ *are non-increasing.*

(b) $(\mathrm{KL}(\pi^{2n+1} | \pi^{2n}))_{n \in \mathbb{N}}$ *and* $(\mathrm{KL}(\pi^{2n+2} | \pi^{2n+1}))_{n \in \mathbb{N}}$ *are non-increasing.*

(c) $(\mathrm{KL}(\pi_1^{2n+1}|\nu_1))_{n\in\mathbb{N}}$ *and* $(\mathrm{KL}(\pi_0^{2n}|\nu_0))_{n\in\mathbb{N}}$ *are non-increasing.*

(d) $(\|\pi_1^{2n+1}-\nu_1\|_{\mathrm{TV}})_{n\in\mathbb{N}}$ *and* $(\|\pi_0^{2n}-\nu_0\|_{\mathrm{TV}})_{n\in\mathbb{N}}$ *are non-increasing.*

First, we show that under **B**1, the IPF sequence is well-defined and is associated with a sequence of potentials.

Proposition 32. *Assume* **B**1*. Then, the IPF sequence is well-defined and there exist* $(a_n)_{n\in\mathbb{N}}$ *and* $(b_n)_{n\in\mathbb{N}}$ *such that for any* $n\in\mathbb{N}$*,* $a_n, b_n : \mathbb{R}^d \to (0,+\infty)$ *and for any* $x, y \in \mathbb{R}^d$

$$\begin{aligned}(\mathrm{d}\pi^{2n+1}/\mathrm{d}(\mu_0\otimes\mu_1))(x,y) &= a_n(x)h(x,y)b_n(y)\\ (\mathrm{d}\pi^{2n+2}/\mathrm{d}(\mu_0\otimes\mu_1))(x,y) &= a_{n+1}(x)h(x,y)b_n(y),\end{aligned} \tag{74}$$

and

$$v_0(x) = a_{n+1}(x)\textstyle\int_{\mathbb{R}^d} h(x,y)b_n(y)\mathrm{d}\mu_1(y), \quad v_1(y) = b_n(y)\int_{\mathbb{R}^d} h(x,y)a_n(x)\mathrm{d}\mu_0(x), \tag{75}$$

where $v_i = \mathrm{d}\nu_i/\mathrm{d}\mu_i$ *for* $i\in\{0,1\}$*.*

Proof. First, we show that the IPF sequence is well-defined. Note that π^1 is well-defined since $\mathrm{KL}(\nu_0\otimes\nu_1|\mu) < +\infty$. Assume that $\{\pi^\ell\}_{\ell=1}^n$ is well-defined. Using (Csiszár, 1975, Theorem 2.2) we have

$$\mathrm{KL}(\nu_0\otimes\nu_1|\mu) = \mathrm{KL}(\nu_0\otimes\nu_1|\pi^n) + \textstyle\sum_{\ell=0}^{n-1}\mathrm{KL}(\pi^{\ell+1}|\pi^\ell).$$

In particular, $\mathrm{KL}(\nu_0\otimes\nu_1|\pi^n) < +\infty$ and π^{n+1} is well-defined. We conclude by recursion.

Using (Csiszár, 1975, Theorem 3.1) and **B**1, there exists $(\tilde{b}_n)_{n\in\mathbb{N}}$ such that for any $n\in\mathbb{N}$, $\tilde{b}_n : \mathbb{R}^d \to [0,+\infty)$ and for any $x, y \in \mathsf{A}_n$, $(\mathrm{d}\pi^{2n+1}/\mathrm{d}\pi^{2n})(x,y) = \tilde{b}_n(y)$ with $\mathsf{A}_n \in \mathcal{B}(\mathbb{R}^d)$, $\tilde{\pi}(\mathsf{A}_n) = 0$ for any $\tilde{\pi}$ such that $\tilde{\pi}_1 = \nu_1$ and $\mathrm{KL}(\tilde{\pi}|\pi^{2n}) < +\infty$. In particular we have $(\nu_0\otimes\nu_1)(\mathsf{A}_n) = 0$. Since ν_i is equivalent to μ_i for any $i\in\{0,1\}$ we have $(\mu_0\otimes\mu_1)(\mathsf{A}_n) = 0$. Similarly, there exists $(\tilde{a}_n)_{n\in\mathbb{N}}$ such that for any $n\in\mathbb{N}$, $\tilde{a}_n : \mathbb{R}^d \to [0,+\infty)$ and for any $x, y \in \mathsf{B}_n$, $(\mathrm{d}\pi^{2n+2}/\mathrm{d}\pi^{2n+1})(x,y) = \tilde{a}_{n+1}(x)$ with $\mathsf{B}_n \in \mathcal{B}(\mathbb{R}^d)$ and $(\mu_0\otimes\mu_1)(\mathsf{B}_n) = 0$. As a result, there exist $(a_n)_{n\in\mathbb{N}}$ and $(b_n)_{n\in\mathbb{N}}$ with $a_n : \mathbb{R}^d \to [0,+\infty)$ and $b_n : \mathbb{R}^d \to [0,+\infty)$ such that for any $n\in\mathbb{N}$ and $x, y \in \mathbb{R}^d$

$$\begin{aligned}(\mathrm{d}\pi^{2n+1}/\mathrm{d}(\mu_0\otimes\mu_1))(x,y) &= a_n(x)h(x,y)b_n(y)\\ (\mathrm{d}\pi^{2n+2}/\mathrm{d}(\mu_0\otimes\mu_1))(x,y) &= a_{n+1}(x)h(x,y)b_n(y),\end{aligned}$$

where $h = \mathrm{d}\mu/\mathrm{d}(\mu_0\otimes\mu_1)$ and $a_0 = 1$. In addition, setting $b_{-1} = 1$, we have for any $x, y \in \mathbb{R}^d$,

$$(\mathrm{d}\pi^0/\mathrm{d}(\mu_0\otimes\mu_1))(x,y) = a_0(x)h(x,y)b_{-1}(y).$$

Using that ν_i is absolutely continuous w.r.t. μ_i for $i\in\{0,1\}$ with density $v_i : \mathbb{R}^d \to (0,+\infty)$ we get that for any $x, y \in \mathbb{R}^d$ and $n\in\mathbb{N}$

$$v_0(x) = a_{n+1}(x)\textstyle\int_{\mathbb{R}^d} h(x,y)b_n(y)\mathrm{d}\mu_1(y), \quad v_1(y) = b_n(y)\int_{\mathbb{R}^d} h(x,y)a_n(x)\mathrm{d}\mu_0(x).$$

Since $v_0, v_1 > 0$ for any $n\in\mathbb{N}$, $a_n, b_n > 0$. □

Note that the system of equations (75) corresponds to iteratively solving the Schrödinger system, see Léonard (2014b) for a survey. In addition, (75) has connections with Fortet's mapping (Léonard, 2019; Fortet, 1940).

In the rest of the section we detail the proof of Proposition 4. We start by deriving identities between the marginals of the IPF and its joint distribution both w.r.t. the Kullback-Leibler divergence and the total variation norm in Lemma 33. Second, we establish that $(\|\pi^{n+1}-\pi^n\|_{\mathrm{TV}})_{n\in\mathbb{N}}$ is non-increasing in Lemma 34. Then, we prove (73) in Lemma 35. We conclude with the proof of Proposition 31.

Lemma 33. *Assume* **B**1*. Then, for any* $n\in\mathbb{N}$ *we have*

$$\|\pi^{2n+1}-\pi^{2n}\|_{\mathrm{TV}} = \|\pi_1^{2n}-\nu_1\|_{\mathrm{TV}}, \qquad \|\pi^{2n+2}-\pi^{2n+1}\|_{\mathrm{TV}} = \|\pi_0^{2n+1}-\nu_0\|_{\mathrm{TV}}. \tag{76}$$

In addition, we have

$$\mathrm{KL}(\pi^{2n}|\pi^{2n+1}) = \mathrm{KL}(\pi_1^{2n}|\nu_1), \qquad \mathrm{KL}(\pi^{2n+1}|\pi^{2n+2}) = \mathrm{KL}(\pi_0^{2n+1}|\nu_0). \tag{77}$$

Proof. We divide the proof into two parts. First, we prove (76). Second, we show that (77) holds.

(a) We only show that for any $n \in \mathbb{N}$ we have $\|\pi^{2n+1} - \pi^{2n}\|_{\mathrm{TV}} = \|\pi_1^{2n} - \nu_1\|_{\mathrm{TV}}$. The proof that for any $n \in \mathbb{N}$, $\|\pi^{2n+2} - \pi^{2n+1}\|_{\mathrm{TV}} = \|\pi_0^{2n+1} - \nu_0\|_{\mathrm{TV}}$ is similar. Let $n \in \mathbb{N}$. Using (74) and (75) we have

$$\begin{aligned}\|\pi^{2n+1} - \pi^{2n}\|_{\mathrm{TV}} &= \int_{(\mathbb{R}^d)^2} |b_n(y) - b_{n-1}(y)|\, a_n(x) h(x,y) \mathrm{d}\mu_0(x) \mathrm{d}\mu_1(y) \\ &= \int_{\mathbb{R}^d} |1 - b_{n-1}(x)/b_n(x)|\, \mathrm{d}\nu_1(y).\end{aligned} \tag{78}$$

In addition, we have that for any $\mathsf{A} \in \mathcal{B}(\mathbb{R}^d)$

$$\pi_1^{2n}(\mathsf{A}) = \int_{\mathbb{R}^d \times \mathsf{A}} a_n(x) b_{n-1}(y) h(x,y) \mathrm{d}\mu_0(x) \mathrm{d}\mu_1(y) = \int_{\mathsf{A}} (b_{n-1}/b_n)(y) \mathrm{d}\nu_1(y).$$

We get that for any $y \in \mathbb{R}^d$, $(\mathrm{d}\pi_1^{2n}/\mathrm{d}\nu_1)(y) = (b_{n-1}/b_n)(y)$. Hence, using (78) we get that

$$\|\pi_1^{2n} - \nu_1\|_{\mathrm{TV}} = \int_{\mathbb{R}^d} |1 - a_n(x)/a_{n+1}(x)|\, \mathrm{d}\nu_0(x) = \|\pi^{2n+1} - \pi^{2n}\|_{\mathrm{TV}}.$$

(b) We only show that for any $n \in \mathbb{N}$ we have $\mathrm{KL}(\pi^{2n}|\pi^{2n+1}) = \mathrm{KL}(\pi_1^{2n}|\nu_1)$. The proof that for any $n \in \mathbb{N}$, $\mathrm{KL}(\pi^{2n+1}|\pi^{2n+2}) = \mathrm{KL}(\pi_0^{2n+1}|\nu_0)$ is similar. Let $n \in \mathbb{N}$. Using that for any $x, y \in \mathbb{R}^d$, $(\mathrm{d}\pi_1^{2n}/\mathrm{d}\nu_1)(y) = b_{n-1}(y)/b_n(y)$ and that $(\mathrm{d}\pi^{2n+1}/\mathrm{d}\pi^{2n})(x,y) = b_n(y)/b_{n-1}(y)$ we have

$$\mathrm{KL}(\pi^{2n}|\pi^{2n+1}) = -\int_{\mathbb{R}^d} \log(b_n(y)/b_{n-1}(y)) \mathrm{d}\pi_1^{2n}(y) = \mathrm{KL}(\pi_1^{2n}|\nu_1).$$

This concludes the proof.

□

Lemma 34. *Assume* **B**1. *Then* $(\|\pi^{n+1} - \pi^n\|_{\mathrm{TV}})_{n \in \mathbb{N}}$ *is non-increasing.*

Proof. We only prove that for any $n \in \mathbb{N}$ with $n \geq 1$, $\|\pi^{2n+1} - \pi^{2n}\|_{\mathrm{TV}} \leq \|\pi^{2n} - \pi^{2n-1}\|_{\mathrm{TV}}$. The proof that for any $n \in \mathbb{N}$, $\|\pi^{2n+2} - \pi^{2n+1}\|_{\mathrm{TV}} \leq \|\pi^{2n+1} - \pi^{2n}\|_{\mathrm{TV}}$ is similar. Let $n \in \mathbb{N}$ with $n \geq 1$. Similarly to the proof of Lemma 33 we have that

$$\|\pi^{2n+1} - \pi^{2n}\|_{\mathrm{TV}} = \int_{\mathbb{R}^d} |1 - b_{n-1}(y)/b_n(y)|\, \mathrm{d}\nu_1(y) = \int_{\mathbb{R}^d} \left|b_n^{-1}(y) - b_{n-1}^{-1}(y)\right| b_{n-1}(y) \mathrm{d}\nu_1(y). \tag{79}$$

In addition, we have that for any $y \in \mathbb{R}^d$

$$\left|b_{n-1}^{-1}(y) - b_n^{-1}(y)\right| \leq v_1^{-1}(y) \int_{\mathbb{R}^d} h(x,y)\, |a_{n-1}(x) - a_n(x)|\, \mathrm{d}\mu_0(x).$$

Combining this result and (79) we get that

$$\begin{aligned}\|\pi^{2n+1} - \pi^{2n}\|_{\mathrm{TV}} &\leq \int_{\mathbb{R}^d} \left|b_{n-1}^{-1}(y) - b_n^{-1}(y)\right| b_{n-1}(y) \mathrm{d}\nu_1(y) \\ &\leq \int_{(\mathbb{R}^d)^2} |a_n(x) - a_{n-1}(x)|\, h(x,y) b_{n-1}(y) \mathrm{d}\mu_0(x) \mathrm{d}\mu_1(y) \\ &\leq \int_{\mathbb{R}^d} |1 - a_{n-1}(x)/a_n(x)|\, \mathrm{d}\nu_0(x) \leq \|\pi^{2n} - \pi^{2n-1}\|_{\mathrm{TV}},\end{aligned}$$

which concludes the proof. □

Lemma 35. *Assume* **B**1. *Then for any* $n \in \mathbb{N}$ *with* $n \geq 1$ *we have*

$$\mathrm{KL}(\pi^{n+1}|\pi^n) \leq \mathrm{KL}(\pi^{n-1}|\pi^n), \qquad \mathrm{KL}(\pi^n|\pi^{n+1}) \leq \mathrm{KL}(\pi^n|\pi^{n-1}).$$

Proof. Using Lemma 33 and the data processing theorem (Ambrosio et al., 2008, Lemma 9.4.5) we get that for any $n \in \mathbb{N}$

$$\mathrm{KL}(\pi^{2n}|\pi^{2n+1}) = \mathrm{KL}(\pi_1^{2n}|\nu_1) \leq \mathrm{KL}(\pi^{2n}|\pi^{2n+1}).$$

Similarly, we get that for any $n \in \mathbb{N}$, $\mathrm{KL}(\pi^{2n+1}|\pi^{2n+2}) \leq \mathrm{KL}(\pi^{2n+1}|\pi^{2n})$. Hence, we get that for any $n \in \mathbb{N}$, $\mathrm{KL}(\pi^n|\pi^{n+1}) \leq \mathrm{KL}(\pi^n|\pi^{n-1})$.

In addition, using that for any $n \in \mathbb{N}$ with $n \geq 1$ and $x, y \in \mathbb{R}^d$, we have that $\pi_1^{2n+1} = \nu_1$ and $(\mathrm{d}\pi^{2n+1}/\mathrm{d}\pi^{2n})(x,y) = b_n(y)/b_{n-1}(y)$ we get for any $n \in \mathbb{N}$ with $n \geq 1$

$$\mathrm{KL}(\pi^{2n+1}|\pi^{2n}) = -\int_{\mathbb{R}^d} \log(b_{n-1}(y)/b_n(y)) \mathrm{d}\nu_1(y). \tag{80}$$

Using Jensen's inequality we have for any $n \in \mathbb{N}$

$$-\log(b_{n-1}(y)/b_n(y)) \leq -\log\left(\int_{\mathbb{R}^d} h(x,y) a_n(x) \mathrm{d}\mu_0(x) \Big/ \int_{\mathbb{R}^d} h(x,y) a_{n-1}(x) \mathrm{d}\mu_0(x)\right)$$

$$\leq -\log\left(\int_{\mathbb{R}^d}(a_n(x)/a_{n-1}(x))h(x,y)a_{n-1}(x)\mathrm{d}\mu_0(y)/\int_{\mathbb{R}^d}h(x,y)a_{n-1}(x)\mathrm{d}\mu_0(x)\right)$$
$$\leq -\int_{\mathbb{R}^d}\log(a_n(x)/a_{n-1}(x))b_{n-1}(y)h(x,y)a_{n-1}(x)/v_1(y)\mathrm{d}\mu_0(x).$$

Combining this result, (80), Fubini's theorem and that for any $n \in \mathbb{N}$ with $n \geq 1$ and $x \in \mathbb{R}^d$, $(\mathrm{d}\pi_0^{2n-1}/\mathrm{d}\nu_0)(x) = a_{n-1}(x)/a_n(x)$ we get that for any $n \in \mathbb{N}$ with $n \geq 1$

$$\mathrm{KL}(\pi^{2n+1}|\pi^{2n}) \leq \int_{(\mathbb{R}^d)^2}\log(a_{n-1}(x)/a_n(x))a_{n-1}(x)h(x,y)b_{n-1}(y)\mathrm{d}\mu_1(y)\mathrm{d}\mu_0(x)$$
$$\leq \int_{(\mathbb{R}^d)^2}\log(a_{n-1}(x)/a_n(x))(a_{n-1}(x)/a_n(x))\mathrm{d}\nu_0(x) \leq \mathrm{KL}(\pi_0^{2n-1}|\nu_0).$$

Using Lemma 33 (or the data processing theorem) we get that for any $n \in \mathbb{N}$ with $n \geq 1$, $\mathrm{KL}(\pi^{2n+1}|\pi^{2n}) \leq \mathrm{KL}(\pi^{2n-1}|\pi^{2n})$. Similarly, we get that for any $n \in \mathbb{N}$, $\mathrm{KL}(\pi^{2n+2}|\pi^{2n+1}) \leq \mathrm{KL}(\pi^{2n}|\pi^{2n+1})$, which concludes the proof. □

We now turn to the proof of Proposition 31

Proof. First, (73) is a direct consequence of Lemma 35. Using Lemma 34 we get that $(\|\pi^{n+1} - \pi^n\|_{\mathrm{TV}})_{n\in\mathbb{N}}$ is non-increasing. Since for any $\eta_0, \eta_1 \in \mathscr{P}(\mathbb{R}^d)$ we have $\mathrm{J}(\eta_0, \eta_1) = (1/2)\{\mathrm{KL}(\eta_0|\eta_1) + \mathrm{KL}(\eta_0|\eta_1)\}$ and using (73), we get that $(\mathrm{J}(\pi_{n+1}, \pi_n))_{n\in\mathbb{N}}$ is non-increasing which proves Proposition 31-(a). Proposition 31-(b) is a straightforward consequence of (73). Proposition 31-(c) is a consequence of Lemma 33 and Proposition 31-(a). Finally, Proposition 31-(c) is a consequence of Lemma 33 and (73). □

Note that we also have that for any $n \in \mathbb{N}$, $(\mathrm{KL}(\pi^{2n}|\pi^{2n+1}))_{n\in\mathbb{N}}$ and $(\mathrm{KL}(\pi^{2n+1}|\pi^{2n+2}))_{n\in\mathbb{N}}$ are non-increasing.

F.1.2 Quantitative convergence bounds

In this section we prove the following theorem.

Theorem 36. *Assume* **B**1. *Then, the IPF sequence* $(\pi^n)_{n\in\mathbb{N}}$ *is well-defined and there exists a probability measure* π^∞ *such that* $\lim_{n\to+\infty}\|\pi^n - \pi^\infty\|_{\mathrm{TV}} = 0$ *and the following hold:*

(a) $\lim_{n\to+\infty} n^{1/2}\{\|\pi_0^n - \nu_0\|_{\mathrm{TV}} + \|\pi_1^n - \nu_1\|_{\mathrm{TV}}\} = 0.$

(b) $\lim_{n\to+\infty} n\{\mathrm{KL}(\pi_0^n|\nu_0) + \mathrm{KL}(\pi_1^n|\nu_1)\} = 0.$

We begin with Lemma 37 which is an adaption of (Ruschendorf et al., 1995, Proposition 2.1). Then we state and prove Lemma 38 which is a classical lemma from real analysis. Combining these two lemmas and the monotonicity results from Proposition 31 conclude the proof.

Lemma 37. *Assume* **B**1. *Then,* $(\pi^n)_{n\in\mathbb{N}}$ *is well-defined and we have* $\sum_{n\in\mathbb{N}}\mathrm{KL}(\pi^{n+1}|\pi^n) < +\infty$.

Proof. The sequence is well-defined using Proposition 32. In addition, using (Csiszár, 1975, Theorem 2.2) we have for any $n \in \mathbb{N}$

$$\mathrm{KL}(\mu^\star|\pi^0) = \mathrm{KL}(\pi^\star|\pi^n) + \textstyle\sum_{k=0}^{n-1}\mathrm{KL}(\pi^{k+1}|\pi^k),$$

which concludes the proof. □

Lemma 38. *Let* $(c_n)_{n\in\mathbb{N}} \in [0,+\infty)^{\mathbb{N}}$ *a non-increasing sequence such that* $\sum_{n\in\mathbb{N}} c_n < +\infty$. *Then* $\lim_{n\to+\infty} c_n n = 0$.

Proof. Let $\varepsilon > 0$ and $n_0 \in \mathbb{N}$ such that for any $n \geq n_0$, $\sum_{k=n}^{+\infty} c_k \leq \varepsilon$. Let $n \in \mathbb{N}$ with $n \geq 2n_0$. Note that $n - n_0 \geq n/2 \geq n_0$. Therefore we have $\varepsilon \geq (n - n_0)c_n \geq (n/2)c_n$. Hence, for any $n \in \mathbb{N}$ with $n \geq 2n_0$, $c_n n \leq 2\varepsilon$, which concludes the proof. □

We now conclude with the proof of Theorem 36.

Proof. Using Lemma 37 and Pinsker's inequality (Bakry et al., 2014, Equation 5.2.2) we have $\sum_{n\in\mathbb{N}} \|\pi^{n+1} - \pi^n\|_{\mathrm{TV}} < +\infty$. For any $N \in \mathbb{N}$, let $S_N = \sum_{n=0}^{N} \pi^{n+1} - \pi^n = \pi^{N+1} - \mu$. Since the space of finite signed measures endowed with $\|\cdot\|_{\mathrm{TV}}$ is a Banach space (Douc et al., 2019, Theorem D.2.7) we have that $(S_N)_{N\in\mathbb{N}}$ converges. Hence there exists a finite signed measure π^∞ such that $\lim_{n\to+\infty} \|\pi^n - \pi^\infty\|_{\mathrm{TV}} = 0$. π^∞ is a probability measure since for any $n \in \mathbb{N}$, π^n is a probability measure.

In addition, since $(\mathrm{KL}(\pi^{2n+1}|\pi^{2n}))_{n\in\mathbb{N}}$ and $(\mathrm{KL}(\pi^{2n+2}|\pi^{2n+1}))_{n\in\mathbb{N}}$ are non-increasing by Proposition 31, using Lemma 38, we get that

$$\lim_{n\to+\infty} n \left\{\mathrm{KL}(\pi_0^n|\nu_0) + \mathrm{KL}(\pi_1^n|\nu_1)\right\} = 0.$$

We conclude upon using Pinsker's inequality (Bakry et al., 2014, Equation 5.2.2). □

F.2 Proof of Proposition 5

Similarly to Appendix F.1, we consider the static IPF recursion: $\pi^0 = \mu \in \mathscr{P}_2$ and

$$\pi^{2n+1} = \arg\min \left\{\mathrm{KL}(\pi|\pi^{2n}) \,:\, \pi \in \mathscr{P}_2,\ \pi_1 = \nu_1\right\},$$
$$\pi^{2n+2} = \arg\min \left\{\mathrm{KL}(\pi|\pi^{2n+1}) \,:\, \pi \in \mathscr{P}_2,\ \pi_0 = \nu_0\right\},$$

where $\nu_0, \nu_1 \in \mathscr{P}(\mathbb{R}^d)$. We recall that in this context that if the Schrödinger bridge $\pi^\star$ exists it is given by

$$\pi^\star = \arg\min\{\mathrm{KL}(\pi|\mu) \,:\, \pi \in \mathscr{P}_2,\ \pi_0 = \nu_0,\ \pi_1 = \nu_1\}.$$

In this section, we prove the following proposition which directly implies Proposition 5.

Proposition 39. *Assume* **B**1 *and denote* $h = \mathrm{d}\mu/(\mathrm{d}\mu_0 \otimes \mu_1)$. *Assume that* $h \in \mathrm{C}(\mathbb{R}^d \times \mathbb{R}^d, (0, +\infty])$ *and that there exist* $\Phi_0, \Phi_1 \in \mathrm{C}(\mathbb{R}^d, (0, +\infty))$ *such that for any* $x, y \in \mathbb{R}^d$

$$h(x, y) \leq \Phi_0(x)\Phi_1(y), \text{ and}$$
$$\textstyle\int_{\mathbb{R}^d\times\mathbb{R}^d} (|\log h(x_0, x_1)| + |\log \Phi_0(x_0)| + |\log \Phi_1(x_1)|)\mathrm{d}\mu_0(x_0)\mathrm{d}\mu_1(x_1) < +\infty. \tag{81}$$

Then there exists a solution $\pi^\star$ *to the Schrödinger bridge and the IPF sequence satisfies* $\lim_{n\to+\infty} \|\pi^n - \pi^\infty\|_{\mathrm{TV}} = 0$ *with* $\pi^\infty \in \mathscr{P}_2$. *If* μ *is absolutely continuous w.r.t.* π^∞ *then* $\pi^\infty = \pi^\star$.

We begin with an adaptation of (Rüschendorf and Thomsen, 1993, Proposition 2).

Proposition 40. *Let* $\mu \in \mathscr{P}_2$ *and assume that* μ *is absolutely continuous w.r.t.* $\mu_0 \otimes \mu_1$. *Let* $(a_n)_{n\in\mathbb{N}}$ *and* $(b_n)_{n\in\mathbb{N}}$ *such that for any* $n \in \mathbb{N}$, $a_n : \mathbb{R}^d \to (0, +\infty)$ *and* $b_n : \mathbb{R}^d \to (0, +\infty)$. *Assume that there exists* $\Phi : (\mathbb{R}^d)^2 \to [0, +\infty)$ *and* $\mathsf{A} \in \mathcal{B}(\mathbb{R}^d) \otimes \mathcal{B}(\mathbb{R}^d)$ *with* $\mu(\mathsf{A}) = 1$ *such that for any* $(x, y) \in \mathsf{A}$

$$\lim_{n\to+\infty} a_n(x)b_n(y) = \Phi(x, y).$$

Then, there exist $a : \mathbb{R}^d \to [0, +\infty)$, $b : \mathbb{R}^d \to [0, +\infty)$ *and* $\mathsf{B} \in \mathcal{B}(\mathbb{R}^d) \otimes \mathcal{B}(\mathbb{R}^d)$ *with* $\mu(\mathsf{B}) = 1$ *such that for any* $x, y \in \mathsf{B}$

$$\Phi(x, y) = a(x)b(y), \qquad \text{or} \qquad \Phi(x, y) = 0.$$

Proof. Let $\tilde{\mathsf{A}} = \{(x, y) \in (\mathbb{R}^d)^2 \,:\, \Phi(x, y) = 0\}$ and $\mathsf{A}_a = \tilde{\mathsf{A}} \cap \mathsf{A}$ and $\mathsf{A}_b = \tilde{\mathsf{A}}^{\mathrm{c}} \cap \mathsf{A}$. If $\mathsf{A}_b = \emptyset$, we conclude the proof. Otherwise, let $(x_0, y_0) \in \mathsf{A}_b$. Let $\mathsf{C}_0, \mathsf{C}_1 \in \mathcal{B}(\mathbb{R}^d) \otimes \mathcal{B}(\mathbb{R}^d)$ be given by

$$\mathsf{C}_0^0 = \{x \in \mathbb{R}^d \,:\, \lim_{n\to+\infty} a_n^0(x) = a^0(x) \text{ exists and } a^0(x) > 0\}, \tag{82}$$
$$\mathsf{C}_1^0 = \{y \in \mathbb{R}^d \,:\, \lim_{n\to+\infty} b_n^0(y) = b^0(y) \text{ exists and } b^0(y) > 0\},$$

where for any $n \in \mathbb{N}$ and $x, y \in \mathbb{R}^d$, $a_n^0(x) = a_n(x)/a_n(x_0)$ and $b_n^0(y) = b_n(y)a_n(x_0)$, which is well-defined since for any $n \in \mathbb{N}$, $a_n(x_0) > 0$. Note that $x_0 \in \mathsf{C}_0^0$ and that $y_0 \in \mathsf{C}_1^0$. If $\mathsf{A}_b \subset \mathsf{C}_0^0 \times \mathsf{C}_1^0$, we conclude the proof. Otherwise, let $(x_1, y_1) \in \mathsf{A}_b \cap (\mathsf{C}_0^0 \times \mathsf{C}_1^0)^{\mathrm{c}}$ and define

$$\mathsf{C}_0^1 = \{x \in \mathbb{R}^d \,:\, \lim_{n\to+\infty} a_n^1(x) = a^1(x) \text{ exists and } a^1(x) > 0\},$$

$$\mathsf{C}_1^1 = \{y \in \mathbb{R}^d \,:\, \lim_{n\to+\infty} b_n^1(y) = b^1(y) \text{ exists and } b^1(y) > 0\},$$

where for any $n \in \mathbb{N}$ and $x, y \in \mathbb{R}^d$, $a_n^1(x) = a_n(x)/a_n(x_1)$ and $b_n^1(y) = b_n(y)a_n(x_1)$, which is well-defined since for any $n \in \mathbb{N}$, $a_n(x_1) > 0$. Note that $\mathsf{C}_0^0 \cap \mathsf{C}_0^1 = \emptyset$ and $\mathsf{C}_1^0 \cap \mathsf{C}_1^1 = \emptyset$. Indeed, if there exists $x \in \mathsf{C}_0^0 \cap \mathsf{C}_0^1$, then $a^0(x) = \lim_{n\to+\infty} a_n(x)/a_n(x_0) > 0$ and $a^1(x) = \lim_{n\to+\infty} a_n(x)/a_n(x_1) > 0$ exists. Therefore $\lim_{n\to+\infty} a_n(x_1)/a_n(x_0) > 0$ exists and $\lim_{n\to+\infty} b_n(y_1)a_n(x_0) > 0$ exists. Hence $(x_1, y_1) \in \mathsf{C}_0^0 \times \mathsf{C}_1^0$ which is absurd. Similarly, if there exists $y \in \mathsf{C}_1^0 \cap \mathsf{C}_1^1$ then $(x_1, y_1) \in \mathsf{C}_0^0 \times \mathsf{C}_1^0$ which is absurd. Hence, we consider $T : \mathsf{A}_b \to 2^{(\mathbb{R}^d)^2}$ such that for any $(x, y) \in \mathsf{A}_b$, $T(x, y) = \mathsf{C}_0^{(x,y)} \times \mathsf{C}_1^{(x,y)}$, where $\mathsf{C}_0^{(x,y)} \times \mathsf{C}_1^{(x,y)}$ is constructed as in (82) replacing (x_0, y_0) by (x, y).

Consider a well order on $(\mathsf{A}_b, \leq)$, which is possible by the well-ordering principle (Enderton, 1977, p. 196). For any $(x, y) \in \mathbb{R}^d$, let $\mathsf{A}_b^{(x,y)} = \{(x', y') \in (\mathbb{R}^d)^2 \,:\, (x', y') < (x, y)\}$. Using the transfinite recursion theorem (Enderton, 1977, p. 175) there exists $f : \mathsf{A}_b \to \{0, 1\}$ such that for any $(x, y) \in \mathsf{A}_b$ if there exists $(x', y') \in (\mathbb{R}^d)^2$ such that $(x', y') < (x, y)$, $f(x', y') = 1$ and $(x, y) \in T(x', y')$ then $f(x, y) = 0$ and $f(x, y) = 1$ otherwise. Let $I = f^{-1}(\{1\})$. Let $(x, y), (x', y) \in I$ with $(x, y) \neq (x', y')$ then for $(x, y) < (x', y')$ for instance. Since $f(x, y) = f(x', y') = 1$ we have that $(\mathsf{C}_0^{(x,y)} \times \mathsf{C}_1^{(x,y)}) \cap (\mathsf{C}_0^{(x',y')} \times \mathsf{C}_1^{(x',y')}) = \emptyset$. Let $(x, y) \in \mathsf{A}_b$. If $f(x, y) = 1$ then $(x, y) \in \mathsf{C}_0^{(x,y)} \times \mathsf{C}_1^{(x,y)}$. If $f(x, y) = 0$ then there exists $(x', y') < (x, y)$ such that $(x, y) \in \mathsf{C}_0^{(x',y')} \times \mathsf{C}_1^{(x',y')}$. Therefore, we get that $\{\mathsf{C}^{(x,y)} = (\mathsf{C}_0^{(x,y)} \times \mathsf{C}_1^{(x,y)}) \cap \mathsf{A}_b \,:\, (x, y) \in I\}$ is a partition of A_b.

Since $\mu(\mathsf{A}_b) \leq 1$, and $\{\mathsf{C}^{(x,y)} = (\mathsf{C}_0^{(x,y)} \times \mathsf{C}_1^{(x,y)}) \cap \mathsf{A}_b \,:\, (x, y) \in I\}$ is a partition of A_b, we get that $J = \{\mathsf{C}^{(x,y)} \,:\, (x, y) \in I, \mu_0(\mathsf{C}_0^{(x,y)})\mu_1(\mathsf{C}_1^{(x,y)}) > 0\}$ is countable. Denote $\mathsf{A}_c = \cup_{(x,y)\in J} \mathsf{C}^{(x,y)}$. Let us show that $\mu(\mathsf{A}_c^{\mathrm{c}} \cap \mathsf{A}_b) = \mu(\cup_{(x,y)\in I\cap J^{\mathrm{c}}} \mathsf{C}^{(x,y)}) = 0$. Let $x \in \mathbb{R}^d$ and define $\mathsf{D}_x = \{y \in \mathbb{R}^d \,:\, (x, y) \in \mathsf{A}_b \cap \mathsf{A}_c^{\mathrm{c}}\}$. If D_x is not empty, then there exists $(x', y') \in I$ such that $x \in \mathsf{C}_0^{(x',y')}$. Then, for any $y \in \mathsf{D}_x$, $y \in \mathsf{C}_1^{(x',y')}$. Hence, $(x', y') \in I \cap J^{\mathrm{c}}$ by definition of D_x and $\mu_1(\mathsf{D}_x) = 0$. We get that

$$\mu(\mathsf{A}_b \cap \mathsf{A}_c^{\mathrm{c}}) = \int_{\mathbb{R}^d} \left(\int_{\mathsf{D}_x} h(x, y) \mathrm{d}\mu_1(y) \right) \mathrm{d}\mu_0(x) = 0,$$

where h is the density of μ w.r.t. $\mu_0 \otimes \mu_1$. Note that this is the only instance in the proof, where we use that μ is absolutely continuous w.r.t. $\mu_0 \otimes \mu_1$. For any $(x, y) \in \mathsf{A}_c$ define for any $n \in \mathbb{N}$

$$\hat{a}_n(x) = \textstyle\sum_{(x',y')\in J} \mathbb{1}_{\mathsf{C}_0^{(x',y')}}(x) a_n^{(x',y')}(x), \quad \hat{b}_n(y) = \textstyle\sum_{(x',y')\in J} \mathbb{1}_{\mathsf{C}_1^{(x',y')}}(x) b_n^{(x',y')}(y).$$

There exist $\hat{a}, \hat{b} : \mathbb{R}^d \to (0, +\infty)$ such that for any $(x, y) \in \mathsf{A}_c$, $\lim_{n\to+\infty} \hat{a}_n(x) = \hat{a}(x)$ and $\lim_{n\to+\infty} \hat{b}_n(y) = \hat{b}(y)$. In addition, for any $(x, y) \in \mathsf{A}_c$, $a_n(x)b_n(y) = \hat{a}_n(x)\hat{b}_n(y)$. Hence, for any $(x, y) \in \mathsf{A}_c$, $\Phi(x, y) = \hat{a}(x)\hat{b}(y)$. Since $\mathsf{A}_a \cap \mathsf{A}_c = \emptyset$ and $\mu(\mathsf{A}_c) = \mu(\mathsf{A}_b)$, we have

$$\mu(\mathsf{A}_a) + \mu(\mathsf{A}_c) = \mu(\mathsf{A}_a) + \mu(\mathsf{A}_b) = \mu(\mathsf{A}) = 1.$$

We conclude the proof upon remarking that for any $(x, y) \in \mathsf{A}_a$, $\Phi(x, y) = 0$ and for any $(x, y) \in \mathsf{A}_c$, $\Phi(x) = \hat{a}(x)\hat{b}(y)$. □

In what follows we prove Proposition 39.

Proof. Since $\lim_{n\to+\infty} \|\pi^n - \pi^\infty\|_{\mathrm{TV}} = 0$ by Theorem 36 and $\mathrm{KL}(\pi^\infty|\mu) < +\infty$, there exist A with $\mu(\mathsf{A}) = 1$ and $\Phi : (\mathbb{R}^d)^2 \to [0, +\infty)$ such that, up to extraction, for any $x, y \in \mathsf{A}$

$$\lim_{n\to+\infty} a_n(x)b_n(y) = \Phi(x, y),$$

and $(\mathrm{d}\pi^\infty/\mathrm{d}\mu) = \Phi$. Using Proposition 40, there exist $a, b : \mathbb{R}^d \to [0, +\infty)$ and B with $\pi^\infty(\mathsf{B}) = 1$ such that for any $x, y \in \mathsf{B}$, $(\mathrm{d}\pi^\infty/\mathrm{d}\mu)(x, y) = a(x)b(y)$. Since μ is absolutely continuous w.r.t. π^∞, we get that for any $x, y \in \mathbb{R}^d$, $(\mathrm{d}\pi^\infty/\mathrm{d}(\mu_0 \otimes \mu_1))(x, y) = a(x)b(y)h(x, y)$. In addition, the Schrödinger bridge $\pi^\star \in \mathscr{P}((\mathbb{R}^d)^2)$ exists, see (Rüschendorf and Thomsen, 1993, Theorem 3), and there exist $a', b' : \mathbb{R}^d \to [0, +\infty)$ and B' with $\mu(\mathsf{B}') = 1$ such that for any $x, y \in \mathsf{B}'$

$$(\mathrm{d}\pi^\star/\mathrm{d}(\mu_0 \otimes \mu_1))(x, y) = a'(x)b'(y)h(x, y).$$

Let $\mathscr{M}_{+,\times}$ be the space of non-negative product measures over $\mathcal{B}(\mathbb{R}^d) \otimes \mathcal{B}(\mathbb{R}^d)$. Let $\mathbf{\Psi}_{\bar{h}} : \mathscr{M}_{+,\times} \to \mathscr{M}_{+,\times}$ be given for any $\lambda = \lambda_0 \otimes \lambda_1 \in \mathscr{M}_{+,\times}$ by $\mathbf{\Psi}_{\bar{h}}(\lambda) = \Psi_h^\lambda$ where for any $\mathsf{A}, \mathsf{B} \in \mathcal{B}(\mathbb{R}^d)$

$$\Psi_h^\lambda(\mathsf{A} \times \mathsf{B}) = (\int_{\mathsf{A} \times \mathbb{R}^d} \bar{h}(x,y) \mathrm{d}\lambda_0(x) \mathrm{d}\lambda_1(x))(\int_{\mathbb{R}^d \times \mathsf{B}} \bar{h}(x,y) \mathrm{d}\lambda_0(x) \mathrm{d}\lambda_1(y))$$

where for any $x, y \in \mathbb{R}^d$, $\bar{h}(x,y) = h(x,y)\Phi_0^{-1}(x)\Phi_1^{-1}(y)$. Note that $\bar{h} \in \mathrm{C}(\mathbb{R}^d \times \mathbb{R}^d, [0, +\infty))$ and is bounded. Hence, using (Beurling, 1960, Theorem 2) and (81) we get that $\mathbf{\Psi}_{\bar{h}}$ is a bijection. Let $\lambda = (a\Phi_0\mu_0, b\Phi_1\mu_1)$ and $\lambda' = (a'\Phi_0\mu_0, b'\Phi_1\mu_1)$. Then, since $\pi_i^\star = \pi_i^\infty = \nu_i$ for $i \in \{0,1\}$ we get that $\mathbf{\Psi}_h(\lambda) = \mathbf{\Psi}_h(\lambda')$. Hence $\lambda = \lambda'$ and $\pi^\infty = \pi^\star$ which concludes the proof. □

In Proposition 42 we derive an alternative proposition to Proposition 39. We start with the following lemma.

Lemma 41. *Let $\pi^\star \in \mathscr{P}_2$ with $\pi_i^\star = \nu_i$ for $i \in \{0,1\}$. Assume that $\mathrm{KL}(\pi^\star|\mu) < +\infty$ and that $\mathrm{L}^1(\nu_0) \oplus \mathrm{L}^1(\nu_1)$ is closed in $\mathrm{L}^1(\pi^\star)$. In addition, assume that there exist $a, b : \mathbb{R}^d \to [0, +\infty)$ and A with $\pi^\star(\mathsf{A}) = 1$ such that for any $(x,y) \in \mathsf{A}$,*

$$(\mathrm{d}\pi^\star/\mathrm{d}\mu)(x,y) = a(x)b(y).$$

Then $\pi^\star$ is the Schrödinger bridge.

Proof. Since $\mathrm{KL}(\pi^\star|\mu) < +\infty$ we have that

$$\int_{(\mathbb{R}^d)^2} |\log(a(x)b(y))| \, \mathrm{d}\pi^\star(x,y) < +\infty.$$

Using (Kober, 1939, Theorem 1) and that $\pi_i^\star = \nu_i$ for $i \in \{0,1\}$, we get that

$$\int_{\mathbb{R}^d} |\log a(x)| \, \mathrm{d}\nu_0(x) + \int_{\mathbb{R}^d} |\log b(y)| \, \mathrm{d}\nu_1(y) < +\infty. \tag{83}$$

Let $\pi \in \mathscr{P}_2$ such that $\pi_i = \nu_i$ for $i \in \{1,2\}$ and $\mathrm{KL}(\pi|\mu) < +\infty$. Using (83), we have that $\int_{(\mathbb{R}^d)^2} |\log((\mathrm{d}\pi^\star/\mathrm{d}\mu)(x,y))| \, \mathrm{d}\pi(x,y) < +\infty$. Hence, $(\mathrm{d}\pi^\star/\mathrm{d}\mu) > 0$, π-almost surely. Using this result we have for any $\mathsf{A} \in \mathcal{B}(\mathbb{R}^d)$

$$\begin{aligned}
\pi[\mathsf{A}] &= \textstyle\int_{\mathbb{R}^d} \mathbb{1}_{\mathsf{A}}(x)(\mathrm{d}\pi^\star/\mathrm{d}\mu)(x)(\mathrm{d}\pi^\star/\mathrm{d}\mu)(x)^{-1}\mathrm{d}\pi(x) \\
&= \textstyle\int_{\mathbb{R}^d} \mathbb{1}_{\mathsf{A}}(x)(\mathrm{d}\pi^\star/\mathrm{d}\mu)(x)(\mathrm{d}\pi^\star/\mathrm{d}\mu)(x)^{-1}(\mathrm{d}\pi/\mathrm{d}\mu)(x)\mathrm{d}\mu(x) \\
&= \textstyle\int_{\mathbb{R}^d} \mathbb{1}_{\mathsf{A}}(x)(\mathrm{d}\pi^\star/\mathrm{d}\mu)(x)^{-1}(\mathrm{d}\pi/\mathrm{d}\mu)(x)\mathrm{d}\pi^\star(x).
\end{aligned}$$

Hence we get that $\mathrm{d}\pi/\mathrm{d}\pi^\star = (\mathrm{d}\pi/\mathrm{d}\mu)(\mathrm{d}\pi^\star/\mathrm{d}\mu)^{-1}$. In addition, we have that

$$\mathrm{KL}(\pi^\star|\mu) = \textstyle\int_{\mathbb{R}^d} \log(a(x))\mathrm{d}\nu_0(x) + \int_{\mathbb{R}^d} \log(b(y))\mathrm{d}\nu_1(y) = \int_{(\mathbb{R}^d)^2} \log((\mathrm{d}\pi^\star/\mathrm{d}\mu)(x,y))\mathrm{d}\pi(x,y).$$

We get that

$$\mathrm{KL}(\pi|\pi^\star) = \textstyle\int_{\mathbb{R}^d} \log((\mathrm{d}\pi/\mathrm{d}\mu)(\mathrm{d}\pi^\star/\mathrm{d}\mu)(x,y)^{-1})\mathrm{d}\pi(x,y) = \mathrm{KL}(\pi|\mu) - \mathrm{KL}(\pi^\star|\mu).$$

Hence, $\mathrm{KL}(\pi|\mu) \geq \mathrm{KL}(\pi^\star|\mu)$ with equality if and only if $\pi^\star = \pi$. Therefore, $\pi^\star$ is the Schrödinger bridge. □

The following proposition is an alternative to Proposition 39.

Proposition 42. *Assume* **B**1. *Then there exists a solution $\pi^\star$ to the Schrödinger bridge and the IPF sequence $(\pi^n)_{n\in\mathbb{N}}$ satisfies $\lim_{n\to+\infty} \|\pi^n - \pi^\infty\|_{\mathrm{TV}} = 0$ with $\pi^\infty \in \mathscr{P}_2$. If $\mathrm{KL}(\pi^\infty|\mu) < +\infty$ and $\mathrm{L}^1(\nu_0) \oplus \mathrm{L}^1(\nu_1)$ is closed in $\mathrm{L}^1(\pi^\infty)$ then $\pi^\infty = \pi^\star$.*

Proof. Since $\lim_{n\to+\infty} \|\pi^n - \pi^\infty\|_{\mathrm{TV}} = 0$ by Theorem 36 and $\mathrm{KL}(\pi^\infty|\mu) < +\infty$, there exist A with $\mu(\mathsf{A}) = 1$ and $\Phi : (\mathbb{R}^d)^2 \to [0, +\infty)$ such that, up to extraction, for any $x, y \in \mathsf{A}$

$$\lim_{n\to+\infty} a_n(x)b_n(y) = \Phi(x,y),$$

and $(\mathrm{d}\pi^\infty/\mathrm{d}\mu) = \Phi$. Using Proposition 40, there exist $a, b : \mathbb{R}^d \to [0, +\infty)$ and B with $\pi^\infty(\mathsf{B}) = 1$ such that for any $x, y \in \mathsf{B}$, $(\mathrm{d}\pi^\infty/\mathrm{d}\mu)(x,y) = a(x)b(y)$. We conclude upon using Lemma 41. □

G Geometric convergence rates and convergence to ground-truth

In this section, we derive geometric convergence rates in Appendix G.1 in a Gaussian setting. In particular, we provide an explicit upper-bound on the convergence rate that depends only on the covariance of the reference measure and the target. In Appendix G.2, we show that DSB (with Brownian reference measure) converges towards the Schrödinger bridge in a Gaussian setting where the ground-truth is available. In Section 4 we show that our implementation actually recovers the Schrödinger bridge in this setting.

G.1 Geometric convergence rates

In the following proposition we show that we recover a geometric convergence rate in a Gaussian setting and derive intuition from this case study. We set $N = 1$ and assume that for any $x_0, x_N \in \mathbb{R}^d$ we have

$$p(x_0, x_N) \propto \exp[-\|x_0\|^2 + 2\alpha\langle x_0, x_N\rangle - \|x_N\|^2],$$

with $\alpha \in [0, 1)$. In this case assume that there exists $\beta > 0$ such that the target marginals are given for any $x_0, x_N \in \mathbb{R}^d$ by

$$p_{\text{data}}(x_0) \propto \exp[-\beta\|x_0\|^2], \qquad p_{\text{prior}}(x_N) \propto \exp[-\beta\|x_N\|^2].$$

Proposition 43. *Let $\alpha \in (0, 1)$ and $\beta > 0$. Then the Schrödinger bridge $\pi^\star$ exists and there exists $C \geq 0$ (explicit in the proof) such that for any $n \in \mathbb{N}$, $\mathrm{KL}(\pi^\star|\pi^n) \leq C\kappa^{2n}$, with $\kappa < 1$ given by $\kappa = \rho/(1+\rho)$ and $\rho = 2\alpha/\beta^2$. In addition, $\pi^\star$ admits a density w.r.t. the Lebesgue measure denoted $p^\star$ and given for any $x, y \in \mathbb{R}^d$ by*

$$p^\star(x, y) = \exp[-\gamma^\star\|x\|^2 + 2\alpha\langle x, y\rangle - \gamma^\star\|y\|^2] / \int_{\mathbb{R}^d} \exp[-\gamma^\star\|x\|^2 + 2\alpha\langle x, y\rangle - \gamma^\star\|y\|^2]\mathrm{d}x\mathrm{d}y,$$

with $\gamma^\star = (\beta^2/2)(1 + (1 + 4\alpha^2/\beta^2)^{1/2})$.

Remark that if $\beta^2 = 1 - \alpha^2$ then $\gamma^\star$ and $p^\star = p$, *i.e.* the IPF leaves μ invariant. Note that the performance of the IPF improves if κ is close to 0, *i.e.* if $\rho = 2\alpha/\beta^2$ is close to 0. This is the case if $\alpha \approx 0$ (the marginals are almost independent) or if $\beta \approx +\infty$ (the target distribution is close to δ_0), see Figure 8. This behavior is in accordance with the limit case where the marginals are independent or one of the target distribution is a Dirac mass in which case the IPF converges in two iterations.

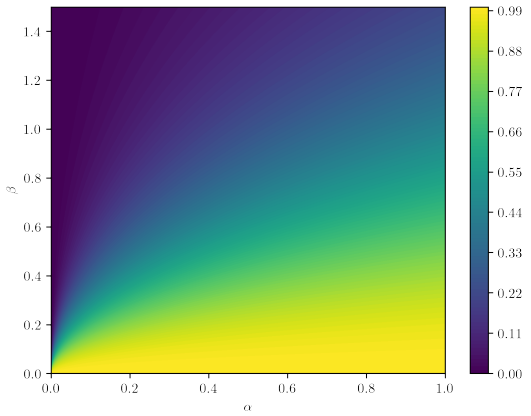


Figure 8: Evolution of κ^2 depending on α and β.

Also, note that the convergence rate does not depend on the dimension but only on the constants of the problem. In what follows we first derive the IPF sequence for this Gaussian problem and establish that α controls the amount of information shared by the marginals. Then we prove Proposition 43. In the rest of this section, we let $\mu \in \mathscr{P}_2$ with density p w.r.t. the Lebesgue measure such that for any $x_0, x_1 \in \mathbb{R}^d$

$$p(x_0, x_1) = \exp[-\|x_0\|^2 + 2\alpha\langle x_0, x_1\rangle - \|x_1\|^2] / \int_{\mathbb{R}^d} \exp[-\|x_0\|^2 + 2\alpha\langle x_0, x_1\rangle - \|x_1\|^2]\mathrm{d}x_0\mathrm{d}x_1.$$

We have that μ is the Gaussian distribution with zero mean and covariance matrix Σ such that

$$\Sigma = (2(1-\alpha^2))^{-1} \begin{pmatrix} \mathrm{Id} & \alpha\,\mathrm{Id} \\ \alpha\,\mathrm{Id} & \mathrm{Id} \end{pmatrix}.$$

We have that $\det(\Sigma) = 2^{2d}(1-\alpha^2)^{-d}$ using Schur complement (Petersen et al., 2008, Section 9.1.2). Hence we get that for any $x_0, x_1 \in \mathbb{R}^d$

$$p(x_0, x_1) = \pi^{-d}(1-\alpha^2)^{d/2} \exp[-\|x_0\|^2 + 2\alpha\langle x_0, x_1\rangle - \|x_1\|^2].$$

In what follows, we denote $C = \pi^d(1-\alpha^2)^{-d/2}$. Similarly, we get that $\mu_0 = \mu_1$ and that they admit the density p_0 w.r.t. the Lebesgue measure given for any $x \in \mathbb{R}^d$ by

$$p_0(x) = \pi^{-d/2}(1-\alpha^2)^{d/2} \exp[-\|x\|^2 (1-\alpha^2)].$$

In what follows, we denote $C_0 = \pi^{d/2}(1-\alpha^2)^{-d/2}$. In this case note that μ admits a density w.r.t. $\mu_0 \otimes \mu_1$ given for any $x_0, x_1 \in \mathbb{R}^d$ by

$$h(x_0, x_1) = (\mathrm{d}\mu/\mathrm{d}(\mu_0 \otimes \mu_1))(x_0, x_1) = (1-\alpha^2)^{-d/2} \exp[-\alpha^2 \|x_0\|^2 - 2\alpha\langle x_0, x_1\rangle - \alpha^2 \|x_1\|^2].$$

Remark that $p_{\text{prior}} = p_{\text{data}} = q$ with for any $x \in \mathbb{R}^d$, $q(x) = \pi^{-d/2}\beta^{d/2} \exp[-\beta \|x\|^2]$. We have for any $x_1, x_0 \in \mathbb{R}^d$

$$p_{1|0}(x_1|x_0) = p(x_0, x_1)/p_0(x_0) = \pi^{-d/2}(1-\alpha^2)^{d/2} \exp[-\alpha^2 \|x_0\|^2 + 2\alpha\langle x_0, x_1\rangle - \|x_1\|^2].$$

Hence, we have that **A**1 holds and the IPF sequence is well-defined and converges using Proposition 5. In what follows we start to show that α controls the amount of information shared by the two marginals μ_0 and μ_1, *i.e.* the mutual information. More precisely we have the following result.

Proposition 44. *For any $\alpha \in (0,1)$ we have* $\mathrm{KL}(\mu|\mu_0 \otimes \mu_1) = -(d/2)\log(1-\alpha^2)$.

Proof. For any $x, y \in \mathbb{R}^d$ we have

$$(\mathrm{d}\mu/(\mathrm{d}\mu_0 \otimes \mathrm{d}\mu_1))(x, y) = \exp[-\alpha^2\|x\|^2 + 2\alpha\langle x, y\rangle - \alpha^2\|y\|^2](1-\alpha^2)^{-d/2}.$$

We have that

$$\int_{\mathbb{R}^d \times \mathbb{R}^d} (-\alpha^2 \|x\|^2 - \alpha^2 \|y\|^2 + 2\alpha\langle x, y\rangle)\mathrm{d}\mu(x, y) = 0.$$

Hence, $\mathrm{KL}(\mu|\mu_0 \otimes \mu_1) = -(d/2)\log(1-\alpha^2)$, which concludes the proof. □

In what follows, we denote by $(\pi^n)_{n\in\mathbb{N}}$ the IPFP sequence, defined for any $n \in \mathbb{N}$ we have for any $x, y \in \mathbb{R}^d$

$$(\mathrm{d}\pi^{2n}/\mathrm{d}\mu)(x, y) = a_n(x)b_n(y)h(x, y), \qquad (\mathrm{d}\pi^{2n+1}/\mathrm{d}\mu)(x, y) = a_{n+1}(x)b_n(y)h(x, y),$$

where for any $x, y \in \mathbb{R}^d$

$$a_{n+1}(x) = (\mathrm{d}\nu_0/\mathrm{d}\mu_0)(x) \left(\int_{\mathbb{R}^d} h(x, y)b_n(y)\mathrm{d}\mu_1(y)\right)^{-1},$$
$$b_{n+1}(x) = (\mathrm{d}\nu_1/\mathrm{d}\mu_1)(y) \left(\int_{\mathbb{R}^d} h(x, y)a_{n+1}(x)\mathrm{d}\mu_0(x)\right)^{-1}.$$

We now turn to the proof of the Proposition 43.

Proof. Let $\alpha \in (0,1)$ and $\beta > 1$. We have for any $x, y \in \mathbb{R}^d$

$$(\mathrm{d}\nu_0/\mathrm{d}\mu_0)(x) = \exp[(1-\beta^2-\alpha^2)\|x\|^2]/C_2, \qquad (\mathrm{d}\nu_1/\mathrm{d}\mu_1)(y) = \exp[(1-\beta^2-\alpha^2)\|y\|^2]/C_2,$$

with $C_2 = C_1/C_0$ with $C_1 = \pi^{d/2}\beta^{d/2}$. For any $x \in \mathbb{R}^d$ and $\gamma \geq 0$ we have

$$\begin{aligned}
&(\mathrm{d}\nu_0/\mathrm{d}\mu_0)(x) \left(\int_{\mathbb{R}^d} \exp[-\gamma \|y\|^2]h(x, y)\mathrm{d}\mu_1(y)\right)^{-1} \\
&\quad = (C_0C_2)^{-1}C \exp[(1-\beta^2-\alpha^2)\|x\|^2] \left(\int_{\mathbb{R}^d} \exp[-\gamma \|y\|^2 - \|y-\alpha x\|^2]\mathrm{d}y\right)^{-1} \\
&\quad = (C_0C_2)^{-1}C \exp[(1-\beta^2-\alpha^2)\|x\|^2]
\end{aligned}$$

$$
\begin{aligned}
&\times \left(\int_{\mathbb{R}^d} \exp[-(\gamma+1)\left\|y-\alpha/(\gamma+1)x\right\|^2 - \alpha^2(1-1/(\gamma+1))\left\|x\right\|^2]\mathrm{d}y\right)^{-1} \\
&= (C_0C_2)^{-1}C\exp[(1-\beta^2-\alpha^2+\alpha^2\gamma/(\gamma+1))\left\|x\right\|^2] \\
&\quad \times \left(\int_{\mathbb{R}^d} \exp[-(\gamma+1)\left\|y-\alpha/(\gamma+1)x\right\|^2]\mathrm{d}y\right)^{-1} \\
&= (C_0C_2\tilde{C}_\gamma)^{-1}C\exp[(1-\beta^2-\alpha^2/(\gamma+1))\left\|x\right\|^2],
\end{aligned}
$$

with $\tilde{C}_\gamma = \pi^{d/2}(1+\gamma)^{-d/2}$. Note that $a_0 = b_0 = 1$. Let $n \in \mathbb{N}$ and assume that for any $y \in \mathbb{R}^d$ $b_n(y) = \exp[-\gamma_{2n}\left\|y\right\|^2]/C_{2n}$ with $\gamma_{2n} \geq 0$ and $C_{2n} > 0$ then we have for any $x \in \mathbb{R}^d$

$$
a_{n+1}(x) = (C_0C_2\tilde{C}_{\gamma_{2n}})^{-1}CC_{2n}\exp[-(1-\beta^2-\alpha^2/(\gamma_{2n}+1))\left\|x\right\|^2] = \exp[-\gamma_{2n+1}\left\|x\right\|^2]/C_{2n+1},
$$

with

$$
\gamma_{2n+1} = \beta^2 - 1 + \alpha^2/(\gamma_{2n}+1), \qquad (C_0C_2\tilde{C}_{\gamma_{2n}})/(CC_{2n}) = C_{2n+1}. \tag{84}
$$

Similarly, if we assume that for any $x \in \mathbb{R}^d$ $a_{n+1}(x) = \exp[-\gamma_{2n+1}\left\|x\right\|^2]/C_{2n+1}$ with $\gamma_{2n+1} \geq 0$ and $C_{2n+1} > 0$ then we have for any $y \in \mathbb{R}^d$

$$
\begin{aligned}
b_{n+1}(y) &= (C_0C_2\tilde{C}_{\gamma_{2n+1}})^{-1}(CC_{2n+1})\exp[-(1-\beta^2-\alpha^2/(\gamma_{2n+1}+1))\left\|y\right\|^2] \\
&= \exp[-\gamma_{2n+2}\left\|y\right\|^2]/C_{2n+2},
\end{aligned}
$$

with

$$
\gamma_{2n+2} = \beta^2 - 1 + \alpha^2/(\gamma_{2n+1}+1), \qquad (C_0C_2\tilde{C}_{\gamma_{2n+1}})/(CC_{2n+1}) = C_{2n+2}.
$$

Combining this result, (84) and using the recursion principle we get that for any $n \in \mathbb{N}$

$$
a_{n+1}(x) = \exp[-\gamma_{2n+1}\left\|x\right\|^2]/C_{2n+1}, \qquad b_{n+1}(y) = \exp[-\gamma_{2n+2}\left\|y\right\|^2]/C_{2n+2}.
$$

The recursion can be extended to a_0 and b_0 by setting $\gamma_{-1} = \gamma_0 = 0$ and $C_{-1} = C_0 = 1$. Therefore, for any $n \in \mathbb{N}$ we have

$$
\gamma_{n+1} = \beta^2 - 1 + \alpha^2/(\gamma_n+1). \tag{85}
$$

We now study the convergence of the sequence $(\gamma_n)_{n\in\mathbb{N}}$. By recursion, we have that for any $k, \ell \in \mathbb{N}$, if $\gamma_k \geq \gamma_\ell$ then for any $m \in \mathbb{N}$ with m even we have $\gamma_{m+k} \geq \gamma_{m+\ell}$ and for any $m \in \mathbb{N}$ with m odd we have $\gamma_{m+k} \leq \gamma_{m+\ell}$. We have $\gamma_0 = 0$ and

$$
\gamma_1 = \beta^2 + \alpha^2 - 1, \qquad \gamma_2 = \beta^2 - 1 + \alpha^2/(\beta^2+\alpha^2). \tag{86}
$$

We divide the rest of the proof into three parts.

(a) First assume that $\beta^2 > 1 - \alpha^2$. Using (86) we have that $\gamma_1 > \gamma_0$ and $\gamma_2 > \gamma_0$. Therefore, we obtain that $(\gamma_{2n})_{n\in\mathbb{N}}$ is non-decreasing, that $(\gamma_{2n+1})_{n\in\mathbb{N}}$ is non-increasing and that for any $n \in \mathbb{N}$, $0 \leq \gamma_{2n} \leq \gamma_{2n+1} \leq \gamma_1$. Therefore, $(\gamma_n)_{n\in\mathbb{N}}$ converges and we denote $\gamma^\star$ its limit. We have $\gamma^\star = \beta^2 - 1 + \alpha^2/(\gamma^\star+1)$. Hence, $\gamma^\star$ is a root of $X^2 + (2-\beta^2)X + 1 - \alpha^2 - \beta^2$. We get that $\gamma^\star = \gamma_0^\star$ or $\gamma^\star = \gamma_1^\star$ with

$$
\gamma_0^\star = \beta^2/2 - 1 - (1/2)(\beta^4+4\alpha^2)^{1/2}, \qquad \gamma_1^\star = \beta^2/2 - 1 + (1/2)(\beta^4+4\alpha^2)^{1/2},
$$

$\gamma_0^\star, \gamma_1^\star$ are non-decreasing function of β. We get that for any $\beta \geq 0$ such that $\beta^2 \geq 1-\alpha^2$, $\gamma_0^\star \leq 0$. In addition, we have $\gamma_1^\star = 0$ for $\beta^2 = 1-\alpha^2$, hence for any $\beta \geq 0$ such that $\beta^2 \geq 1-\alpha^2$, $\gamma_1^\star \geq 0$. Since $\gamma^\star \geq 0$ we have

$$
\gamma^\star = -1 + \beta^2/2 + (1/2)(\beta^4+4\alpha^2)^{1/2}. \tag{87}
$$

For any $n \in \mathbb{N}$, denote $\xi_n = \gamma_n - \gamma^\star$ and $\tau = \gamma^\star + 1$. Let $\varepsilon > 0$. Since $\lim_{n\to+\infty} \xi_n = 0$, there exists $n_0 \in \mathbb{N}$ such that $\left|\xi_n\right|/\tau \leq \varepsilon$. Using (85), we obtain that for any $n \in \mathbb{N}$

$$
\left|\xi_{n+1}\right| = \alpha^2\left|1/(\gamma_n+1) - \tau^{-1}\right| = (\alpha^2/\tau)\left|1 - (\xi_n/\tau+1)^{-1}\right| \leq (\alpha/\tau)^2\left|\xi_n\right|/(1-\varepsilon).
$$

Hence, we get that for any $\varepsilon \in (0,1)$, there exists $C_\varepsilon > 0$ such that for any $n \in \mathbb{N}$

$$
\left|\xi_n\right| \leq C_\varepsilon\kappa^n, \qquad \kappa = (\alpha/(\tau(1-\varepsilon)^{1/2}))^2.
$$

Note that $\tau > \alpha$ using (87) and $\kappa \in (0,1)$ if $\varepsilon < 1 - \alpha/\tau$.

For any $n \in \mathbb{N}$ and $x, y \in \mathbb{R}^d$ we have

$$\begin{aligned}\Phi_n(x,y) &= a_{n+1}(x)b_{n+1}(y) = \exp[-\gamma_{2n+1}\|x\|^2 - \gamma_{2n+2}\|y\|^2]/(C_{2n+1}C_{2n+2}) \\ &= \exp[-\gamma_{2n+1}\|x\|^2 - \gamma_{2n+2}\|y\|^2]/(\tilde{C}\tilde{C}_{\gamma_{2n+1}}),\end{aligned}$$

with $\tilde{C} = C_0C_2/C$. Therefore we obtain that for any $x, y \in \mathbb{R}^d$, $\Phi^\star(x,y) = \lim_{n\to+\infty}\Phi_n(x,y)$ exists and we have

$$\Phi^\star(x,y) = \exp[-\gamma^\star\|x\|^2 - \gamma^\star\|y\|^2]/(\tilde{C}\tilde{C}_{\gamma^\star}).$$

Using this result we get that for any $x, y \in \mathbb{R}^d$

$$\begin{aligned}(\mathrm{d}\pi^{2n}/\mathrm{d}\pi^\star)(x,y) &= \exp[-\xi_{2n+1}\|x\|^2 - \xi_{2n+2}\|y\|^2]C_{\gamma^\star}/C_{\gamma_{2n+1}} \\ &= \exp[-\xi_{2n+1}\|x\|^2 - \xi_{2n+2}\|y\|^2]\left\{(1+\gamma_{2n+1})/(1+\gamma^\star)\right\}^{-d/2} \\ &= \exp[-\xi_{2n+1}\|x\|^2 - \xi_{2n+2}\|y\|^2]\left\{1+\xi_{2n+1}/(1+\gamma^\star)\right\}^{-d/2}.\end{aligned}$$

Therefore we have for any $x, y \in \mathbb{R}^d$

$$\begin{aligned}\log\left((\mathrm{d}\pi^{2n}/\mathrm{d}\pi^\star)(x,y)\right) &\leq |\xi_{2n+1}|\|x\|^2 + |\xi_{2n+2}|\|y\|^2 + (d/2)\left|\log\left(1+\xi_{2n+1}/(1+\gamma^\star)\right)\right| \\ &\leq |\xi_{2n+1}|\|x\|^2 + |\xi_{2n+2}|\|y\|^2 + (d/2)|\xi_{2n+1}|.\end{aligned}$$

Therefore we obtain that for any $n \in \mathbb{N}$

$$\mathrm{KL}(\pi^\star|\pi^n) \leq (d/2)(\beta^{-2}|\xi_{2n+1}| + \beta^{-2}|\xi_{2n+2}| + |\xi_{2n+1}|).$$

A similar inequality holds for $\mathrm{KL}(\pi^\star|\pi^n)$. Therefore we get that for any $\varepsilon \in (0, 1-\alpha/\tau)$ there exists $C_\varepsilon \geq 0$ such that for any $n \in \mathbb{N}$ we have

$$\mathrm{KL}(\pi^\star|\pi^n) \leq C_\varepsilon\kappa_\varepsilon^{2n},$$

with

$$\begin{aligned}\kappa_\varepsilon &= \alpha/(\tau(1-\varepsilon)^{1/2}) = (2\alpha)/((\beta^2 + (\beta^4+4\alpha^2)^{1/2})(1-\varepsilon)^{1/2}) \\ &\leq \rho/((1+(1+\rho^2)^{1/2})(1-\varepsilon)^{1/2}).\end{aligned}$$

Let $\varepsilon < 1 - (1+\rho)/(1+(1+\rho^2)^{1/2})$. Then we get that $\kappa_\varepsilon \leq \kappa$ which concludes the first part of the proof.

(b) If $\beta^2 = 1-\alpha^2$ then the IPF is stationary since the IPF leaves μ invariant.

(c) Finally we assume that $\beta^2 < 1-\alpha^2$. Using (86) we have that $\gamma_1 < \gamma_0$ and $\gamma_2 < \gamma_0$ since $\beta^2 < 1-\alpha^2$. Therefore, we obtain that $(\gamma_{2n})_{n\in\mathbb{N}}$ is non-increasing, that $(\gamma_{2n+1})_{n\in\mathbb{N}}$ is non-decreasing and that for any $n \in \mathbb{N}$, $0 \geq \gamma_{2n} \geq \gamma_{2n+1} \geq \gamma_1$. Therefore, $(\gamma_n)_{n\in\mathbb{N}}$ converges and we denote $\gamma^\star$ its limit. We have $\gamma^\star = \beta^2 - 1 + \alpha^2/(\gamma^\star+1)$. Hence, $\gamma^\star$ is a root of $X^2 + (2-\beta^2)X + 1 - \alpha^2 - \beta^2$. We recall that the two roots of this polynomial are given by

$$\gamma_0^\star = \beta^2/2 - 1 - (1/2)(\beta^4+4\alpha^2)^{1/2}, \qquad \gamma_1^\star = \beta^2/2 - 1 + (1/2)(\beta^4+4\alpha^2)^{1/2}.$$

We have

$$\begin{aligned}\gamma_1 - \gamma_0^\star &= \beta^2 + \alpha^2 - 1 - \beta^2/2 + 1 - (1/2)(\beta^4+4\alpha^2)^{1/2} \\ &= (1/2)(\beta^2 + 2\alpha^2 - (\beta^4+4\alpha^2)^{1/2}) \geq 0.\end{aligned}$$

Since $\gamma_3 > \gamma_1$ we get that for any $n \in \mathbb{N}$ with $n \geq 3$, $\gamma_n \geq \gamma_3 > \gamma_0^\star$. Therefore $\gamma^\star > \gamma_0^\star$ and then $\gamma^\star = \gamma_1^\star$. The rest of the proof is similar to the case where $\beta^2 > 1-\alpha^2$.

□

G.2 Convergence to ground-truth

In this section, we provide an analytic form for the Schrödinger bridge in a Gaussian context. Let ν_0 be the d dimensional Gaussian distribution with mean $-a$ (with $a \in \mathbb{R}^d$) and covariance matrix $\mathbf{I} \in \mathbb{R}^{d\times d}$. Similarly, let ν_1 be the one-dimensional Gaussian distribution with mean a and covariance matrix $\mathbf{I}$. We consider the reference distribution π^0 such that $\pi^0_0 = \nu_0$ and for any $x, y \in \mathbb{R}^d$

$$(\mathrm{d}\pi^0_{1|0}/\mathrm{d}\lambda)(x, y) = (2\pi)^{-d/2} \exp[-\|x - y\|^2/2],$$

where λ denotes the Lebesgue measure on $\mathbb{R}$. Note that $\pi^0_{1|0}$ can be obtained by running a d-dimensional Brownian motion up to time 1. We consider the following Schrödinger bridge problem

$$\pi^\star = \arg\min\{\mathrm{KL}(\pi|\pi^0) \,:\, \pi \in \mathscr{P}(\mathbb{R}^{2d}),\ \pi_0 = \nu_0, \pi_1 = \nu_1\}. \tag{88}$$

Before giving the analytic solution of the SB problem we consider the following algebraic lemma.

Lemma 45. *Let $A \in \mathbb{R}^{d\times d}$ and*

$$M = \begin{pmatrix} \mathbf{I} & A \\ A^\top & \mathbf{I} \end{pmatrix}, \qquad M^S = \begin{pmatrix} \mathbf{I} & (A + A^\top)/2 \\ (A + A^\top)/2 & \mathbf{I} \end{pmatrix},$$

such that M is symmetric and positive semi-definite. Then $\det(M) \le \det(M^S)$.

Proof. Let $M^{\mathrm{up}} = M$ and $M^{\mathrm{down}} = \begin{pmatrix} \mathbf{I} & A^\top \\ A & \mathbf{I} \end{pmatrix}$. Since M^{up} is symmetric and real-valued, M^{up} is diagonalizable. Let $x, y \in \mathbb{R}^d$ and $\theta \ge 0$ such that $M^{\mathrm{up}}X = \theta X$ with $X = (x, y)$. Let $Y = (y, x)$. We have $M^{\mathrm{down}}Y = \theta Y$. Hence M^{down} is symmetric, positive semi-definite and $\det(M^{\mathrm{up}}) = \det(M^{\mathrm{down}})$. Hence using that $M \mapsto \log(\det(M))$ is concave on the space of symmetric positive semi-definite matrices we get that $\det(M^{\mathrm{up}}) \le \det((M^{\mathrm{up}} + M^{\mathrm{down}})/2) = \det(M^S)$, which concludes the proof. □

Proposition 46. *The solution to (88) exists and $\pi^\star$ is a Gaussian distribution with mean $m \in \mathbb{R}^{2d}$ and covariance matrix $\Sigma \in \mathbb{R}^{2d\times 2d}$ where*

$$m = (-a, a), \qquad \Sigma = \begin{pmatrix} \mathbf{I} & \beta\mathbf{I} \\ \beta\mathbf{I} & \mathbf{I} \end{pmatrix},$$

where $\beta = (-1 + \sqrt{5})/2$ and $\mathbf{I}$ is the d-dimensional identity matrix.

Proof. The fact that $\pi^\star$ exists and is Gaussian is similar to Proposition 43. $\pi^\star$ has mean m since $\pi^\star_i = \nu_i$ for $i \in \{0, 1\}$. Similarly, we have that $\Sigma_{00} = \Sigma_{11} = \mathbf{I}$ since $\pi^\star_i = \nu_i$ for $i \in \{0, 1\}$. We have that π^0 admits a density p^0 with respect to the Lebesgue measure such that for any $x, y \in \mathbb{R}$ we have

$$p^0(x, y) \propto \exp[-(1/2)\{2\|x\|^2 + \|y\|^2 + 2\langle a, x\rangle - 2\langle x, y\rangle + \|a\|^2\}].$$

Hence π^0 is a Gaussian distribution with mean m^0 and covariance matrix Σ^0 where

$$m^0 = (-a, -a), \qquad \Sigma^0 = \begin{pmatrix} \mathbf{I} & \mathbf{I} \\ \mathbf{I} & 2\mathbf{I} \end{pmatrix}.$$

The Kullback–Leibler divergence between a Gaussian distribution π, with mean $\tilde{m}$ and covariance matrix $\tilde{\Sigma}$, and π^0, with mean m^0 and covariance Σ^0 is given by

$$\mathrm{KL}(\pi|\pi^0) = (1/2)\{\log(\det(\Sigma^0)/\det(\tilde{\Sigma})) - d + \mathrm{Tr}(\left(\Sigma^0\right)^{-1}\tilde{\Sigma}) + (\tilde{m} - m^0)^\top \left(\Sigma^0\right)^{-1}(\tilde{m} - m^0)\}.$$

Assume that $\tilde{m} = (-a, a)$ and $\tilde{\Sigma} = \begin{pmatrix} \mathbf{I} & S \\ S^\top & \mathbf{I} \end{pmatrix}$ with $S \in \mathbb{R}^{d\times d}$ such that $\tilde{\Sigma}$ is positive semi-definite .

Then we have

$$\mathrm{KL}(\pi|\pi^0) = (1/2)\{-\log(\det(\tilde{\Sigma})) - 2\,\mathrm{Tr}(S) + C\},$$

where $C \ge 0$ is a constant which does not depend on Σ. In what follows, let $\tilde{\Sigma}' = \begin{pmatrix} \mathbf{I} & (S + S^\top)/2 \\ (S + S^\top)/2 & \mathbf{I} \end{pmatrix}$ and denote π the distribution with mean $\tilde{m}$ and covariance matrix $\tilde{\Sigma}'$.

Using Lemma 45 we have

$$\mathrm{KL}(\pi'|\pi^0) = (1/2)\{-\log(\det(\tilde{\Sigma}')) - 2\,\mathrm{Tr}(S) + C\}$$

$$\leq (1/2)\{-\log(\det(\tilde{\Sigma})) - 2\,\mathrm{Tr}(S) + C\} = \mathrm{KL}(\pi|\pi^0).$$

Hence, we can assume that $S = S^\top$ and therefore (since S is real-valued), S is diagonalizable. Let $\{\lambda_i\}_{i=1}^d$ the eigenvalues of S. Using Schur complements (Petersen et al., 2008, Section 9.1.2) we have

$$\det(\tilde{\Sigma}) = \det(\mathbf{I} - S^2) = \det(\mathbf{I} - S)\det(\mathbf{I} + S) = \textstyle\prod_{i=1}^d (1 - \lambda_i^2).$$

Therefore we have that for any $\lambda \in (0, 1)$

$$\mathrm{KL}(\pi|\pi^0) = (1/2)\textstyle\sum_{i=1}^d f(\beta_i) + C, \qquad f(\lambda) = -\log(1 - \lambda^2) - 2\lambda.$$

Hence we get that $\Sigma_{0,1} = \beta\mathbf{I}$ with $\beta = \arg\min_I f$, where $I = (-1, 0) \cup (0, 1)$. We have that $f'(\beta) = 0$ if and only if $\beta = (-1 + \sqrt{5})/2$ or $\beta = -(1 + \sqrt{5})/2$. We conclude the proof using that $\beta \in I$. □

H Continuous-time Schrödinger bridges

In this section, we prove Proposition 6 in Appendix H.1 and draw a link between the potential approach to Schrödinger bridges and DSB in continuous time in Appendix H.2.

H.1 Proof of Proposition 6

We recall the continuous Schrödinger problem is given by

$$\Pi^\star = \arg\min\left\{\mathrm{KL}(\Pi|\mathbb{P}) \,:\, \Pi \in \mathscr{P}(\mathcal{C}),\, \Pi_0 = p_{\text{data}},\, \Pi_T = p_{\text{prior}}\right\}, \quad T = \textstyle\sum_{k=0}^{N-1} \gamma_{k+1}. \tag{89}$$

In this section, we prove Proposition 6. We start with the following property which can be found in (Léonard, 2014b, Proposition 2.3, Proposition 2.10) and establishes basic properties of dynamic continuous Schrödinger bridges.

Proposition 47. *The solution to (89) exists if and only if the solution to the static Schrödinger bridge exists. In addition, if the solution exists and $\mathbb{P}$ is Markov then the Schrödinger bridge is Markov.*

We now turn to the proof of Proposition 6. First we highlight that $(\Pi^n)_{n\in\mathbb{N}}$ is well-defined since its static counterpart $(\pi_n)_{n\in\mathbb{N}}$ is well-defined using Proposition 32. We only prove that for any $n \in \mathbb{N}$, $(\Pi^{2n+1})^R$ is the path measure associated with the process $(\mathbf{Y}_t^{2n+1})_{t\in[0,T]}$ such that $\mathbf{Y}_0^{2n+1}$ has distribution p_{prior} and satisfies

$$\mathrm{d}\mathbf{Y}_t^{2n+1} = b_{T-t}^n(\mathbf{X}_t^{2n+1})\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t.$$

The proof for Π^{2n+2} is similar. Let $n \in \mathbb{N}$ and assume that Π^{2n} is the path measure associated with the process $(\mathbf{X}_t^{2n})_{t\in[0,T]}$ such that $\mathbf{X}_0^{2n}$ has distribution p_{data} and satisfies

$$\mathrm{d}\mathbf{X}_t^{2n} = f_t^n(\mathbf{X}_t^{2n})\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t.$$

We have that

$$\Pi^{2n+1} = \arg\min\left\{\mathrm{KL}(\Pi|\Pi^{2n}) \,:\, \Pi \in \mathscr{P}(\mathcal{C}),\, \Pi_T = p_{\text{prior}}\right\}.$$

Let $\phi = \mathrm{proj}_T$ such that for any $\omega \in \mathcal{C}$, $\mathrm{proj}_T(\omega) = \omega_T$. Using Proposition 24 we get that for any $\Pi \in \mathscr{P}(\mathcal{C})$ we have

$$\mathrm{KL}(\Pi|\Pi^{2n}) = \mathrm{KL}(\Pi_T|\Pi_T^{2n}) + \textstyle\int_{\mathbb{R}^d} \mathrm{KL}(\mathrm{K}(x,\cdot)|\mathrm{K}^{2n}(x,\cdot))\mathrm{d}\Pi_T(x),$$

where K and K^{2n} are the disintegrations of Π and Π^{2n} with respect to ϕ. Therefore, we get that $\Pi^{2n+1} = p_{\text{prior}}\mathrm{K}^{2n}$. Since $\mathrm{KL}(\Pi^{2n}|\mathbb{Q}) < +\infty$ and Π^{2n} is Markov, Using (Cattiaux et al., 2021, Theorem 4.9) we get that $(\Pi^{2n})^R = \Pi_T\mathrm{K}^{2n}$ satisfies the martingale problem associated with the diffusion

$$\mathrm{d}\mathbf{Y}_t^{2n} = \left\{-f_{T-t}^n(\mathbf{Y}_t^{2n}) + 2\nabla \log p_{T-t}^n(\mathbf{Y}_t^{2n})\right\}\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t. \tag{90}$$

Since $\Pi^{2n+1} = p_{\text{prior}}\mathrm{K}^{2n}$ we get that Π^{2n+1} also satisfies the martingale problem associated with (90) and is Markov which concludes the proof by recursion.

H.2 IPF in continuous time and potentials

First, we recall that the IPF $(\Pi^n)_{n\in\mathbb{N}}$ with $\Pi^0 = \mathbb{P}$ associated with (6) and for any $n \in \mathbb{N}$

$$\Pi^{2n+1} = \arg\min \left\{ \mathrm{KL}(\Pi|\Pi^{2n}) \,:\, \Pi \in \mathscr{P}(\mathcal{C}),\ \Pi_T = p_{\text{prior}} \right\},$$
$$\Pi^{2n+2} = \arg\min \left\{ \mathrm{KL}(\Pi|\Pi^{2n+1}) \,:\, \Pi \in \mathscr{P}(\mathcal{C}),\ \Pi_0 = p_{\text{data}} \right\}.$$

In this section, we draw a link between our time-reversal approach and the potential approach in continuous time. More precisely, we explicit an identity between the two in Proposition 48.

Proposition 48. *Assume* **A**1 *and that there exist* $\mathbb{M} \in \mathscr{P}(\mathcal{C})$, $U \in \mathrm{C}^1(\mathbb{R}^d, \mathbb{R})$, $C \geq 0$ *such that for any* $n \in \mathbb{N}$, $x \in \mathbb{R}^d$, $\mathrm{KL}(\Pi^n|\mathbb{M}) < +\infty$, $\langle x, \nabla U(x)\rangle \geq -C(1 + \|x\|^2)$ *and* $\mathbb{M}$ *is associated with*

$$\mathrm{d}\mathbf{X}_t = -\nabla U(\mathbf{X}_t)\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t,$$

with $\mathbf{X}_0$ *distributed according to the invariant distribution of* (15). *For any* $n \in \mathbb{N}$, *let* $\{\varphi_t^{n,\star}, \varphi_t^{n,\circ}\}_{t=0}^T$ *such that for any* $t \in [0,T]$, $\varphi_T^{n,\star} : \mathbb{R}^d \to \mathbb{R}$, $\varphi_0^{n,\circ} : \mathbb{R}^d \to \mathbb{R}$, *for any* $x_0, x_T \in \mathbb{R}^d$

$$\varphi_T^{\star,n}(x_T) = p_{\text{prior}}(x_T)/p_T^n(x_T), \qquad \varphi_0^{\circ,n}(x_0) = p_{\text{data}}(x_0)/p_0^{n+1}(x_0),$$

and for any $t \in (0,T)$ *and* $x_t \in \mathbb{R}^d$

$$\varphi_t^{\star,n}(x_t) = \int \varphi_T^{\star,n}(x_T) p_{T|t}^n(x_T|x_t)\mathrm{d}x_T, \qquad \varphi_t^{\circ,n+1}(x) = \int \varphi_0^{\circ,n+1}(x_0) q_{0|t}^n(x_0|x_t)\mathrm{d}x_0.$$

We have for any $n \in \mathbb{N}$, $t \in [0,T]$ *and* $x_t \in \mathbb{R}^d$

$$q_t^n(x_t) = p_t^n(x_t)\varphi_t^{\star,n}(x_t), \qquad p_t^{n+1}(x_t) = q_t^n(x_t)\varphi_t^{\circ,n}(x_t). \tag{91}$$

In particular, for any $n \in \mathbb{N}$ *we have*

(a) $(\Pi^{2n+1})^R$ *is associated with* $\mathrm{d}\mathbf{Y}_t^{2n+1} = b_{T-t}^n(\mathbf{Y}_t^{2n+1})\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t$ *with* $\mathbf{Y}_0^{2n+1} \sim p_{\text{prior}}$;

(b) Π^{2n+2} *is associated with* $\mathrm{d}\mathbf{X}_t^{2n+2} = f_t^{n+1}(\mathbf{X}_t^{2n+2})\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t$ *with* $\mathbf{X}_0^{2n+2} \sim p_{\text{data}}$;

with for any $x \in \mathbb{R}^d$ *and* $t \in (0,T)$

$$f_t^n(x) = f(x) + 2\sum_{k=1}^n \nabla \log \varphi_t^{\star,n}(x),\ b_t^n(x) = -f(x) + \nabla \log p_t^0(x) + 2\sum_{k=1}^n \nabla \log \varphi_t^{\circ,n}(x). \tag{92}$$

Proof. We only prove that (91) holds. Then (92) is a direct consequence of (91) and Proposition 6. Let $n \in \mathbb{N}$. Similarly to the proof of Proposition 26, there exists $\varphi_T^{\star,n} : \mathbb{R}^d \to \mathbb{R}_+$ such that for any $\{\omega_t\}_{t=0}^T \in \mathcal{C}$ we have

$$(\mathrm{d}\Pi^{2n+1}/\mathrm{d}\Pi^{2n})(\{\omega_t\}_{t=0}^T) = \varphi_T^{\star,n}(\omega_T). \tag{93}$$

Note that as in Proposition 6, that for any $s,t \in [0,T]$, $\Pi_{s,t}^{2n+1}$ admits a positive density w.r.t the Lebesgue measure denoted $q_{s,t}^n$ and $\Pi_{s,t}^{2n}$ admits a positive density w.r.t the Lebesgue measure denoted $p_{s,t}^n$. Combining this result and (93), we get that for any $t \in [0,T]$ and $x_t, x_T \in \mathbb{R}^d$ we have

$$q_{t,T}^n(x_t, x_T) = p_{t,T}^n(x_t, x_T)\varphi_T^{\star,n}(x_T).$$

We have that for any $t \in [0,T]$

$$q_t(x_t) = p_t^n(x_t) \int \varphi_T^{\star,n}(x_T) p_{T|t}^n(x_T|x_t)\mathrm{d}x_T = p_t^n(x_t)\varphi_t^{\star,n}(x_t).$$

The proof for that for any $n \in \mathbb{N}$, $t \in [0,T]$ and $x_t \in \mathbb{R}^d$, $p_t^{n+1}(x_t) = q_t^n(x_t)\varphi_t^{\circ,n}(x_t)$, is similar. □

The link between the two formulations is explicit in (91). Then, (92) is a straightforward consequence of (91) and should be compared with Appendix H.1. Another proof of Proposition 48 make use of a generalization of (91) to joint densities and use the fact that for any $n \in \mathbb{N}$, Π^{n+1} is a Doob h-transform of Π^n (see (Rogers and Williams, 2000, Paragraph 39.1) for a definition). Note that this relationship between the potential and the density of the half-bridge is not new. In particular, a similar version of this equation can be found in Bernton et al. (2019). In Finlay et al. (2020), the authors establish a similar relationship in the case of the full Schrödinger bridge.

H.3 Likelihood computation for Schrödinger bridges

We provide here details on the likelihood computation of generative models obtained with Schrödinger bridges. Under the conditions of (Léonard, 2011, Theorem 4.12), we define $(\mathbf{X}_t^\star)_{t\in[0,T]}$ the diffusion associated with $\Pi^\star$, see (89) as well as its time reversal, $(\mathbf{Y}_t^\star)_{t\in[0,T]}$. There exist $f^\star, b^\star : [0,T]\times\mathbb{R}^d \to \mathbb{R}^d$ such that $(\mathbf{X}_t^\star)_{t\in[0,T]}$ and $(\mathbf{Y}_t^\star)_{t\in[0,T]}$ are weak solutions to the following SDEs

$$\mathrm{d}\mathbf{X}_t^\star = f_t^\star(\mathbf{X}_t^\star)\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t, \qquad \mathrm{d}\mathbf{Y}_t^\star = b_{T-t}^\star(\mathbf{Y}_t^\star)\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t.$$

We assume that for any $t\in[0,T]$ there exists $p_t^\star : \mathbb{R}^d \to \mathbb{R}_+$ such that for any $x\in\mathbb{R}^d$, $(\mathrm{d}\Pi_t^\star/\mathrm{d}\lambda)(x) = p_t^\star(x)$. In addition, we assume that $p^\star \in \mathrm{C}^\infty([0,T]\times\mathbb{R}^d,\mathbb{R}_+)$. In this case, we have that $\Pi^\star$ is also associated with the process $(\tilde{\mathbf{X}}_t^\star)_{t\in[0,T]}$ associated with the ODE

$$\mathrm{d}\tilde{\mathbf{X}}_t^\star = \{f_t^\star(\tilde{\mathbf{X}}_t^\star) - \nabla\log p_t^\star(\tilde{\mathbf{X}}_t^\star)\}\mathrm{d}t,$$

and $\tilde{\mathbf{X}}_T^\star$ has distribution p_{prior}; see e.g. (Song et al., 2021, Section A). Since $(\mathbf{Y}_t^\star)_{t\in[0,T]}$ is the time-reversal of $(\mathbf{X}_t^\star)_{t\in[0,T]}$ we have that for any $t\in[0,T]$ and $x\in\mathbb{R}^d$

$$b_t^\star(x) = -f_t^\star(x) + 2\nabla\log p_t^\star(x).$$

Therefore, we get that $(\tilde{\mathbf{X}}_t^\star)_{t\in[0,T]}$ is associated with the ODE

$$\mathrm{d}\tilde{\mathbf{X}}_t^\star = \frac{1}{2}\left(f_t^\star(\tilde{\mathbf{X}}_t^\star) - b_t^\star(\tilde{\mathbf{X}}_t^\star)\right)\mathrm{d}t. \tag{94}$$

Using this result we can compute the log-likelihood of the model using the instantaneous change of variable formula (Chen et al., 2018), see also (Song et al., 2021, Appendix D.2)

$$\log p_{\text{data}}(\tilde{\mathbf{X}}_0^\star) = \log p_{\text{prior}}(\tilde{\mathbf{X}}_T^\star) + \tfrac{1}{2}\int_0^T \mathrm{div}(f_t^\star - b_t^\star)(\tilde{\mathbf{X}}_t^\star)\mathrm{d}t\ . \tag{95}$$

As in Song et al. (2021), we can use the Skilling–Hutchinson trace estimator to compute the divergence operator (Skilling, 1989; Hutchinson, 1989). In practice, we discretize the dynamics of $(\tilde{\mathbf{X}}_t^\star)_{t\in[0,T]}$ and use the network B_{β^n} obtained with the last iterate of Algorithm 1 and solve the ODE backward in time, recalling that $\tilde{\mathbf{X}}_T^\star$ has distribution p_{prior}. Similarly, we can define

$$\mathrm{d}\tilde{\mathbf{Y}}_t = \{b_{T-t}^\star(\tilde{\mathbf{Y}}_t^\star) - \nabla\log p_{T-t}^\star(\tilde{\mathbf{Y}}_t^\star)\}\mathrm{d}t,$$

and $\mathbf{Y}_0^\star$ has distribution p_{prior}. Similarly to (94), we get that $(\tilde{\mathbf{Y}}_t^\star)_{t\in[0,T]}$ is associated with the ODE

$$\mathrm{d}\tilde{\mathbf{Y}}_t = \frac{1}{2}\left(b_{T-t}^\star(\tilde{\mathbf{Y}}_t^\star) - f_{T-t}^\star(\tilde{\mathbf{Y}}_t^\star)\right)\mathrm{d}t.$$

Similarly to (96), we have

$$\log p_{\text{data}}(\tilde{\mathbf{Y}}_T^\star) = \log p_{\text{prior}}(\tilde{\mathbf{Y}}_0^\star) + \tfrac{1}{2}\int_0^T \mathrm{div}(b_{T-t}^\star - f_{T-t}^\star)(\tilde{\mathbf{Y}}_t^\star)\mathrm{d}t\ . \tag{96}$$

In practice, we discretize the dynamics of $(\tilde{\mathbf{Y}}_t^\star)_{t\in[0,T]}$ and use the networks F_{α^n}, B_{β^n} obtained with the last iterate of Algorithm 1 and solve the ODE forward in time, recalling that $\tilde{\mathbf{Y}}_0^\star$ has distribution p_{prior}. Note that in this case, we solve the ODE forward in time contrary to Durkan and Song (2021).

I Training Techniques

In this section we present some practical guidelines for the implementation of DSB, based on Algorithm 1. We emphasize that, contrarily to previous approaches Song et al. (2021); Song and Ermon (2020); Ho et al. (2020); Dhariwal and Nichol (2021), we do not weight the loss functions as we do not notice any improvement. Let $I\subset\{0,N-1\}\times\{1,M\}$. We define the generalized losses $\hat{\ell}_{n,I}^b$ and $\hat{\ell}_{n,I}^f$ given by

$$\hat{\ell}_{n,I}^b(\beta) = M^{-1}\textstyle\sum_{(k,j)\in I}\|B_\beta(k+1,X_{k+1}^j) - (X_{k+1}^j + F_k^n(X_{k+1}^j) - F_k^n(X_k^j))\|^2, \tag{97}$$

$$\hat{\ell}_{n+1,I}^f(\alpha) = M^{-1}\textstyle\sum_{(k,j)\in I}\|F_\alpha(k,X_k^j) - (X_k^j + B_{k+1}^n(X_{k+1}^j) - B_{k+1}^n(X_k^j))\|^2. \tag{98}$$

We first describe three techniques to compute these losses, then further methods to improve performance.

Technique 1. *Simulated Trajectory*

The losses (97) and (98) may be computed by simulating diffusion trajectories as described in Algorithm 1. For each sample $j \in \{1, \dots, M\}$ the skeleton of points in the sampled trajectory, $\{X_k^j\}_k$, will be correlated hence only a single uniformly sampled time-step per sample is used to compute the loss per gradient step. In addition, after the initial DSB iteration, simulating the diffusion trajectory involves computationally heavy neural network operations per diffusion step.

Technique 2. *Closed Form Sampling*

Since $f_\alpha^0(x) = -\alpha x$, with fixed α, it is not necessary to compute full trajectories for the first IPF iteration and one may sample points along the trajectory in closed-form by sampling from a Gaussian distribution with appropriate mean and covariance. This technique also improves the computational speed of the first DSB iteration.

Technique 3. *Cached Trajectory*

After the initial DSB iterations it is not possible perform closed form sampling as per Technique 2. Simulating the full diffusion trajectory is both wasteful and expensive as described in Technique 1. In order to obtain a speed-up we consider a cached-version of Algorithm 1 given by Algorithm 3 which entails storing and then resampling diffusion trajectories. Resampled trajectories are then used to compute losses (97) and (98). The cache may be refreshed at a certain frequency by once again simulating the diffusion. One may tune the cache-size and refresh frequency to available memory. This modification allows for significant speed-up as the trajectories are not simulated at each training iteration.

Algorithm 3 Cached Diffusion Schrödinger Bridge

for $n \in \{0, \dots, L\}$ **do**
  while not converged **do**
    Sample and store $\{X_k^j\}_{k,j=0}^{N,M}$ where $X_0^j \sim p_{\text{data}}$ and $X_{k+1}^j = X_k^j + \gamma_{k+1} f_{\alpha^n}(k, X_k^j) + \sqrt{2\gamma_{k+1}} Z_{k+1}^j$
    while not refreshed **do**
      Sample I (uniform in $\{0, N-1\} \times \{1, M\}$)
      Compute $\hat{\ell}_{n,I}^b(\beta^n)$ using (97)
      $\beta^n = \text{Gradient Step}(\hat{\ell}_{n,I}^b(\beta^n))$
    end while
  end while
  while not converged **do**
    Sample $\{X_k^j\}_{k,j=0}^{N,M}$, where $X_N^j \sim p_{\text{prior}}$, and $X_k^j = X_{k+1}^j + \gamma_k b_{\beta^n}(k, X_k^j) + \sqrt{2\gamma_{k+1}} Z_k^j$
    while not refreshed **do**
      Sample I (uniform in $\{0, N-1\} \times \{1, M\}$)
      Compute $\hat{\ell}_{n+1,I}^f(\alpha^{n+1})$ using (98)
      $\alpha^{n+1} = \text{Gradient Step}(\hat{\ell}_{n+1,I}^f(\alpha^{n+1}))$
    end while
  end while
end for
Output: (α^{L+1}, β^L)

Technique 4. *Tune Gaussian Prior mean/ variance*

The convergence of the IPF is affected by the mean and covariance matrix of the target Gaussian. In Appendix J.1 we investigate possible choices for these values. In practice we recommend to choose the variance of the Gaussian prior p_{prior} to be slightly larger than the one of the target dataset and to choose the mean of p_{prior} to be equal to the one of the target dataset. This remark is in accordance with (Song and Ermon, 2020, Technique 1).

Technique 5. *Network Refinement / Fine Tuning*

Training large networks from scratch, per DSB iteration, is very expensive. However, from (64)-(65),

$$b_{k+1}^n(x) = b_{k+1}^{n-1}(x) + 2\nabla \log p_{k+1}^n(x) - 2\nabla \log q_k^{n-1}(x),$$
$$f_k^n(x) = f_k^{n-1}(x) + 2\nabla \log q_k^{n-1}(x) - 2\nabla \log p_{k+1}^{n-1}(x).$$

One may therefore initialize networks at DBS iteration n from $n-1$ in order to reduce training time. In future work, we plan to investigate more sophisticated warm-start approaches through meta-learning.

Technique 6. *Exponential Moving Average*

Similar to (Song and Ermon, 2020, Technique 5), we found taking the exponential moving average of network parameters across training iterations, with rate 0.999, improved performance.

J Additional Experimental Results and Details

We provide additional examples for the two-dimensional setting in Appendix J.1. We then turn to higher dimensional generative modeling in Appendix J.2. Finally, we detail our dataset interpolation experiments in Appendix J.3. Code is available here: https://github.com/JTT94/diffusion_schrodinger_bridge.

J.1 Two-dimensional experiments

In the case of two-dimensional distributions we use a simple architecture for the networks f_α and b_β, see Figure 9. We use the variational formulation Appendix E.2.2 because our network architecture does not have a residual structure. To optimize our networks we use ADAM Kingma and Ba (2014) with momentum 0.9 and learning rate 10^{-4}.

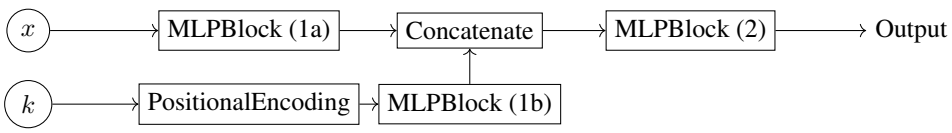


Figure 9: Architecture of the networks used in the two-dimensional setting. Each MLP Block is a Multilayer perceptron network. The "PositionalEncoding" block applies the sine transform described in Vaswani et al. (2017). MLPBlock (1a) has shape $(2, 16, 32)$, MLPBlock (1b) has shape $(1, 16, 32)$ and MLPBlock has shape $(64, 128, 128, 2)$. The total number of parameters is 26498.

In all two-dimensional experiments we fix $\gamma_k = 10^{-2}$ and use a batch size of 512. The mean and variance of p_{prior} are matched to those of p_{data}. The cache contains 10^4 samples and is refreshed every 10^3 iterations. We train each DSB step for 10^4 iterations. All two-dimensional experiments are run on Intel(R) Core(TM) i7-10850H CPU @ 2.70GHz CPUs.

In Figure 10 we present additional two-dimensional experiments.

We found that the variance of p_{prior} has an impact on the convergence speed of DSB, see Figure 11 for an illustration. This remark is in accordance with (Song and Ermon, 2020, Technique 1). In practice we recommend to set the variance to be larger than the variance of the target dataset, see Technique 4 in Appendix I.

Finally, since DSB does not require the number of Langevin iterations N to be large, one may question why not use $N = 1$ in order to derive a feed-forward generative model. In practice this choice of N is not desirable for two reasons. (a) Firstly, since p_N is not a good approximation of p_{prior}, theoretical results such as (Léger, 2020, Corollary 1) indicates that more IPF iterations are needed. (b) Second, in our experiments we observe that in order to obtain similar results to $N = 10$ with $N = 1$ we need to substantially increase the size of the networks, even for a large number of IPF iterations, see Figure 12.

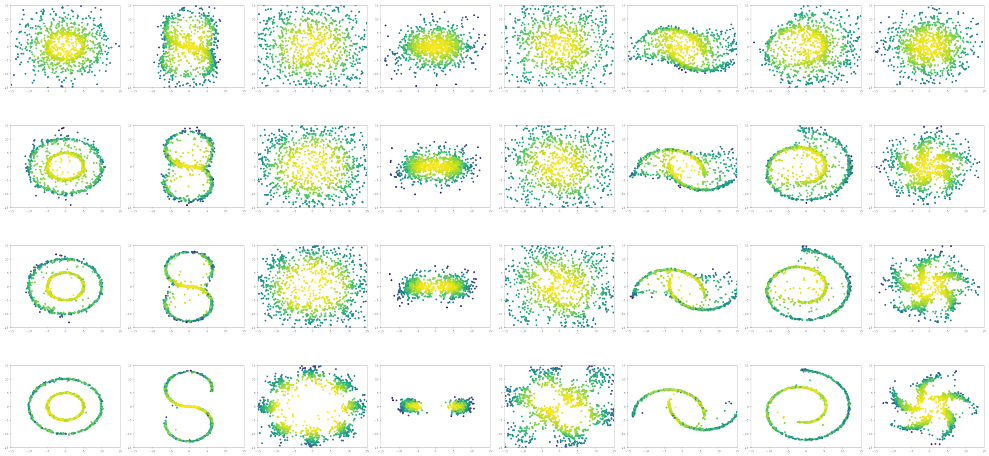
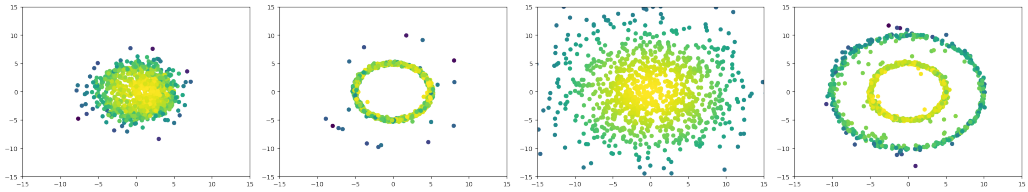


Figure 10: The first row corresponds to iteration 1 of DSB, the second to iteration 3 of DSB, the third to iteration 5 of DSB and the last to iteration 20 of DSB.



(a) p_{prior} (b) DSB iteration 5 (c) p_{prior} (d) DSB iteration 5

Figure 11: Effect of the variance of p_{prior} on the convergence of DSB. If p_{prior} has a small variance σ^2 (here $\sigma^2 = 5$ in (a) and (b)) then DSB converges more slowly. If $\sigma^2 \approx \sigma^2_{\text{data}}$, where σ^2_{data} is the variance of p_{data} then we observe more diversity in the samples obtained using DSB even for few iterations.

J.2 Generative Modeling

Implementation details We use a reduced version of the U-net architecture from Nichol and Dhariwal (2021) for F_α and B_β, where we set the number of channels to 64 rather than 128 for computational resource purposes. We tried the architecture of Song and Ermon (2020), however we observed worse results in our framework. Although we observed improvement using the corrector scheme of Song et al. (2021), this improvement was similar to augmenting the number of steps in the Langevin scheme. We therefore chose to avoid using such techniques altogether because of the increase in computing time when sampling, often by doubling the number of passes through the network.

We chose the sequence $\{\gamma_k\}_{k=0}^N$ to be invariant by time reversal, *i.e.* for any $k \in \{0, \dots, N\}$, $\gamma_k = \gamma_{N-k}$. In practice, we assume that N is even and let $\gamma_k = \gamma_0 + (2k/N)(\bar{\gamma} - \gamma_0)$ for $k \in \{0, \dots, N/2\}$ with $\gamma_0 = 10^{-5}$ and $\bar{\gamma} = 10^{-1}$. The rest of the sequence is obtained by symmetry.

In the case of the MNIST dataset (dimension $d = 28 \times 28 = 784$) we set the batch size to 128, the number of samples in the cache to 5×10^4 with 10 time-points sampled from each trajectory for each sample of p_{data}. We end up with an effective cache of size 5×10^5. The cache is refreshed each 10^3 iterations and the networks are trained for 5×10^3 iterations. Again we use the ADAM optimizer with momentum 0.9 and learning rate 10^{-4}. p_{prior} is a Gaussian density with zero mean and identity covariance matrix. We have presented results for varying number of diffusion steps, N.

In the case of the CelebA dataset (dimension $d = 32 \times 32 \times 3 = 3072$) we set the batch size to 256, number of steps $N = 50$, the number of samples in the cache to 250 with 1 time-point sampled from each trajectory for each sample of p_{data}. The cache is refreshed each 10^2 iterations and the

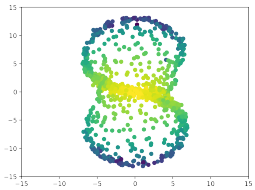


Figure 12: Failure of DSB for low N. DSB iteration 3 with $N = 2$ and $30,000$ training steps per DSB iteration. The results deteriorate significantly after 5 iterations of the algorithm.

networks are trained for 5×10^3 iterations. Again we use the ADAM optimizer with momentum 0.9 and learning rate 10^{-4}. p_{prior} is a Gaussian density with zero mean and identity covariance matrix.

Our results on MNIST and CelebA are computed using up to 4 NVIDIA Tesla V100 from the Google Cloud Platform.

Additional examples In this section we present additional examples for our high-dimensional generative modeling experiments. In Figure 13 we perform interpolation in the latent space. More precisely we let X_N^0 and X_N^1 be two samples from p_{prior}. We then compute $X_N^\lambda = (1-\lambda)X_N^0 + \lambda X_N^1$ for different values of $\lambda \in [0,1]$. For each value of $\lambda \in [0,1]$ we associate X_0^λ which corresponds to the output sample obtained using the generative model given by DSB with final condition X_N^λ. Note that in order to obtain a deterministic embedding we fix the Gaussian random variables used in the sampling. One could also have used the deterministic embedding used by Song et al. (2021), *i.e.* a neural ordinary differential equation that admits the same marginals as the diffusion thus enabling exact likelihood computation, see Appendix H.3 for details.

Figure 13: Interpolation in the latent space for MNIST.

In Figure 14 we present high quality samples for MNIST. In order to obtain these high quality samples we consider our baseline MNIST configuration but instead of choosing $N = 10$ time steps we consider $N = 30$. In addition, we train the networks for 15×10^3 iterations instead of 5×10^3. The number of samples in the cache is $M = 500$

In Figure 15 we present a temperature scaling exploration of the embedding obtained for CelebA. Similarly to the interpolation experiment we fix the Gaussian random variables in order to obtain a deterministic mapping from the latent space to the image space.

In Figure 16 we explore the latent space of our embedding of CelebA. To do so, we obtain samples using a Ornstein-Ulhenbeck process targeting p_{prior}. We refer to our project page project webpage for an animated version of this latent space exploration.

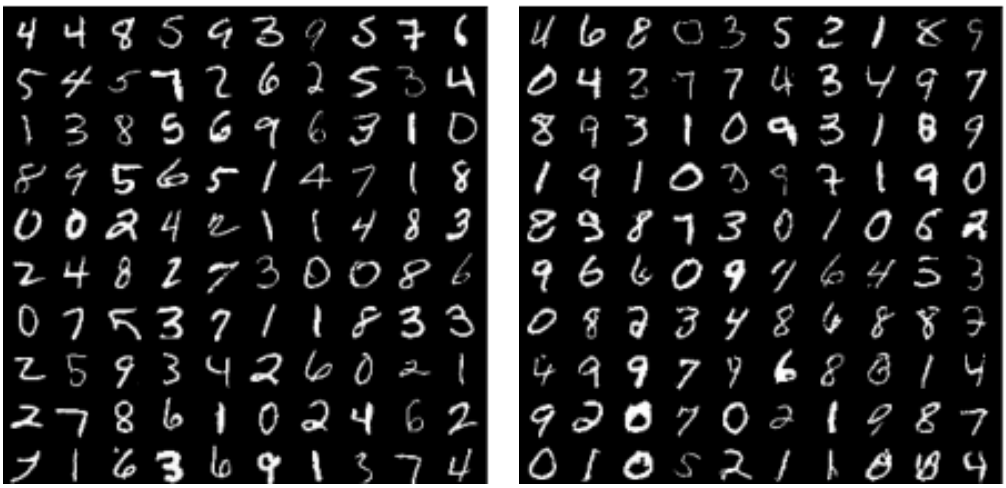


Figure 14: MNIST samples: original dataset (left) and generated MNIST samples (right) after 12 DSB iterations

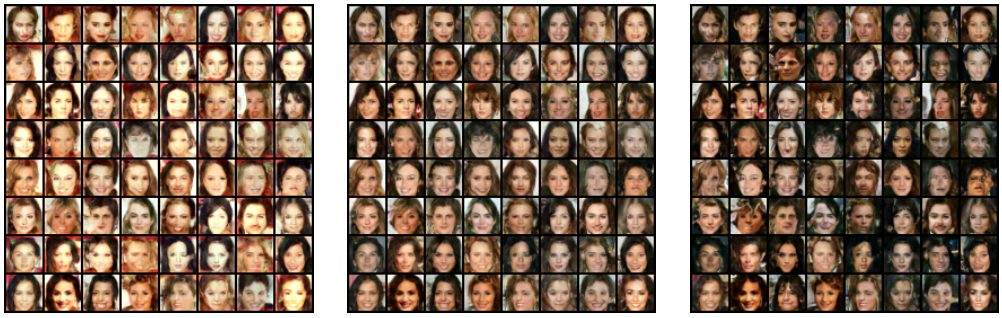


Figure 15: Temperature scaling in the latent space.

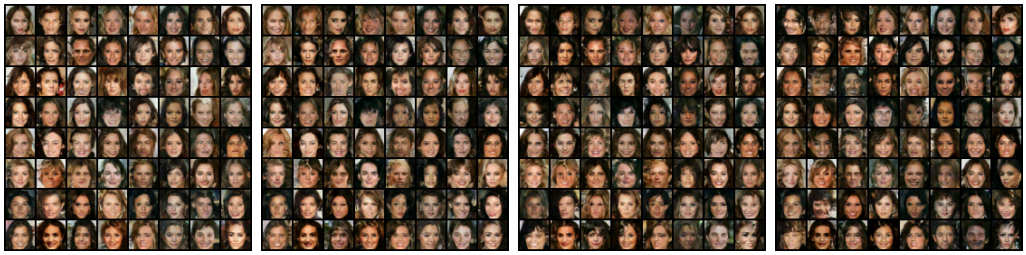


Figure 16: Exploration of the latent space. Samples are generated using a Ornstein-Ulhenbeck process targeting p_{prior} to obtain the initial condition then using the generative model given by DSB. From left to right to right: samples at time $t = 0, 1.3, 3.6, 8.6$.

J.3 Dataset interpolation

For the dataset interpolation task we keep the same parameters and architecture as before except that the number of Langevin steps is increased to 50 steps in the two-dimensional examples and to 30 steps in the EMNIST/MNIST interpolation task. We also change the reference dynamics which is chosen to be the one obtained with the DSB where p_{prior} is a Gaussian. This choice allows us to speed up the training of DSB in this setting. Animated plots are available at project webpage.

EMNIST/MNIST In order to perform translation between the dataset of handwritten letters (EMNIST) and handwritten digits (MNIST) we reduce EMNIST to 5 letters so that it contains as many classes as MNIST (we distinguish upper-case and lower-case letters), see Cohen et al. (2017) for the original dataset.

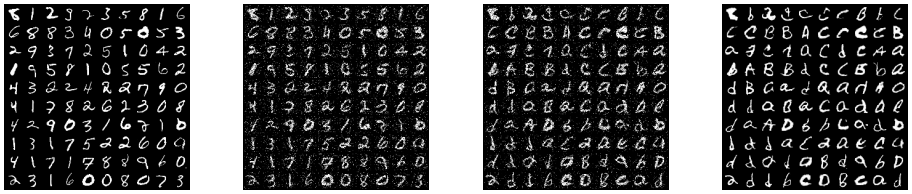


Figure 17: Iteration 10 of the IPF with $T = 1.5$ (30 diffusions steps). From left to right: $t = 0, 0.4, 1.25, 1.5$.

Two dimensional examples We present dataset interpolation for a number of classical two-dimensional datasets.

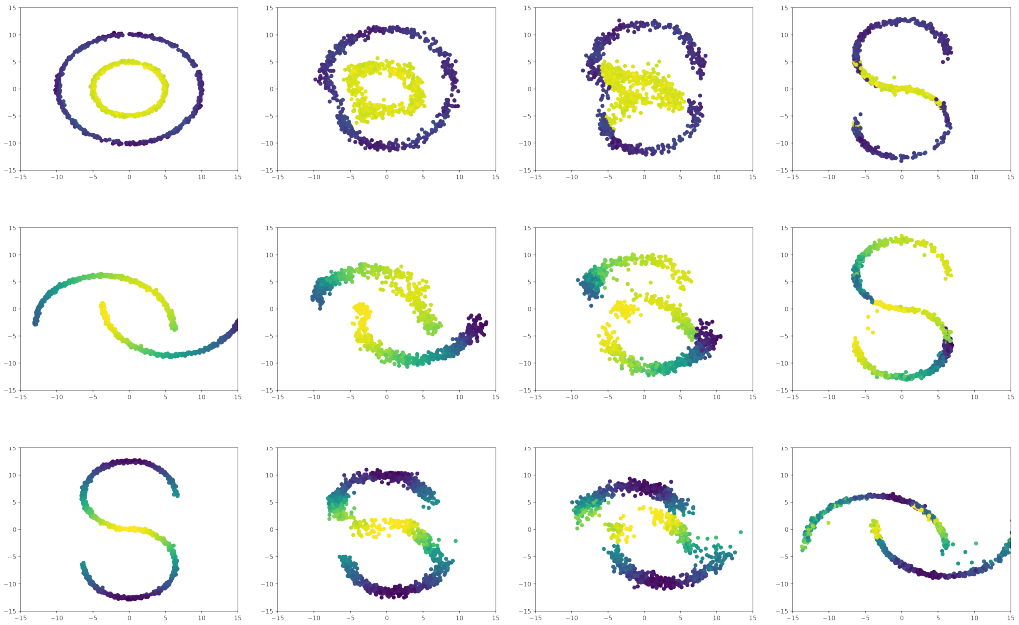


Figure 18: Dataset interpolation (DSB iteration 9). From left to right: $t = 0, 0.15, 0.30, 0.5$.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Diffusion Schrodinger Bridge with Applications to Score-based Generative Modeling
Publication Status	□ Accepted for Publication
Publication Details	Valentin De Bortoli, James Thornton, Jeremy Heng, Arnaud Doucet, published at NeurIPS 2021

Student Confirmation

Student Name:	James Thornton	
Contribution to the Paper	- I contributed to development of algorithm, in particular the training procedure. - I wrote the code for primary method, data-to-data method and experiments. - I carried out the majority of experiments (exception of Gaussian to Gaussian experiment). - I jointly introduced new training techniques for scalable and stable training (subsample replay buffer / cache of trajectories; parameterize drift rather than score and scale by step-size, symmetric step-size schedule) with Valentin De Bortoli. - I jointly came-up with the image-to-image diffusion method with Valentin De Bortoli. - I provided the likelihood computation formulation. - I wrote the proposal application and received $5k compute credits to fund the project. experiments on cloud compute. - All jointly wrote the paper. - The initial idea to use diffusion models for IPF was suggested by Arnaud Doucet from prior work with Jeremy Heng. - The iterative time reversal idea was from Valentin De Bortoli. - Theoretical results were derived by Valentin De Bortoli.	
	Date:	23/03/2023

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Arnaud Doucet, Professor of Statistics		
Supervisor comments I agree with the statement of contributions.		
Signature 	Date:	23/03/2023

This completed form should be included in the thesis, at the end of the relevant chapter.

Chapter 6

Riemannian Diffusion Schrödinger Bridge

Riemannian Diffusion Schrödinger Bridge

James Thornton [1] **Michael Hutchinson** [1] **Emile Mathieu** [1]
Valentin De Bortoli [2] **Yee Whye Teh** [1] **Arnaud Doucet** [1]

Abstract

Score-based generative models exhibit state of the art performance on density estimation and generative modeling tasks. These models typically assume that the data geometry is flat, yet recent extensions have been developed to synthesize data living on Riemannian manifolds. Existing methods to accelerate sampling of diffusion models are typically not applicable in the Riemannian setting and Riemannian score-based methods have not yet been adapted to the important task of interpolation of datasets. To overcome these issues, we introduce *Riemannian Diffusion Schrödinger Bridge*. Our proposed method generalizes Diffusion Schrödinger Bridge introduced in (De Bortoli et al., 2021) to the non-Euclidean setting and extends Riemannian score-based models beyond the first time reversal. We validate our proposed method on synthetic data and real Earth and climate data.

1. Background

1.1. Score Based Generative Modeling

In Euclidean spaces, Score-based Generative Modeling (SGM) (Song & Ermon, 2019; Song et al., 2021b) consists of two main components. The first is a forward *noising* process $(\mathbf{X}_t)_{t\geq 0}$ defined via the stochastic differential equation (SDE) (1) and the initial distribution $\mathbf{X}_0 \sim p_{\text{data}}$, targeting an easy-to-sample prior p_{prior}. The second component is a backward *denoising* process $(\mathbf{Y}_t)_{t\geq 0} = (\mathbf{X}_{T-t})_{t\in[0,T]}$ defined by the time-reversal of the noising SDE (1) (Cattiaux et al., 2021; Haussmann & Pardoux, 1986) from $p_{\text{prior}} = p_T$ to $p_{\text{data}} = p_0$. Here f is the drift, g is the time-dependent volatility, while $(\mathbf{B}_t)_{t\geq 0}$ is a d-dimensional Brownian motion and $\mathbb{P}_t$ is the distribution of $\mathbf{X}_t$ with corresponding density p_t. The denoising process defines a generative model by sampling $\mathbf{Y}_0 \sim p_{\text{prior}}$

$$\mathrm{d}\mathbf{X}_t = f(t,\mathbf{X}_t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{B}_t, \qquad \mathrm{d}\mathbf{Y}_t = \{-f(t,\mathbf{Y}_t) + g^2(t)\nabla \log p_{T-t}(\mathbf{Y}_t)\}\mathrm{d}t + g(t)\mathrm{d}\mathbf{B}_t. \tag{1}$$

The intractable score term, $\nabla \log p_{T-t}(\mathbf{Y}_t)$, may be approximated by using the following score matching identity $\nabla_{x_t} \log p_t(x_t) = \int_{\mathbb{R}^d} \nabla_{x_t} \log p_{t|0}(x_t|x_0)\, p_{0|t}(x_0|x_t)\mathrm{d}x_0$, with tractable transition density $p_{t|0}$. This is then used to train a neural network $\boldsymbol{s}_\theta : \mathbb{R}\times\mathbb{R}^d \to \mathbb{R}^d$ with regression objective $\boldsymbol{s}_{\theta^*} = \arg\min_\theta \mathbb{E}_{\mathbf{X}_t,\mathbf{X}_0}\|\nabla_x \log p_{t|0}(\mathbf{X}_t|\mathbf{X}_0) - \boldsymbol{s}_\theta(t,\mathbf{X}_t)\|^2$. Once trained, one can generate samples which are approximately distributed according to p_{data} via (1) (e.g. considering the Euler–Maruyama discretization of this SDE) by substituting $\boldsymbol{s}_{\theta^*}(t,x_t) \approx \nabla_{x_t} \log p_t(x_t)$.

1.2. Riemannian Score Based Generative Modeling

De Bortoli et al. (2022) extended SGM to compact Riemannian manifolds, denoted $\mathcal{M}$, henceforth abbreviated RSGM for Riemannian Score-based Generative Modeling. Given a $\mathcal{M}$-valued diffusion process (2, left), where $\mathbf{B}_t^{\mathcal{M}}$ denotes Brownian motion on $\mathcal{M}$; the time reversal process may be written as (2, right) (De Bortoli et al., 2022, Theorem 1)

$$\mathrm{d}\mathbf{X}_t = f(t,\mathbf{X}_t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{B}_t^{\mathcal{M}}, \qquad \mathrm{d}\mathbf{Y}_t = \{-f(t,\mathbf{Y}_t) + g^2(t)\nabla \log p_{T-t}(\mathbf{Y}_t)\}\mathrm{d}t + g(t)\mathrm{d}\tilde{\mathbf{B}}_t^{\mathcal{M}}. \tag{2}$$

For a more thorough background on Riemannian geometry and time-reversal on manifolds see De Bortoli et al. (2022, App. B and G). Simulating a diffusion on a manifold is crucial to this approach. In contrast to Euclidean SGMs, closed-form

[1]Department of Statistics, University of Oxford, UK [2]Computer Science Department,
ENS, CNRS, PSL University. Correspondence to: James Thornton <james.thornton@stats.ox.ac.uk>.

sampling schemes are generally not available for diffusions on manifolds, hence approximation schemes such as the Geodesic Random Walk from Algo. 1 are required. Note that $\mathbf{B}_t^{\mathcal{M}}$ may be simulated using Algo 1 for $g = 1$, $f = 0$. This consists of applying an Euler–Maruyama step in the tangent space, whereby Gaussian noise has also been projected to the tangent space using the projection operator P, then projecting back to the manifold using the exponential mapping.

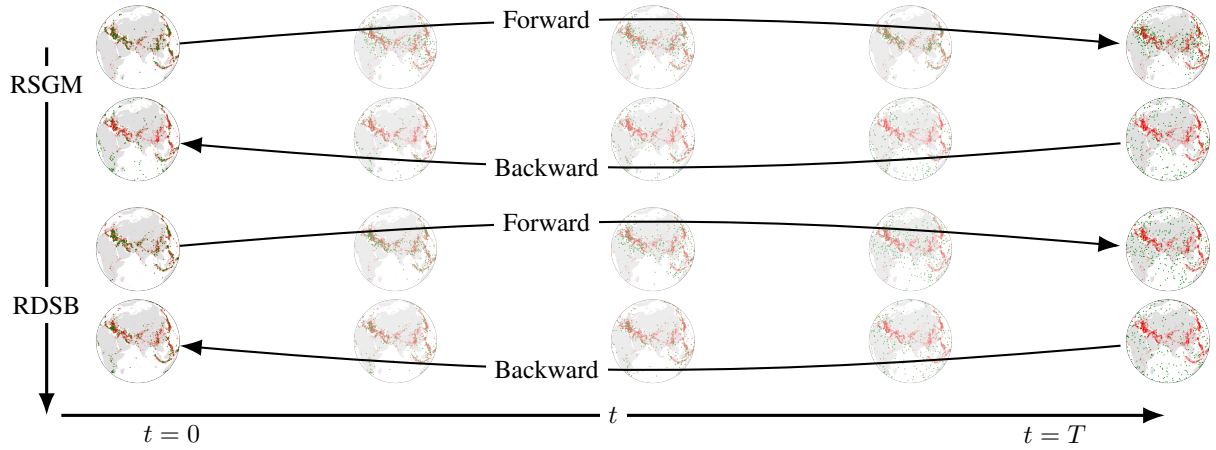


Figure 1: Earthquake data: empirical data in red, generated samples in green using $N = 10$ diffusion steps. Top forward/backward pair: RSGM. Bottom pair: RDSB, with 5 IPF steps.

Score matching may be used to approximate the time-reversal (2) in the Riemannian setting (De Bortoli et al., 2022). Unlike for Euclidean SGM, $\nabla \log p_{t|0}(x_t|x_0)$ is not available in closed-form. Instead, one may use the score identity [1] $\nabla_{x_t} \log p_t(x_t) = \int_{\mathcal{M}} \nabla_{x_t} \log p_{t|s}(x_t|x_s) \mathbb{P}_{s|t}(x_t, \mathrm{d}x_s)$ for $s \approx t$, where again $\mathbb{P}$ is the distribution of $\mathbf{X}$ and p_t corresponds to the density of $\mathbb{P}_t$ with respect to the uniform distribution on $\mathcal{M}$, then $\nabla \log p_t(x_t) = \arg\min_{\mathbf{s}_t} \int_{\mathcal{M}^2} \|\nabla_x \log p_{t|s}(x_t|x_s) - \mathbf{s}_t(x_t)\|^2 \mathrm{d}\mathbb{P}_{s,t}(x_s, x_t)$.

Algorithm 1: Simulating diffusions on a manifold

Input: step size: γ , initial state X_0
for $k \in \{0, \ldots, N-1\}$ **do**
  $\bar{\mathbf{Z}}_{k+1} \sim \mathcal{N}(0, I_p)$, $\mathbf{Z}_{k+1} = \mathrm{P}(\mathbf{X}_k)\bar{\mathbf{Z}}_{k+1}$
  $\mathbf{W}_{k+1} = \gamma f(k\gamma, \mathbf{X}_k) + \sqrt{\gamma} g(k\gamma)\mathbf{Z}_{k+1}$
  $\mathbf{X}_{k+1} = \exp_{\mathbf{X}_k}(\mathbf{W}_{k+1})$
end for
return $\{\mathbf{X}_k\}_{k=0}^{N-1}$

1.3. The Schrödinger Bridge Problem

A Schrödinger bridge extension of SGM has been introduced to reduce the number of diffusion steps for SGM by learning *both* forward and backward diffusions (De Bortoli et al., 2021). We briefly recall the notion of dynamical Schrödinger bridge (Léonard, 2012; Chen et al., 2016; Vargas et al., 2021; De Bortoli et al., 2021; Chen et al., 2022). We consider a reference path probability measure $\mathbb{P} \in \mathcal{P}(\mathrm{C}([0,T]), \mathcal{M})$ where $\mathcal{P}(\mathrm{C}([0,T]), \mathcal{M})$ is the space measures on continuous paths $(\mathbf{X}_t)_{t\in[0,1]}$ in $\mathcal{M}$. In practice, we set $\mathbb{P}$ to be the distribution of the Brownian motion $(\mathbf{B}_t^{\mathcal{M}})_{t\in[0,T]}$ initialized from p_{data}, i.e. $\mathbf{B}_0^{\mathcal{M}}$ has distribution p_{data}. We consider the *dynamical Schrödinger bridge problem*

$$\mathbb{Q}^\star = \arg\min\{\mathrm{KL}\left(\mathbb{Q}|\mathbb{P}\right) \,:\, \mathbb{Q} \in \mathcal{P}(\mathrm{C}([0,T]), \mathcal{M}),\ \mathbb{Q}_0 = p_{\text{data}},\ \mathbb{Q}_T = p_{\text{prior}}\}. \tag{3}$$

The idealised solution $\mathbb{Q}^\star$ is called the Schrödinger Bridge (SB). Given a backward process $(\mathbf{Y}_t^\star)_{t\in[0,T]}$ associated to $\mathbb{Q}^\star$, one can obtain a generative model as follows. First sample from $\mathbf{Y}_T^\star \sim p_{\text{prior}}$ and then follow the (backward) dynamics of $(\mathbf{Y}_t^\star)_{t\in[0,T]}$. By definition, we obtain that $\mathbf{Y}_0^\star \sim p_{\text{data}}$, the data distribution.

In practice, the solution of the SB problem is approximated using the Iterative Proportional Fitting (IPF) algorithm, which coincides with the Sinhkorn algorithm in discrete space (Sinkhorn, 1967; Peyré & Cuturi, 2019). IPF defines a sequence of path probability measures $(\mathbb{Q}^n)_{n\in\mathbb{N}} \in (\mathcal{P}(\mathrm{C}([0,T], \mathcal{M})))^{\mathbb{N}}$, such that $\mathbb{Q}^0 = \mathbb{P}$ and for any $n \in \mathbb{N}$

$$\mathbb{Q}^{2n+1} = \arg\min\{\mathrm{KL}\left(\mathbb{Q}|\mathbb{Q}^{2n}\right) \,:\, \mathbb{Q} \in \mathcal{P}(\mathrm{C}([0,T]), \mathcal{M}), \mathbb{Q}_T = p_{\text{prior}}\}, \tag{4}$$

$$\mathbb{Q}^{2n+2} = \arg\min\{\mathrm{KL}\left(\mathbb{Q}|\mathbb{Q}^{2n+1}\right) \,:\, \mathbb{Q} \in \mathcal{P}(\mathrm{C}([0,T]), \mathcal{M}), \mathbb{Q}_0 = p_{\text{data}}\}. \tag{5}$$

[1] Here all gradients are considered w.r.t. the Riemannian metric of $\mathcal{M}$.

Under mild assumptions on $\mathbb{P}$, p_{data} and p_{prior}, we have that $(\mathbb{Q}^n)_{n\in\mathbb{N}}$ converges towards $\mathbb{Q}^\star$ (Nutz & Wiesel, 2022).

2. Riemannian Diffusion Schrödinger Bridge

In Euclidean state spaces, De Bortoli et al. (2021) proposed Diffusion Schrödinger Bridge (DSB), an algorithm to approximate the solution to the SB problem based on time-reversal, using score matching to approximate the IPF iterates. We propose Riemannian Diffusion Schrödinger Bridge (RDSB), an extension of DSB to approximate solutions of the SB problem for compact Riemannian manifolds.

We first connect the IPF iterates $(\mathbb{Q}^n)_{n\in\mathbb{N}}$ with time reversal of diffusion processes on $\mathcal{M}$. To simplify notation, we rewrite Equation (2) in terms of drift functions f^n and b^n in Equations (6) and (7). Here $\mathbf{Y}^n$ corresponds to the time-reversal of $\mathbf{X}^n$, initialized at $\mathbf{Y}^n_T \sim p_{\text{prior}}$. $\mathbf{X}^0$ is the Brownian motion on $\mathcal{M}$, $f^n = 0$, and $\mathbf{X}^{n+1}$ denotes the time-reversal of the diffusion $\mathbf{Y}^n$, initialized at $\mathbf{X}^n_0 \sim p_{\text{data}}$.

$$\mathrm{d}\mathbf{X}^n_t = f^n(t, \mathbf{X}^n_t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{B}^{\mathcal{M}}_t, \tag{6}$$

$$\mathrm{d}\mathbf{Y}^n_t = b^n(T-t, \mathbf{Y}^n_t)\mathrm{d}t + g(t)\mathrm{d}\tilde{\mathbf{B}}^{\mathcal{M}}_t. \tag{7}$$

Proposition 2.1. *Let $\mathbb{P}$ be the path measure of the Brownian motion initialized at p_{prior} and IPF iterates $(\mathbb{Q}^n)_n$ be as defined in Section 1.3. Assume that for any $n \in \mathbb{N}$, $\mathrm{KL}(\mathbb{Q}^n|\mathbb{P}) < +\infty$ and that for any $t \in [0,T]$ and $n \in \mathbb{N}$, $\mathbb{Q}^n_t$ admits a smooth positive density w.r.t. p_{prior}. Then, for any $n \in \mathbb{N}$: $\mathbb{Q}^{2n}$ and $R(\mathbb{Q}^{2n+1})$ solve the time-reversal for (6) and (7) respectively. $R(\mathbb{P})_t = \mathbb{P}_{T-t}$ denotes the reverse time and for any $n \in \mathbb{N}$, $t \in [0,T]$ and $x \in \mathcal{M}$, $b^n(t,x) = -f^n(t,x) + g(t)^2\nabla \log p^n_t(x)$, $f^{n+1}(t,x) = -b^n(t,x) + g(t)^2\nabla \log q^n_t(x)$, with $f^0(t,x) = 0$, and p^n_t, q^n_t the densities of $\mathbb{Q}^{2n}_t$ and $\mathbb{Q}^{2n+1}_t$.*

Proof The proof follows De Bortoli et al. (2021, Proposition 6) using De Bortoli et al. (2022, Theorem 1) instead of Cattiaux et al. (2021, Theorem 4.19)

In particular, we have that $\mathbb{Q}^1$ is the diffusion process associated with RSGM, i.e. the time-reversal of the Brownian motion initialized at p_{prior}. Hence, $\mathbb{Q}^{2n+1}$ for $n \in \mathbb{N}$ with $n \geq 1$ can be seen as a refinement of $\mathbb{Q}^1$. In the next proposition, we show that the drift term of the diffusion processes associated with $(\mathbb{Q}^n)_{n\in\mathbb{N}}$ can be approximated leveraging score-based techniques.

Proposition 2.2. *Let $(\mathbf{X}_t)_{t\in[0,T]}$ be a $\mathcal{M}$-valued process with distribution $\mathbb{P} \in \mathcal{P}(\mathrm{C}([0,T]), \mathcal{M})$ such that for any $t \in [0,T]$, $\mathbf{X}_t$ admits a positive density $p_t \in \mathrm{C}^\infty(\mathcal{M})$ w.r.t. p_{prior}. For any $t \in [0,T]$ and $x \in \mathcal{M}$, let $b(t,x) = -f(t,x) + g(t)^2\nabla \log p_t(x)$. Then, for any $t \in [0,T]$, we have that*

$$b(t,\cdot) = \operatorname*{arg\,min}_{r \in \mathrm{L}^2(\mathbb{P}_t)} \mathbb{E}[\tfrac{1}{2}\|f(t,\mathbf{X}_t) + r(\mathbf{X}_t)\|_2^2 + g(t)^2 \mathrm{div}(r)(\mathbf{X}_t)].$$

Algorithm 2: RDSB

for $n \in \{0, \dots, L\}$ **do**
  while not converged **do**
    Sample $t_i \sim \mathrm{Uniform}([0,T])$
    Simulate $\{\mathbf{X}^i_{t_i}\}_{i=0}^B$, where $\mathbf{X}^i_0 \sim p_{\text{data}}$
    Compute $\hat{\ell}^b_n(\phi^n)$ using (8)
    $\phi^n \leftarrow$ Gradient Step$(\hat{\ell}^b_n(\phi^n))$
  end while
  while not converged **do**
    Sample $t_i \sim \mathrm{Uniform}([0,T])$
    Simulate $\{\mathbf{Y}^i_{t_i}\}_{i=0}^B$, where $\mathbf{Y}^i_T \sim p_{\text{prior}}$
    Compute $\hat{\ell}^f_{n+1}(\theta^{n+1})$ using (9)
    $\theta^{n+1} \leftarrow$ Gradient Step$(\hat{\ell}^f_{n+1}(\theta^{n+1}))$
  end while
end for
Output: (θ^{L+1}, ϕ^L)

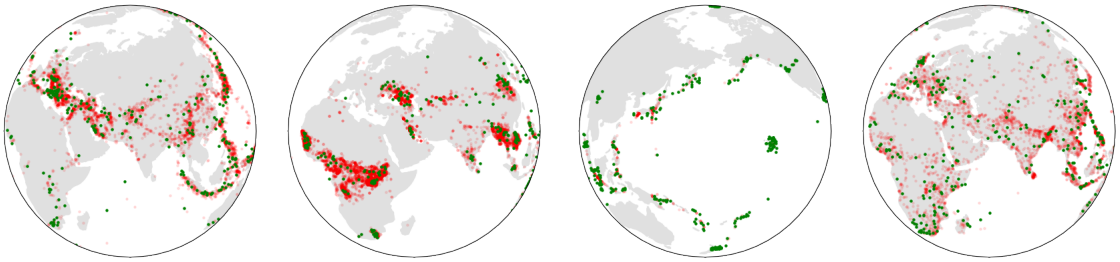
Proof. For any $x \in \mathcal{M}$, $t \in [0,T]$, $b(t,\cdot) = \arg\min_{r\in\mathcal{X}(\mathcal{M})} \mathbb{E}[\|r(t,\mathbf{X}_t) - \{-f(t,\mathbf{X}_t) + g(t)^2\nabla \log p_t(\mathbf{X}_t)\}\|]_2^2$. Expanding the quadratic, dropping terms not dependent on r and using the divergence theorem, (see Lee, 2018, p.51), $\mathbb{E}[\langle r(\mathbf{X}_t), \nabla \log p_t(\mathbf{X}_t)\rangle_{\mathcal{M}}] = -\mathbb{E}[\mathrm{div}(r)(\mathbf{X}_t)]$ concludes the proof.

In practice, for L IPF steps, $(f^n)_{n=1}^L$ and $(b^n)_{n=0}^L$ are approximated using neural networks, $(f^n_{\theta^n})_{n=1}^L$, $(b^n_{\phi^n})_{n=1}^L$ with parameters $(\theta^n, \phi^n)_n$. Approximating drifts or means of the SDE is computationally cheaper than storing an evaluating $2L$ separate score networks. Proposition 2.2 provides loss functions. The divergence terms may be estimated using automatic differentiation or Hutchinson's trace estimator (Hutchinson, 1989).

$$\hat{\ell}^b_n(\phi) = \textstyle\sum_{i=1}^B \frac{1}{2}\|f^n_\theta(t_i, x^i_{t_i}) + b^n_\phi(t_i, x^i_{t_i}))\|_2^2 + g^2(t_i)\mathrm{div}(b^n_\phi)(t_i, x^i_{t_i}) \tag{8}$$

$$\hat{\ell}^f_n(\theta) = \textstyle\sum_{i=1}^B \frac{1}{2}\|b^{n-1}_\phi(t_i, x^i_{t_i}) + f^n_\theta(t_i, x^i_{t_i})\|_2^2 + g^2(t_i)\mathrm{div}(f^n_\theta)(t_i, x^i_{t_i}) \tag{9}$$

Likelihood Computation. The Ordinary Differential Equation (ODE), $\mathrm{d}\hat{\mathbf{X}}_t = \{f(t,\hat{\mathbf{X}}_t) - \frac{1}{2}g(t)^2\nabla \log p_{T-t}(\hat{\mathbf{X}}_t)\}\mathrm{d}t$, has the same marginal probabilities as SDE (1), in particular p_{data} and hence may be used for likelihood computation using an adaptive ODE solver and trained score (Song et al., 2021b). Similar results hold for the Schrödinger Bridge (De Bortoli et al., 2021; Chen et al., 2022) and in the Riemannian setting (De Bortoli et al., 2022). We construct the appropriate



(a) Earthquake (b) Fire (c) Volcano (d) Flood

Figure 2: Climate data: $N = 10$ steps, 5 IPF iterations. Red points are training data and green are generated points.

probability flow ODE using drift approximations as $\mathrm{d}\hat{\mathbf{X}}_t = \frac{1}{2}\{f_\theta(t, \hat{\mathbf{X}}_t) - b_\phi(t, \hat{\mathbf{X}}_t)\}\mathrm{d}t$. The computed likelihood assumes convergence of the forward noising process to be valid, in particular that $p_T = p_{\text{prior}}$. RDSB enforces this convergence.

Other acceleration methods. Leading SGM acceleration techniques (Xiao et al., 2022; Song et al., 2021a) use an implicit approach to denoising by estimating x_0 from x_t, then sampling $\mathbf{X}_s|x_t, x_0$ for $s < t$. These techniques are not applicable in the Riemannian setting as it is not typically possible to sample $\mathbf{X}_s|x_t, x_0$ for $t > s$ or $\mathbf{X}_0|x_t$ for large jumps $t \not\approx s$, $t \not\approx 0$.

3. Experiments

In each of the experiments, 4-layer, fully connected networks with hidden width 512 are used for both f_θ, b_ϕ. $N = 10$ diffusion steps of size $1/N$ was used. Triangular schedule $g^2(t)$ was selected, linearly interpolated from a peak of 0.05 at $t = T/2$ and low of 0.001 at $t = 0, T$. Other experimental details follow De Bortoli et al. (2022).

3.1. Earth and Climate Data

We validate RDSB on empirical distributions of occurrences of real Earth and climate science events including earthquakes (NGDC/WDS, 2022a); wild fires (EOSDIS, 2020); volcanic eruptions (NGDC/WDS, 2022b), and floods (Brakenridge, 2017). Figure 2 illustrates that generated samples are visually similar to the empirical datasets.

Close inspection of Figure 1 shows RDSB with 5 IPF iterations exhibits better convergence than RSGM (equivalent to RDSB with 1 IPF iteration) for $N = 10$ diffusion steps on the earthquake data. In particular, RDSB generates fewer outlying samples to true data samples.

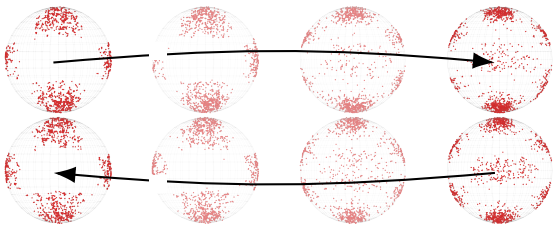


Figure 3: Interpolation between spherical harmonic datasets $(l = 2, m = 4)$ and $(l = 2, m = 4)$ after 2 IPF iterations

3.2. Dataset Interpolation on a Manifold

RDSB enables interpolation between datasets by choosing p_{prior} to be another dataset, and not necessary p_T, the terminal distribution of the inital noising process. Figure 3 illustrates RDSB applied to samples on the sphere taken with probability proportional to the real component of spherical harmonic function with parameters $(l = 2, m = 4)$ and $(l = 2, m = 6)$.

4. Future Directions

We introduce novel methodology for generative modeling and interpolation on compact Riemannian manifolds, which accelerates and generalizes RSGM. In future work, we will apply RDSB to more challenging settings such as interpolation in robotics and for protein modeling, see (Jing et al., 2022). From a theoretical point of view, the restriction to compact spaces allows us to leverage tools from Optimal Transport (Peyré & Cuturi, 2019) to prove the geometric convergence of RDSB to an approximation of the Schrödinger Bridge and quantify this bias.

References

Brakenridge, G. Global active archive of large flood events. http://floodobservatory.colorado.edu/Archives/index.html, 2017.

Cattiaux, P., Conforti, G., Gentil, I., and Léonard, C. Time reversal of diffusion processes under a finite entropy condition. *arXiv preprint arXiv:2104.07708*, 2021.

Chen, T., Liu, G.-H., and Theodorou, E. A. Likelihood training of Schrödinger bridge using forward-backward SDEs theory. In *International Conference on Learning Representations*, 2022.

Chen, Y., Georgiou, T., and Pavon, M. Entropic and displacement interpolation: a computational approach using the Hilbert metric. *SIAM Journal on Applied Mathematics*, 76(6):2375–2396, 2016.

De Bortoli, V., Thornton, J., Heng, J., and Doucet, A. Diffusion Schrödinger bridge with applications to score-based generative modeling. In *Advances in Neural Information Processing Systems*, 2021.

De Bortoli, V., Mathieu, E., Hutchinson, M., Thornton, J., Teh, Y. W., and Doucet, A. Riemannian score-based generative modeling. *arXiv preprint arXiv:2202.02763*, 2022.

EOSDIS. Land, atmosphere near real-time capability for eos (lance) system operated by NASA's earth science data and information system (esdis). https://earthdata.nasa.gov/earth-observation-data/near-real-time/firms/active-fire-data, 2020.

Haussmann, U. G. and Pardoux, E. Time reversal of diffusions. *The Annals of Probability*, 14(4):1188–1205, 1986.

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

Jing, B., Corso, G., Barzilay, R., and Jaakkola, T. S. Torsional diffusion for molecular conformer generation. In *ICLR Workshop on Machine Learning for Drug Discovery*, 2022.

Lee, J. M. *Introduction to Riemannian manifolds*. Springer, 2018.

Léonard, C. From the Schrödinger problem to the Monge–Kantorovich problem. *Journal of Functional Analysis*, 262(4): 1879–1920, 2012.

NGDC/WDS. Ncei/wds global significant earthquake database. https://www.ncei.noaa.gov/access/metadata/landing-page/bin/iso?id=gov.noaa.ngdc.mgg.hazards:G012153, 2022a.

NGDC/WDS. Ncei/wds global significant volcanic eruptions database. https://www.ncei.noaa.gov/access/metadata/landing-page/bin/iso?id=gov.noaa.ngdc.mgg.hazards:G10147, 2022b.

Nutz, M. and Wiesel, J. Stability of Schrödinger potentials and convergence of Sinkhorn's algorithm. *arXiv preprint arXiv:2201.10059*, 2022.

Peyré, G. and Cuturi, M. Computational optimal transport. *Foundations and Trends® in Machine Learning*, 11(5-6): 355–607, 2019.

Sinkhorn, R. Diagonal equivalence to matrices with prescribed row and column sums. *The American Mathematical Monthly*, 74(4):402–405, 1967.

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

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, 2019.

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

Vargas, F., Thodoroff, P., Lamacraft, A., and Lawrence, N. Solving Schrödinger bridges via maximum likelihood. *Entropy*, 23(9):1134, 2021.

Xiao, Z., Kreis, K., and Vahdat, A. Tackling the generative learning trilemma with denoising diffusion GANs. In *International Conference on Learning Representations*, 2022.

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Riemannian Diffusion Schr\"odinger Bridge
Publication Status	□ Accepted for Publication
Publication Details	ICML Continuous Time methods in Machine Learning 2022 Note: This paper was split out from Riemannian Score Based Generative Modeling, NeurIPS 2022, with the same authors.

Student Confirmation

Student Name:	James Thornton		
Contribution to the Paper	– I proposed primary idea for extending RSGM to the Schrödinger bridge setting. – I derived loss objective, corrector steps and likelihood computation for RDSB. – I wrote code and executed experiments for RDSB. – I wrote the initial draft. – Arnaud, Valentin and Emile helped refine the draft for submission.		
Signature 		Date	23/03/2023

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Arnaud Doucet, Professor of Statistics		
Supervisor comments I agree with the statement of contributions.		
Signature 	Date	23/03/2023

This completed form should be included in the thesis, at the end of the relevant chapter.

Chapter 7

Conclusion

This thesis presents four pieces of work centered around the use of regularized optimal transport in neural network parameterized probabilistic models. This section will reiterate the core contributions, limitations, and discusses future research directions.

Differentiable Particle Filtering via Entropy Regularized Optimal Transport. The work presented in **Chapter 3** investigates replacing the resampling step of particle filters, which are crucial to the stability, with a differentiable, optimal transport based proxy coined the differentiable ensemble transform (DET). DET has been shown to be highly performant for training slow-mixing, deep sequential probabilistic state-space models relative to non-differentiable heuristic methods, and has some weak convergence guarantees.

Although only used during training and can be replaced with regular multinomial sampling at deployment, computing the ensemble transform using optimal transport is computationally expensive and can lead to slow training. In addition, taking gradients sequentially through time is memory expensive. Another limitation is that the convergence results are very weak, and appear far from optimal compared to the empirical performance. There are many directions to improve and further the study of differentiable particle filtering. In particular, motivated by the success of score-matching within diffusion models [47, 86, 88], it would be interesting to investigate methods to train state-space models through particle filtering but

without tracing gradients through time. This would encourage more scalable training but perhaps be less flexible. There are also many other applications of smooth particle filtering applications that have not been fully investigated using DET, such as: within autonomous driving; on tree-shaped graphs; or for reinforcement learning, control and robotics. Although particle filtering on Riemannian manifolds has previously been considered, it would be interesting to extend the ensemble transform and regularized OT to the Riemannian manifold setting. A differentiable Riemannian particle filter could be used to tackle problems in robotics. The use of differentiable likelihoods from the particle filter allows one to perform other gradient based sampling procedures such as Hamiltonian Monte Carlo for state-space models. Acceleration strategies for OT is also an important and active area of research which would make DET more scalable. This is further discussed and motivated in **Chapter 4**.

Rethinking Initialization of the Sinkhorn Algorithm. **Chapter 4** presents a number of carefully constructed initialization strategies for accelerating the Sinkhorn algorithm while remaining differentiable and capable of being embedded within deep probabilistic models.

Although the proposed methods show dramatic speed-ups for the problems considered, no quantitative theoretical convergence bounds have yet been proven to fully understand the benefit initializations. Currently proposed initializers have only been proposed for a subset of OT tasks with Euclidean support or in uni-dimensional settings. Further work would be needed to explore the use for Riemmannian manifold valued data or for other OT tasks such as barycenter problems.

Recent work [1] also proposes a Sinkhorn initializer using neural networks. In many applications, the time spent training the neural networks exceeds the speed-ups gained from the custom initialization. In addition, currently investigated network initializers do not cater for varying sizes of discrete input measures. Extensions involving graph neural networks may be an interesting avenue to explore in order to extend such neural network initializers. It would be of interest to explore

application to larger scale problems such as for 3D shapes; single cell biology [11]; or self-supervised learning [13, 62]; which may make the initializers more worthwhile in the computational sense.

Diffusion Schrödinger Bridge. **Chapter 5** introduces the Diffusion Schrödinger Bridge (DSB), an extension of the diffusion modelling framework by iterated time-reversals. This permits data-to-data interpolation; accelerated diffusion models and facilitates high dimensional optimal transport.

Although useful for understanding diffusion models through the lens of optimal transport, the proposed method is yet to show the same remarkable empirical performance as single time-reversal diffusion methods. It would be interesting to investigate if DSB could be made competitive with regular diffusion based models.

The iterative time reversal procedure is very time consuming as it requires reversing nonlinear neural network parameterized diffusions, and is altogether computationally expensive. In addition, may result in an accumulation of errors given multiple diffusion models are trained, using the previous model's output, without correction. This makes DSB difficult to train. Recent works [65, 77] avoid some of the downsides of DSB for linear reference diffusions by iteratively constructing a coupling and then performing bridge matching updates, similar to [94]. Whilst still iterative; there is no accumulation of errors and one is not required to reverse a neural network parameterized diffusion. Simulation of the coupling is still required however, which can be expensive. Distillation [70] has been shown to accelerate the sampling of neural network parameterized diffusions. Perhaps such distillation may be used within the DSB training scheme to amortize sampling and hence accelerate training. Similarly, the DSB formulation of diffusion models is itself a form of stochastic distillation of diffusion models, complementary but differing from the deterministic distillation of prior methods [70] - this has yet to be fully explored.

The benefit of an optimal coupling has yet to be fully realized. Nor has there been detailed investigation into applications in optimal transport tasks. There are

applications, such as in filtering [78] or for barycenters where multiple DSBs are required. Given the complexity in training a single DSB, it would be extremely beneficial to somehow amortize training, and then apply this to OT tasks.

The DSB methodology has a number of issues with regards to the reference diffusion. Firstly, it has been shown by [35] that although DSB recovers the optimal coupling between marginals, the initial reference diffusion is forgotten during IPF steps, as it is only used at the start of the IPF iterations. There may be ways to correct for this, for example by initializing the forward process at each step with the reference diffusion. Finally, unlike in the Sinkhorn algorithm, the DSB approach to OT is typically limited to squared Euclidean ground cost. Although it is not clear how to tailor DSB to more general costs, there are a number of attempts to rectify this, for example by using filtering ideas [89], using the Schrödinger bridge between Gaussian approximations [10] as a reference process; or constraining the diffusion to a Riemannian manifold, as detailed in **Chapter 6** and [93].

Riemannian Diffusion Schrödinger Bridge. The final contributing paper in **Chapter 6** extends DSB to the Riemannian setting, leveraging work by the same authors for Riemannian score-based generative modelling [23]. The proposed method has the benefit of accelerating diffusions on Riemannian manifolds and permits data-to-data interpolation of Riemannian valued data.

The Riemannian Diffusion Schrödinger Bridge (RDSB) has only been investigated empirically on very simple manifolds, primarily the sphere. It would be interesting to try this method for meaningful real-world applications such as for protein or molecular modelling. Robotic interpolation would also be a prime candidate for further experimentation. Similar to the Euclidean counterpart, it would also be interesting to investigate RDSB within OT applications such as for multi-marginal OT. Another research avenue worth pursuing is in investigating higher-order Riemannian diffusions, essentially extending [28] to the Riemannian and bridge settings.

References

[1] Brandon Amos, Samuel Cohen, Giulia Luise, and Ievgen Redko. "Meta optimal transport". In: *arXiv preprint arXiv:2206.05262* (2022).

[2] Arpit Bansal, Eitan Borgnia, Hong-Min Chu, Jie S Li, Hamid Kazemi, Furong Huang, Micah Goldblum, Jonas Geiping, and Tom Goldstein. "Cold diffusion: Inverting arbitrary image transforms without noise". In: *arXiv preprint arXiv:2208.09392* (2022).

[3] Joe Benton, Yuyang Shi, Valentin De Bortoli, George Deligiannidis, and Arnaud Doucet. "From Denoising Diffusions to Denoising Markov Models". In: *arXiv preprint arXiv:2211.03595* (2022).

[4] Dimitris Bertsimas and John N Tsitsiklis. *Introduction to Linear Optimization*. Vol. 6. Athena Scientific Belmont, MA, 1997.

[5] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. "Align your latents: High-resolution video synthesis with latent diffusion models". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 22563–22575.

[6] Sam Bond-Taylor and Chris G Willcocks. "\$infty$-Diff: Infinite Resolution Diffusion with Subsampled Mollified States". In: *arXiv preprint arXiv:2303.18242* (2023).

[7] Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. "Diffusion Schrödinger Bridge with Applications to Score-Based Generative Modeling". In: *Advances in Neural Information Processing Systems* (2021).

[8] L.M. Bregman. "The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming". In: *USSR Computational Mathematics and Mathematical Physics* 7.3 (1967), pp. 200–217. URL: https://www.sciencedirect.com/science/article/pii/0041555367900407.

[9] Yann Brenier. "Décomposition polaire et réarrangement monotone des champs de vecteurs". In: *CR Acad. Sci. Paris Sér. I Math.* 305 (1987), pp. 805–808.

[10] Charlotte Bunne, Ya-Ping Hsieh, Marco Cuturi, and Andreas Krause. "Recovering stochastic dynamics via gaussian schr\" odinger bridges". In: *arXiv preprint arXiv:2202.05722* (2022).

[11] Charlotte Bunne, Laetitia Papaxanthos, Andreas Krause, and Marco Cuturi. "Proximal optimal transport modeling of population dynamics". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2022, pp. 6511–6528.

[12] Andrew Campbell, Joe Benton, Valentin De Bortoli, Thomas Rainforth, George Deligiannidis, and Arnaud Doucet. "A continuous time framework for discrete denoising models". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 28266–28279.

[13] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. "Unsupervised learning of visual features by contrasting cluster assignments". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 9912–9924.

[14] Patrick Cattiaux, Giovanni Conforti, Ivan Gentil, and Christian Léonard. "Time reversal of diffusion processes under a finite entropy condition". In: *arXiv preprint arXiv:2104.07708* (2021).

[15] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. "Neural ordinary differential equations". In: *Advances in neural information processing systems* 31 (2018).

[16] Yongxin Chen, Tryphon T Georgiou, and Michele Pavon. "Optimal steering of a linear stochastic system to a final probability distribution—part iii". In: *IEEE Transactions on Automatic Control* 63.9 (2018), pp. 3112–3118.

[17] Lenaic Chizat, Pierre Roussillon, Flavien Léger, François-Xavier Vialard, and Gabriel Peyré. "Faster wasserstein distance estimation with the sinkhorn divergence". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 2257–2269.

[18] Nicolas Chopin and Omiros Papaspiliopoulos. *An Introduction to Sequential Monte Carlo*. Springer, 2020.

[19] Adrien Corenflos, James Thornton, George Deligiannidis, and Arnaud Doucet. "Differentiable particle filtering via entropy-regularized optimal transport". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 2100–2111.

[20] Marco Cuturi. "Sinkhorn distances: Lightspeed computation of optimal transport". In: *Advances in Neural Information Processing Systems*. 2013, pp. 2292–2300.

[21] Marco Cuturi, Olivier Teboul, and Jean-Philippe Vert. "Differentiable ranking and sorting using optimal transport". In: *Advances in neural information processing systems* 32 (2019).

[22] Giannis Daras, Mauricio Delbracio, Hossein Talebi, Alexandros G Dimakis, and Peyman Milanfar. "Soft diffusion: Score matching for general corruptions". In: *arXiv preprint arXiv:2209.05442* (2022).

[23] Valentin De Bortoli, Emile Mathieu, Michael Hutchinson, James Thornton, Yee Whye Teh, and Arnaud Doucet. "Riemannian score-based generative modeling". In: *Advances in Neural Information Processing Systems* (2022).

[24] Prafulla Dhariwal and Alexander Nichol. "Diffusion models beat gans on image synthesis". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 8780–8794.

[25] Sander Dieleman. *Diffusion models are autoencoders*. 2022. URL: https://benanne.github.io/2022/01/31/diffusion.html.

[26] Sander Dieleman. *Guidance: a cheat code for diffusion models*. 2022. URL: https://benanne.github.io/2022/05/26/guidance.html.

[27] Tim Dockhorn, Arash Vahdat, and Karsten Kreis. "GENIE: Higher-order denoising diffusion solvers". In: *arXiv preprint arXiv:2210.05475* (2022).

[28] Tim Dockhorn, Arash Vahdat, and Karsten Kreis. "Score-based generative modeling with critically-damped langevin diffusion". In: *arXiv preprint arXiv:2112.07068* (2021).

[29] Randal Douc, Eric Moulines, and David Stoffer. *Nonlinear Time Series: Theory, Methods and Applications with R Examples*. CRC press, 2014.

[30] Arnaud Doucet and Adam M Johansen. "A tutorial on particle filtering and smoothing: Fifteen years later". In: *Handbook of Nonlinear Filtering* 12 (2009), pp. 656–704.

[31] Arnaud Doucet and Anthony Lee. "Sequential Monte Carlo methods". In: *Handbook of Graphical Models* (2018), pp. 165–189.

[32] Yilun Du, Conor Durkan, Robin Strudel, Joshua B Tenenbaum, Sander Dieleman, Rob Fergus, Jascha Sohl-Dickstein, Arnaud Doucet, and Will Grathwohl. "Reduce, reuse, recycle: Compositional generation with Energy-Based diffusion models and MCMC". In: *arXiv preprint arXiv:2302.11552* (2023).

[33] Vanja Dukic, Hedibert F Lopes, and Nicholas G Polson. "Tracking epidemics with Google flu trends data and a state-space SEIR model". In: *Journal of the American Statistical Association* 107.500 (2012), pp. 1410–1426.

[34] Wendelin Feiten, Muriel Lang, and Sandra Hirche. "Rigid motion estimation using mixtures of projected Gaussians". In: *International Conference on Information Fusion*. IEEE. 2013, pp. 1465–1472.

[35] David Lopes Fernandes, Francisco Vargas, Carl Henrik Ek, and Neill D. F. Campbell. "Shooting Schrödinger's Cat". In: *Fourth Symposium on Advances in Approximate Bayesian Inference*. 2022. URL: https://openreview.net/forum?id=A8bxsnCAvDs.

[36] Jean Feydy, Thibault Séjourné, François-Xavier Vialard, Shun-Ichi Amari, Alain Trouvé, and Gabriel Peyré. "Interpolating between optimal transport and MMD using Sinkhorn divergences". In: *AISTATS*. 2019.

[37] Hans Föllmer. "An entropy approach to the time reversal of diffusion processes". In: *Stochastic Differential Systems: Filtering and Control*. Springer, 1985, pp. 156–163.

[38] Robert Fortet. "Résolution d'un système d'équations de M. Schrödinger". In: *Journal de Mathématiques Pures et Appliqués* 1 (1940), pp. 83–105.

[39] Songwei Ge, Seungjun Nah, Guilin Liu, Tyler Poon, Andrew Tao, Bryan Catanzaro, David Jacobs, Jia-Bin Huang, Ming-Yu Liu, and Yogesh Balaji. "Preserve Your Own Correlation: A Noise Prior for Video Diffusion Models". In: *arXiv preprint arXiv:2305.10474* (2023).

[40] Aude Genevay. “Entropy-regularized Optimal Transport for Machine Learning”. Theses. Université Paris sciences et lettres, Mar. 2019. URL: https://theses.hal.science/tel-02458044.

[41] Aude Genevay, Gabriel Dulac-Arnold, and Jean-Philippe Vert. *Differentiable Deep Clustering with Cluster Size Constraints*. 2019. arXiv: 1910.09036 [cs.LG].

[42] Will Grathwohl, Kuan-Chieh Wang, Jörn-Henrik Jacobsen, David Duvenaud, and Richard Zemel. “Learning the stein discrepancy for training and evaluating energy-based models without sampling”. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 3732–3747.

[43] Jiatao Gu, Shuangfei Zhai, Yizhe Zhang, Miguel Angel Bautista, and Josh Susskind. “f-DM: A Multi-stage Diffusion Model via Progressive Signal Transformation”. In: *arXiv preprint arXiv:2210.04955* (2022).

[44] Xizewen Han, Huangjie Zheng, and Mingyuan Zhou. “Card: Classification and regression diffusion models”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 18100–18115.

[45] Ulrich G Haussmann and Etienne Pardoux. “Time reversal of diffusions”. In: *The Annals of Probability* 14.4 (1986), pp. 1188–1205.

[46] Jeremy Heng, Valentin De Bortoli, Arnaud Doucet, and James Thornton. “Simulating diffusion bridges with score matching”. In: *arXiv preprint arXiv:2111.07243* (2021).

[47] Jonathan Ho, Ajay Jain, and Pieter Abbeel. *Denoising Diffusion Probabilistic Models*. 2020. arXiv: 2006.11239 [cs.LG].

[48] Emiel Hoogeboom and Tim Salimans. “Blurring diffusion models”. In: *arXiv preprint arXiv:2209.05557* (2022).

[49] M.F. Hutchinson. “A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines”. In: *Communications in Statistics - Simulation and Computation* 19.2 (1990), pp. 433–450. eprint: https://doi.org/10.1080/03610919008812866. URL: https://doi.org/10.1080/03610919008812866.

[50] Aapo Hyvärinen. “Estimation of Non-Normalized Statistical Models by Score Matching”. In: *Journal of Machine Learning Research* 6.24 (2005), pp. 695–709. URL: http://jmlr.org/papers/v6/hyvarinen05a.html.

[51] Alexia Jolicoeur-Martineau, Ke Li, Rémi Piché-Taillefer, Tal Kachman, and Ioannis Mitliagkas. “Gotta go fast when generating data with score-based models”. In: *arXiv preprint arXiv:2105.14080* (2021).

[52] Lev Kantorovich. “On the Translocation of Masses”. In: *Journal of Mathematical Sciences* 133 (2006), pp. 1381–1382. URL: https://api.semanticscholar.org/CorpusID:122853046.

[53] Anuj Karpatne, Imme Ebert-Uphoff, Sai Ravela, Hassan Ali Babaie, and Vipin Kumar. “Machine learning for the geosciences: Challenges and opportunities”. In: *IEEE Transactions on Knowledge and Data Engineering* 31.8 (2018), pp. 1544–1554.

[54] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. "Elucidating the design space of diffusion-based generative models". In: *arXiv preprint arXiv:2206.00364* (2022).

[55] Gavin Kerrigan, Justin Ley, and Padhraic Smyth. "Diffusion Generative Models in Infinite Dimensions". In: *arXiv preprint arXiv:2212.00886* (2022).

[56] Tobias Lehmann, Max-K Von Renesse, Alexander Sambale, and André Uschmajew. "A note on overrelaxation in the Sinkhorn algorithm". In: *Optimization Letters* (2021), pp. 1–12.

[57] Guan-Horng Liu, Arash Vahdat, De-An Huang, Evangelos A Theodorou, Weili Nie, and Anima Anandkumar. "I^{2}SB: Image-to-Image Schr\" odinger Bridge". In: *arXiv preprint arXiv:2302.05872* (2023).

[58] Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D Plumbley. "Audioldm: Text-to-audio generation with latent diffusion models". In: *arXiv preprint arXiv:2301.12503* (2023).

[59] Morteza Mardani, Jiaming Song, Jan Kautz, and Arash Vahdat. "A Variational Perspective on Solving Inverse Problems with Diffusion Models". In: *arXiv preprint arXiv:2305.04391* (2023).

[60] Gaspard Monge. "Mémoire sur la théorie des déblais et des remblais". In: *Histoire de l'Académie Royale des Sciences* (1781), pp. 666–704.

[61] Marcel Nutz and Johannes Wiesel. "Stability of Schr\" odinger Potentials and Convergence of Sinkhorn's Algorithm". In: *arXiv preprint arXiv:2201.10059* (2022).

[62] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. "DINOv2: Learning Robust Visual Features without Supervision". In: *arXiv preprint arXiv:2304.07193* (2023).

[63] Kushagra Pandey and Stephan Mandt. "Generative Diffusions in Augmented Spaces: A Complete Recipe". In: *arXiv preprint arXiv:2303.01748* (2023).

[64] David Peel, William J Whiten, and Geoffrey J McLachlan. "Fitting mixtures of Kent distributions to aid in joint set identification". In: *Journal of the American Statistical Association* 96.453 (2001), pp. 56–63.

[65] Stefano Peluchetti. "Diffusion Bridge Mixture Transports, Schr\" odinger Bridge Problems and Generative Modeling". In: *arXiv preprint arXiv:2304.00917* (2023).

[66] Gabriel Peyré and Marco Cuturi. "Computational optimal transport". In: *Foundations and Trends® in Machine Learning* 11.5-6 (2019), pp. 355–607.

[67] Aram-Alexandre Pooladian, Heli Ben-Hamu, Carles Domingo-Enrich, Brandon Amos, Yaron Lipman, and Ricky Chen. "Multisample Flow Matching: Straightening Flows with Minibatch Couplings". In: *arXiv preprint arXiv:2304.14772* (2023).

[68] Sebastian Reich. "A nonparametric ensemble transform method for Bayesian inference". In: *SIAM Journal on Scientific Computing* 35.4 (2013), A2013–A2024.

[69] Severi Rissanen, Markus Heinonen, and Arno Solin. "Generative modelling with inverse heat dissipation". In: *arXiv preprint arXiv:2206.13397* (2022).

[70] Tim Salimans and Jonathan Ho. “Progressive distillation for fast sampling of diffusion models”. In: *arXiv preprint arXiv:2202.00512* (2022).

[71] Tim Salimans and Jonathan Ho. “Should EBMs model the energy or the score?” In: *Energy Based Models Workshop - ICLR 2021*. 2021. URL: https://openreview.net/forum?id=9AS-TF2jRNb.

[72] Filippo Santambrogio. “Optimal transport for applied mathematicians”. In: ().

[73] Simo Särkkä and Arno Solin. *Applied Stochastic Differential Equations*. Institute of Mathematical Statistics Textbooks. Cambridge University Press, 2019.

[74] Erwin Schrödinger. “Sur la théorie relativiste de l’électron et l’interprétation de la mécanique quantique”. In: *Annales de l’Institut Henri Poincaré* 2.4 (1932), pp. 269–310.

[75] Ransalu Senanayake and Fabio Ramos. “Directional grid maps: modeling multimodal angular uncertainty in dynamic environments”. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 3241–3248.

[76] Maxim V Shapovalov and Roland L Dunbrack Jr. “A smoothed backbone-dependent rotamer library for proteins derived from adaptive kernel density estimates and regressions”. In: *Structure* 19.6 (2011), pp. 844–858.

[77] Yuyang Shi, Valentin De Bortoli, Andrew Campbell, and Arnaud Doucet. “Diffusion Schr\" odinger Bridge Matching”. In: *arXiv preprint arXiv:2303.16852* (2023).

[78] Yuyang Shi, Valentin De Bortoli, George Deligiannidis, and Arnaud Doucet. “Conditional simulation using diffusion Schrödinger bridges”. In: *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*. Ed. by James Cussens and Kun Zhang. Vol. 180. Proceedings of Machine Learning Research. PMLR, 2022, pp. 1792–1802. URL: https://proceedings.mlr.press/v180/shi22a.html.

[79] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. “Make-a-video: Text-to-video generation without text-video data”. In: *arXiv preprint arXiv:2209.14792* (2022).

[80] Raghav Singhal, Mark Goldstein, and Rajesh Ranganath. “Where to Diffuse, How to Diffuse, and How to Get Back: Automated Learning for Multivariate Diffusions”. In: *arXiv preprint arXiv:2302.07261* (2023).

[81] Richard Sinkhorn. “Diagonal equivalence to matrices with prescribed row and column sums”. In: *American Mathematical Monthly* 74 (1967), pp. 402–405.

[82] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. *Deep Unsupervised Learning using Nonequilibrium Thermodynamics*. 2015. arXiv: 1503.03585 [cs.LG].

[83] Vignesh Ram Somnath, Matteo Pariset, Ya-Ping Hsieh, Maria Rodriguez Martinez, Andreas Krause, and Charlotte Bunne. “Aligned Diffusion Schr\" odinger Bridges”. In: *arXiv preprint arXiv:2302.11419* (2023).

[84] Jiaming Song, Chenlin Meng, and Stefano Ermon. “Denoising diffusion implicit models”. In: *arXiv preprint arXiv:2010.02502* (2020).

[85] Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. "Maximum likelihood training of score-based diffusion models". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 1415–1428.

[86] Yang Song and Stefano Ermon. *Generative Modeling by Estimating Gradients of the Data Distribution*. 2020. arXiv: 1907.05600 [cs.LG].

[87] Yang Song, Sahaj Garg, Jiaxin Shi, and Stefano Ermon. "Sliced score matching: A scalable approach to density and score estimation". In: *Uncertainty in Artificial Intelligence*. PMLR. 2020, pp. 574–584.

[88] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. *Score-Based Generative Modeling through Stochastic Differential Equations*. 2021. arXiv: 2011.13456 [cs.LG].

[89] Ella Tamir, Martin Trapp, and Arno Solin. "Transport with support: Data-conditional diffusion bridges". In: *arXiv preprint arXiv:2301.13636* (2023).

[90] Alexis Thibault, Lénaıc Chizat, Charles Dossal, and Nicolas Papadakis. "Overrelaxed Sinkhorn–Knopp algorithm for regularized optimal transport". In: *Algorithms* 14.5 (2021), p. 143.

[91] James Thornton and Marco Cuturi. "Rethinking initialization of the sinkhorn algorithm". In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2023, pp. 8682–8698.

[92] James Thornton, George Deligiannidis, and Arnaud Doucet. "The Masked Bouncy Particle Sampler: A Parallel, Chromatic, Piecewise-Deterministic Markov Chain Monte Carlo Method". In: (2021).

[93] James Thornton, Michael Hutchinson, Emile Mathieu, Valentin De Bortoli, Yee Whye Teh, and Arnaud Doucet. "Riemannian Diffusion Schr\"odinger Bridge". In: *Continuous Time Methods for Machine Learning, International Conference of Machine Learning,* (2022).

[94] Alexander Tong, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Kilian Fatras, Guy Wolf, and Yoshua Bengio. "Conditional flow matching: Simulation-free dynamic optimal transport". In: *arXiv preprint arXiv:2302.00482* (2023).

[95] Belinda Tzen and Maxim Raginsky. "Neural stochastic differential equations: Deep latent gaussian models in the diffusion limit". In: *arXiv preprint arXiv:1905.09883* (2019).

[96] C. Villani. *Optimal Transport: Old and New*. Grundlehren der mathematischen Wissenschaften. Springer Berlin Heidelberg, 2008. URL: https://books.google.co.uk/books?id=hV8o5R7%5C_5tkC.

[97] Gefei Wang, Yuling Jiao, Qian Xu, Yang Wang, and Can Yang. "Deep generative learning via schrödinger bridge". In: *International Conference on Machine Learning*. PMLR. 2021, pp. 10794–10804.

[98] Jonathan Weed. "An explicit analysis of the entropic penalty in linear programming". In: *Proceedings of the 31st Conference On Learning Theory*. 2018.

[99] Max Welling and Yee W Teh. "Bayesian learning via stochastic gradient Langevin dynamics". In: *Proceedings of the 28th international conference on machine learning (ICML-11)*. 2011, pp. 681–688.

[100] Lilian Weng. "What are diffusion models?" In: *lilianweng.github.io* (2021). URL: https://lilianweng.github.io/posts/2021-07-11-diffusion-models/.

[101] Yilun Xu, Ziming Liu, Max Tegmark, and Tommi Jaakkola. "Poisson flow generative models". In: *arXiv preprint arXiv:2209.11178* (2022).

[102] Maxim Zvyagin et al. "GenSLMs: Genome-scale language models reveal SARS-CoV-2 evolutionary dynamics". In: *bioRxiv* (2022). eprint: https://www.biorxiv.org/content/early/2022/10/11/2022.10.10.511571.full.pdf. URL: https://www.biorxiv.org/content/early/2022/10/11/2022.10.10.511571.