Transport with Support: Data-Conditional Diffusion Bridges

Ella Tamir *ella.tamir@aalto.fi*
Department of Computer Science
Aalto University

Martin Trapp *martin.trapp@aalto.fi*
Department of Computer Science
Aalto University

Arno Solin *arno.solin@aalto.fi*
Department of Computer Science
Aalto University

Reviewed on OpenReview: *https://openreview.net/forum?id=1309*

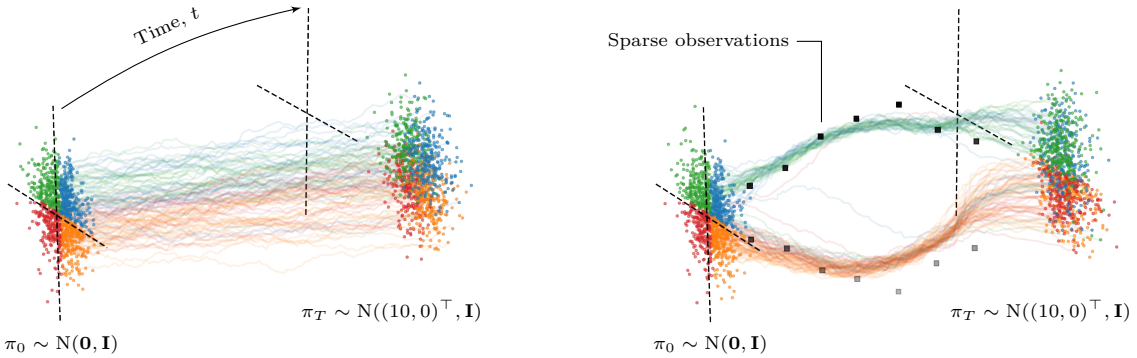
Abstract

The dynamic Schrödinger bridge problem provides an appealing setting for solving constrained time-series data generation tasks posed as optimal transport problems. It consists of learning non-linear diffusion processes using efficient iterative solvers. Recent works have demonstrated state-of-the-art results (*e.g.*, in modelling single-cell embryo RNA sequences or sampling from complex posteriors) but are limited to learning bridges with only initial and terminal constraints. Our work extends this paradigm by proposing the Iterative Smoothing Bridge (ISB). We integrate Bayesian filtering and optimal control into learning the diffusion process, enabling the generation of constrained stochastic processes governed by sparse observations at intermediate stages and terminal constraints. We assess the effectiveness of our method on synthetic and real-world data generation tasks and we show that the ISB generalises well to high-dimensional data, is computationally efficient, and provides accurate estimates of the marginals at intermediate and terminal times.

1 Introduction

Generative diffusion models have gained increasing popularity and achieved impressive results in a variety of challenging application domains, such as computer vision (*e.g.*, Ho et al., 2020; Song et al., 2021a; Dhariwal & Nichol, 2021), reinforcement learning (*e.g.*, Janner et al., 2022), and time series modelling (*e.g.*, Rasul et al., 2021; Vargas et al., 2021; Tashiro et al., 2021; Park et al., 2022). Recent works have explored connections between denoising diffusion models and the dynamic Schrödinger bridge problem (SBP, *e.g.*, Vargas et al., 2021; De Bortoli et al., 2021; Shi et al., 2022) to adopt iterative schemes for solving the dynamic optimal transport problem more efficiently. The solution of the SBP then acts as a denoising diffusion model in finite time and is the closest in Kullback–Leibler (KL) divergence to the forward noising process of the noising model under marginal constraints. Data may then be generated by time reversal of the process, *i.e.*, through the denoising process.

In many applications, the interest is not purely in modelling transport between an initial and terminal state distribution. For example, in naturally occurring generative processes, we typically observe snapshots of realizations *along intermediate stages* of individual sample trajectories (see Fig. 1). Such problems arise in medical diagnosis (*e.g.*, tissue changes and cell growth), demographic modelling, environmental dynamics, and animal movement modelling—see Fig. 4 for modelling bird migration and wintering patterns. Recently, constrained optimal control problems have been explored by adding additional fixed path constraints (Maoutsa et al., 2020; Maoutsa & Opper, 2022) or modifying the prior processes (Fernandes et al., 2021). However,



(a) Unconstrained transport (Schrödinger bridge)

(b) Constrained transport (Iterative smoothing bridge)

Figure 1: Illustrative example transport between an initial unit Gaussian and a shifted unit Gaussian at the terminal time T. Unconstrained transport on the left and the solution constrained by sparse observations (■) on the right. Colour coding of the initial points is only for distinguishing the paths.

defining meaningful fixed path constraints or prior processes for the optimal control problems can be challenging, while sparse observational data are accessible in many real-world applications.

In this work, we propose the *Iterative Smoothing Bridge* (ISB), an iterative method for learning a dynamical system for time-series data generation under constraints on both the initial and terminal distribution and sparse observational data constraints. The sparse observational constraints act as a way to encourage the paths sampled from the transport process to lie close to the observed data points. We perform the conditioning by leveraging the iterative pass idea from the Iterative Proportional Fitting procedure (IPFP, see Kullback, 1968; De Bortoli et al., 2021) and applying differentiable particle filtering (Reich, 2013; Corenflos et al., 2021) within the outer loop. Integrating sequential Monte Carlo methods (*e.g.*, Doucet et al., 2001; Chopin & Papaspiliopoulos, 2020) into the IPFP framework in such a way is non-trivial and can be understood as a novel iterative version of the algorithm by Maoutsa & Opper (2022) but with more general marginal constraints and additional path constraints defined by data.

We summarize the contributions as follows. *(i)* We propose a novel method for learning a dynamical model where the terminal constraints match a bridge problem and additional constraints are placed in form of sparse observations, inspired by optimal transport approaches. *(ii)* Thereof, we utilize the strong connections between the constrained bridging problem and particle filtering in sequential Monte Carlo, extending those links from pure inference to learning. Additionally, *(iii)* we demonstrate practical efficiency and show that the iterative smoothing bridge approach scales to high-dimensional data.

1.1 Related Work

Schrödinger bridges The problem of learning a stochastic process moving samples from one distribution to another can be posed as a type of transport problem known as a dynamic Schrödinger bridge problem (SBP, *e.g.*, Schrödinger, 1932; Léonard, 2014), where the marginal densities of the stochastic process are desired to resemble a given reference measure. In machine learning literature, the problem has been studied through learning the drift function of the dynamical system (De Bortoli et al., 2021; Wang et al., 2021; Vargas et al., 2021; Bunne et al., 2022; Shi et al., 2023). When an SDE system also defines the reference measure, the bridge problem becomes a constrained optimal control problem (*e.g.*, Caluya & Halder, 2022; 2021; Chen et al., 2021; Liu et al., 2022), which has been leveraged in learning Schrödinger bridges by Tianrong Chen (2022) through forward–backward SDEs. Moreover, neural stochastic control has been studied in Zhang et al. (2022). An optimal control problem with both initial and terminal distribution constraints and a fixed path constraint has been studied in Maoutsa et al. (2020) and Maoutsa & Opper (2022), where particle filtering is applied to continuous path constraints but the boundary constraints are defined by a single point. Maoutsa

(2023) studies a problem setting where stochastic dynamics are inferred based on sparse observation, but with geometric constraints rather than in the form of a Schrödinger bridge problem.

Diffusion models in machine learning The recent advances in diffusion models in machine learning literature have been focused on generating samples from complex distributions defined by data through transforming samples from an easy-to-sample distribution by a dynamical system (*e.g.*, Ho et al., 2020; Song et al., 2021b;a; Nichol & Dhariwal, 2021). The concept of reversing SDE trajectories via score-based learning (Hyvärinen & Dayan, 2005; Vincent, 2011) has allowed for models scalable enough to be applied to high-dimensional data sets directly in the data space. In earlier work, score-based diffusion models have been applied to problems where the dynamical system itself is of interest, for example, for the problem of time series amputation in Tashiro et al. (2021), inverse problems in imaging in Song et al. (2022) and for importance sampling Doucet et al. (2022). Interpreting the diffusion modelling problem as optimal control has recently been studied in Berner et al. (2022). Other dynamical models parametrized by neural networks have been applied to modelling latent time-series based on observed snapshots of dynamics (Rubanova et al., 2019; Li et al., 2020), but without further constraints on the initial or terminal distributions.

State-space models In their general form, state-space models combine a latent space dynamical system with an observation (likelihood) model. Evaluating the latent state distribution based on observational data can be performed by applying particle filtering and smoothing (Doucet et al., 2000) or by approximations of the underlying state distribution of a non-linear state-space model by a specific model family, for instance, a Gaussian (see Särkkä, 2013, for an overview). Speeding up parameter inference and learning in state-space models has been widely studied (*e.g.*, Schön et al., 2011; Svensson & Schön, 2017; Kokkala et al., 2014). Particle smoothing can be connected to Schrödinger bridges via the two-filter smoother (*e.g.*, Bresler, 1986; Briers et al., 2009; Hostettler, 2015), where the smoothing distribution is estimated by performing filtering both forward from the initial constraint and backwards from the terminal constraint. We refer to Mitter (1996) and Todorov (2008) for a more detailed discussion on the connection of stochastic control and filtering and to Chopin & Papaspiliopoulos (2020) for an introduction to particle filters.

Data Assimilation methods Data assimilation (DA) methods leverage techniques from state-space literature to 'assimilate' observations into a mechanistic model in order to inform the model dynamics based on measurements (*e.g.*, Asch et al., 2016; Wang et al., 2000). Approaches based on DA have found wide-spread use in scientific applications and have been extended to incorporate sparse observational data, for example, in numerical weather predictions (Whitaker et al., 2009), modelling cell state evolution in epithelial-mesenchymal transitions (Mendez et al., 2020), or in oceanographic scenarios (Beiser et al., 2023). A crucial difference to our work is that DA relies on a precise mechanistic model while our approach is data-driven, providing additional flexibility in modelling scenarios where formulating precise model dynamics is not possible.

2 Background

Let $\mathcal{C} = C([0,T], \mathbb{R}^d)$ denote the space of continuous functions from $[0,T]$ to $\mathbb{R}^d$ and let $\mathcal{B}(\mathcal{C})$ denote the Borel σ-algebra on $\mathcal{C}$. Let $\mathscr{P}(\pi_0, \pi_T)$ denote the space of probability measures on $(\mathcal{C}, \mathcal{B}(\mathcal{C}))$ such that the marginals at $0, T$ coincide with probability densities π_0 and π_T, respectively. The KL divergence from measure $\mathbb{Q}$ to measure $\mathbb{P}$ is written as $\mathrm{D}_{\mathrm{KL}}\left[\mathbb{Q} \,\|\, \mathbb{P}\right]$, where we assume that $\mathbb{Q} \ll \mathbb{P}$. For modelling the time dynamics, we assume a (continuous-time) state-space model consisting of a non-linear latent Itô SDE (see, *e.g.*, Øksendal, 2003; Särkkä & Solin, 2019) in $[0,T] \times \mathbb{R}^d$ with drift function $f_\theta(\cdot)$ and diffusion function $g(\cdot)$, and a Gaussian observation model, *i.e.*,

$$\mathbf{x}_0 \sim \pi_0, \quad \mathrm{d}\mathbf{x}_t = f_\theta(\mathbf{x}_t, t)\,\mathrm{d}t + g(t)\,\mathrm{d}\boldsymbol{\beta}_t, \tag{1}$$

and $\mathbf{y}_k \sim \mathrm{N}(\mathbf{y}_k \,|\, \mathbf{x}_t, \sigma^2\,\mathbf{I}_d)\,\big|_{t=t_k}$ where the drift function $f_\theta : \mathbb{R}^d \times [0,T] \to \mathbb{R}^d$ is a mapping modelled by a neural network (NN) parameterized by $\theta \in \Theta$, diffusion $g : [0,T] \to \mathbb{R}$ and $\boldsymbol{\beta}_t$ denotes standard d-dimensional Brownian motion. $\mathbf{x}_t$ denotes the latent stochastic process and $\mathbf{y}_t$ denotes the observation-space process. In practice, we consider the continuous-discrete time setting, where the process is observed at discrete time instances t_k such that observational data can be given in terms of a collection of input–output pairs $\{(t_j, \mathbf{y}_j)\}_{j=1}^M$.

2.1 Schrödinger Bridges and Optimal Control

The Schrödinger bridge problem (SBP, Schrödinger, 1932; Léonard, 2014) is an entropy-regularized optimal transport problem where the optimality is measured through the KL divergence from a reference measure $\mathbb{P}$ to the posterior $\mathbb{Q}$, with fixed initial and final densities π_0 and π_T, *i.e.*,

$$\min_{\mathbb{Q}\in\mathscr{P}(\pi_0,\pi_T)} \mathrm{D}_{\mathrm{KL}}\left[\mathbb{Q}\,\|\,\mathbb{P}\right]. \tag{2}$$

In this work, we consider only the case where the measures $\mathbb{P}$ and $\mathbb{Q}$ are constructed as the marginals of an SDE, *i.e.*, $\mathbb{Q}_t$ is the probability measure of the marginal of the SDE in Eq. (1) at time t, whereas $\mathbb{P}_t$ corresponds to the probability measure of the marginal of a reference SDE $\mathrm{d}\mathbf{x}_t = f(\mathbf{x}_t, t)\,\mathrm{d}t + g(t)\,\mathrm{d}\boldsymbol{\beta}_t$, at time t, where we call f the reference drift. Under the optimal control formulation of the SBP (Caluya & Halder, 2021) the KL divergence in Eq. (2) reduces to

$$\mathbb{E}\left[\int_0^T \frac{1}{2g(t)^2}\|f_\theta(\mathbf{x}_t,t) - f(\mathbf{x}_t,t)\|^2\,\mathrm{d}t\right], \tag{3}$$

where the expectation is over paths from Eq. (1). Rüschendorf & Thomsen (1993) and Ruschendorf (1995) showed that a solution to the SBP can be obtained by iteratively solving two half-bridge problems using the Iterative Proportional Fitting procedure (IPFP) for $l = 0, 1, \ldots, L$ steps,

$$\mathbb{Q}_{2l+1} = \operatorname*{arg\,min}_{\mathbb{Q}\in\mathscr{P}(\cdot,\pi_T)} \mathrm{D}_{\mathrm{KL}}\left[\mathbb{Q}\,\|\,\mathbb{Q}_{2l}\right] \quad \text{and} \quad \mathbb{Q}_{2l+2} = \operatorname*{arg\,min}_{\mathbb{Q}\in\mathscr{P}(\pi_0,\cdot)} \mathrm{D}_{\mathrm{KL}}\left[\mathbb{Q}\,\|\,\mathbb{Q}_{2l+1}\right], \tag{4}$$

where $\mathbb{Q}_0$ is set as the reference measure, and $\mathscr{P}(\pi_0,\cdot)$ and $\mathscr{P}(\cdot,\pi_T)$ denote the sets of probability measures with only either the marginal at time 0 or time T coinciding with π_0 or π_T, respectively. Recently, the IPFP to solving Schrödinger bridges has been adapted as a machine learning problem (Bernton et al., 2019; Vargas et al., 2021; De Bortoli et al., 2021). In practice, the interval $[0, T]$ is discretized and the forward drift f_θ and the backward drift b_ϕ of the corresponding reverse-time process (Haussmann & Pardoux, 1986; Föllmer, 1988) are modelled by NNs. Under the Gaussian transition approximations, each step in the discrete-time diffusion model can be reversed by applying an objective based on mean-matching.

3 Methods

Given an initial and terminal distribution π_0 and π_T, we are interested in learning a data-conditional bridge between π_0 and π_T. Let $\mathcal{D} = \{(t_j, \mathbf{y}_j)\}_{j=1}^M$ be a set of M sparsely observed values, *i.e.*, only a few or no observations are made at each point in time and let the state-space model of interest be given by Eq. (1). Note that we deliberately use $(t_j, \mathbf{y}_j)$ (instead of $(t_k, \mathbf{y}_k)$) to highlight that we allow for multiple observations at the same time point t_k. Our aim is to find a parameterization of the drift function f_θ such that evolving N particles $\mathbf{x}_t^i$, with $\mathbf{x}_0^i \sim \pi_0$ (with $i = 1, 2, \ldots, N$), according to Eq. (1) will result in samples $\mathbf{x}_T^i$ from the terminal distribution π_T. Inspired by the IPFP by De Bortoli et al. (2021), which decomposes the SBP into finding two half-bridges, we propose to iteratively solve two modified half-bridge problems where the additional sparse observations are accounted for simultaneously. For this, let

$$\mathrm{d}\mathbf{x}_t = f_{l,\theta}(\mathbf{x}_t, t)\,\mathrm{d}t + g(t)\,\mathrm{d}\boldsymbol{\beta}_t, \qquad \mathbf{x}_0 \sim \pi_0, \tag{5}$$

$$\mathrm{d}\mathbf{z}_t = b_{l,\phi}(\mathbf{z}_t, t)\,\mathrm{d}t + g(t)\,\mathrm{d}\hat{\boldsymbol{\beta}}_t, \qquad \mathbf{z}_0 \sim \pi_T, \tag{6}$$

denote the forward and backward SDE at iteration $l = 1, 2, \ldots, L$, where $\hat{\boldsymbol{\beta}}_t$ is the reverse-time Brownian motion. For simplicity, we denote $\boldsymbol{\beta}_t = \hat{\boldsymbol{\beta}}_t$ when the direction of the SDE is clear.

To learn the Iterative Smoothing Bridge dynamics, we iteratively employ the following steps: 1 evolve *forward* particle trajectories according to Eq. (5) with drift $f_{l-1,\theta}$ and filter w.r.t. the observations $\{(t_j, \mathbf{y}_j)\}_{j=1}^M$, 2 learn the drift function $b_{l,\phi}$ for the reverse-time SDE, 3 evolve *backward* particle trajectories according to Eq. (6) with the drift $b_{l,\phi}$ learned in step 2 and filter w.r.t. the observations $\{(t_j, \mathbf{y}_j)\}_{j=1}^M$, and 4 learn the drift function $f_{l,\theta}$ for the forward SDE based on the backward particles. Fig. 2 illustrates the forward and

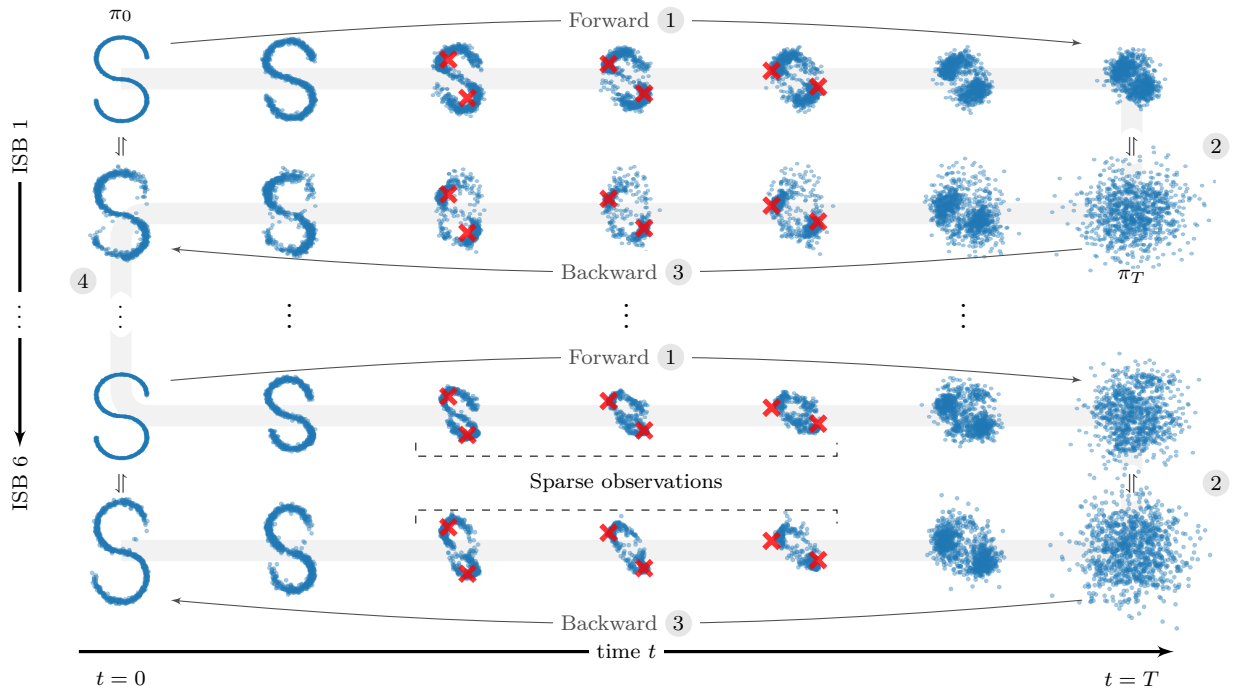


Figure 2: Sketch of a diffusion bridge between a 2D data distribution (π_0) and an isotropic Gaussian (π_T) constrained by sparse observations (✖). The forward diffusion at the first iteration (ISB 1) learns to account for the sparse observations but does not converge to the correct terminal distribution ($t = T$), and the backward diffusion *vice versa*. After iterating (ISB 6), the forward and backward diffusions converge to the correct targets and are able to account for the sparse observational data.

backward process of our iterative scheme for a data-conditioned denoising diffusion bridge. Next, we will go through steps 1–4 in detail and introduce the Iterative Smoothing Bridge method for data-conditional diffusion bridges.

3.1 The Iterative Smoothing Bridge

The Iterative Smoothing Bridge (ISB) method iteratively generates particle filtering trajectories (steps 1 and 3 in Fig. 2) and learns the parameterizations of the forward and backward drift functions $f_{l,\theta}$ and $b_{l,\phi}$ (steps 2 and 4) by minimizing a modified version of the mean-matching objective presented by De Bortoli et al. (2021). Note that steps 2 and 4 are dependent on applying differential resampling in the particle filtering steps 1 and 3 for reversing the generated trajectories. We will now describe the forward trajectory generating step 1 and the backward drift learning step 2 in detail. Steps 3 and 4 are given by application of 1 and 2 on their reverse-time counterparts.

Step 1 (and 3): Given a fixed discretization of the time interval $[0, T]$ denoted as $\{t_k\}_{k=1}^K$ with $t_1 = 0$ and $t_K = T$, denote the time step lengths as $\Delta_k = t_{k+1} - t_k$. By truncating the Itô–Taylor series of the SDE, we can consider an Euler–Maruyama (*e.g.*, Ch. 8 in Särkkä & Solin, 2019) type of discretization for the continuous-time problem. We give the time-update of the ith particle at time t_k evolved according to Eq. (5), *i.e.*,

$$\tilde{\mathbf{x}}_{t_k}^i = \mathbf{x}_{t_{k-1}} + f_{l-1,\theta}(\mathbf{x}_{t_{k-1},t_{k-1}})\Delta_k + g(t_{k-1})\sqrt{\Delta_k}\,\boldsymbol{\xi}_k^i, \tag{7}$$

where $\boldsymbol{\xi}_k^i \sim \mathrm{N}(\mathbf{0}, \mathbf{I})$. Notice that we have not yet conditioned on the observational data. In step 3, the particles $\tilde{\mathbf{z}}_{t_k}^i$ of the backward SDE Eq. (6) are similarly obtained. The SDE dynamics sampled in steps 1 and 3 apply the learned drift functions $f_{l-1,\theta}$ and $b_{l,\phi}$ from the previous step and do not require sampling from the underlying SDE model. For times t_k at which no observations are available, we set $\mathbf{x}_t^i = \tilde{\mathbf{x}}_t^i$ (and

$\mathbf{z}^i_{t_k} = \tilde{\mathbf{z}}^i_{t_k}$ respectively) and otherwise compute the particle filtering weights $w^i_{t_k}$ based on the observations $\{(t_j, \mathbf{y}_j) \in \mathcal{D} \,|\, t_j = t_k\}$ for resampling. See Sec. 3.2 for details on the particle filtering.

For resampling, we employ a *differentiable resampling* procedure, where the particles and weights $(\tilde{\mathbf{x}}^i_{t_k}, w^i_{t_k})$ are transported to uniformly weighted particles $(\mathbf{x}^i_{t_k}, \frac{1}{N})$ by solving an entropy-regularized optimal transport problem (Cuturi, 2013; Peyré & Cuturi, 2019; Corenflos et al., 2021) (see App. D). Through application of the ε-regularized optimal transport map $\mathbf{T}_{(\varepsilon)} \in \mathbb{R}^{N \times N}$ (see Corenflos et al., 2021) the particles are resampled via the map to $\mathbf{x}^i_{t_k} = \tilde{\mathbf{X}}^\top_{t_k} \mathbf{T}_{(\varepsilon),i}$, where $\tilde{\mathbf{X}}_{t_k} \in \mathbb{R}^{N \times d}$ denotes the stacked particles $\{\tilde{\mathbf{x}}^i_{t_k}\}_{i=1}^N$ at time t_k before resampling.

Step 2 (and 4): Given the particles $\{\mathbf{x}^i_{t_k}\}_{k=1,i=1}^{K,N}$, we now aim to learn the drift function for the respective reverse-time process. The purpose of this step is to find a mean-matching reversal of the trajectories, in other words we aim to find $f_{l,\theta}$ such that it best explains the change we observe from $\{\mathbf{x}^i_{t_k}\}$ to $\{\mathbf{x}^i_{t_{k+1}}\}$ for each trajectory $i = 1, 2, \ldots, N$ and particle $k = 1, 2, \ldots, N$. We will review the loss functions used for this optimization step, where the loss outside observation times will match the approach in De Bortoli et al. (2021) and the loss at observation times is motivated by a smoothing of trajectories (see App. A.2 for a discussion).

In case no observation is available at time t_k, we apply the mean-matching loss based on a Gaussian transition approximation proposed in De Bortoli et al. (2021):

$$\ell^i_{k+1,\text{nobs}} = \|b_{l,\phi}(\mathbf{x}^i_{t_{k+1}}, t_{k+1})\Delta_k - \mathbf{x}^i_{t_{k+1}} - f_{l-1,\theta}(\mathbf{x}^i_{t_{k+1}}, t_k)\Delta_k + \mathbf{x}^i_{t_k} + f_{l-1,\theta}(\mathbf{x}^i_{t_k}, t_k)\Delta_k\|^2. \tag{8}$$

In case an observation is available at time t_k the particle values $\tilde{\mathbf{X}}_{t_k}$ will be coupled through the optimal transport map. Therefore, the transition density is a sum of Gaussian variables (see App. A for details and a derivation), and the mean-matching loss is therefore given by:

$$\ell^i_{k+1,\text{obs}} = \|b_{l,\phi}(\mathbf{x}^i_{t_{k+1}}, t_{k+1})\Delta_k - \mathbf{x}^i_{t_{k+1}} - f_{l-1,\theta}(\mathbf{x}^i_{t_{k+1}}, t_k)\Delta_k + \textstyle\sum_{n=1}^N T_{(\varepsilon),i,n}\left(\mathbf{x}^n_{t_k} + f_{l-1,\theta}(\mathbf{x}^n_{t_k}, t_k)\Delta_k\right)\|^2. \tag{9}$$

The derivation of Eq. (9) relies on mean-matching the trajectories as in De Bortoli et al. (2021) combined with applying the differentiable resampling optimal transport map $\mathbf{T}_{(\varepsilon)}\tilde{\mathbf{x}}^i_{t_k} = \mathbf{x}^i_{t_k}$ on all the particles to obtain the transition density $p_{\mathbf{x}_{t_k} \,|\, \mathbf{x}^i_{t_{k-1}}}(\mathbf{x}_{t_k})$ at observation times, resulting in a Gaussian distribution dependent on all the particles. In addition, we apply the property that the reverse drift should satisfy

$$b_{l,\phi}(\mathbf{x}_{t_{k+1}}, t_{k+1}) = f_{l-1,\theta}(\mathbf{x}_{t_{k+1}}, t_k) - g(t_{k+1})^2 \nabla \ln p_{t_{k+1}}, \tag{10}$$

where $p_{t_{k+1}}$ is the particle filtering density after differential resampling at time t_{k+1}. Thus the impact of observations to the loss function is two-fold, the observations define the value of the transport matrix $\mathbf{T}_{(\varepsilon)}$ and the marginal score $\nabla \ln p_{t_{k+1}}$. The use of the reverse drift Eq. (10) is further motivated by the smoothing reverse presented in App. A.2, where we discuss how Eq. (10) matches a backwards controlled drift.

The overall objective function is a combination of both loss functions, with the respective mean-matching loss depending on whether t_k is an observation time. The final loss function is written as:

$$\ell(\phi) = \sum_{i=1}^N \left[\sum_{k=1}^K \ell^i_{k,\text{obs}}(\phi)\mathbb{I}_{y_{t_k} \neq \emptyset} + \ell^i_{k,\text{nobs}}(\phi)\mathbb{I}_{y_{t_k} = \emptyset}\right], \tag{11}$$

where $\mathbb{I}_{\text{cond.}}$ denotes an indicator function that returns '1' iff the condition is true, and '0' otherwise. Consequently, the parameters ϕ of $b_{l,\phi}$ are learned by minimizing Eq. (11) through gradient descent. In practice, a cache of trajectories $\{\mathbf{x}^i_{t_k}\}_{k=1,i=1}^{K,N}$ is maintained through training of the drift functions, and refreshed at a fixed number of inner loop iterations, as in De Bortoli et al. (2021), avoiding differentiation over the SDE generation computational graph. The calculations for step 4 follow similarly. We present a high-level description of the ISB steps in Alg. 1.

The learned backward drift $b_{l,\phi}$ can be interpreted as an analogy of the backward drift in Maoutsa & Opper (2022), connecting our approach to solving optimal control problems through Hamilton–Jacobi equations, see App. A.2 for an analysis of the backwards SDE and the control objective. While we are generally considering problem settings where the number of observations is low, we propose that letting $M \to \infty$ yields the underlying marginal distribution, see Prop. 2 in App. A.3.

Algorithm 1 The Iterative Smoothing Bridge

Input: Marginal constraints (π_0, π_T), observations $\mathcal{D} = \{(t_j, \mathbf{y}_j)\}_{j=1}^M$, initial drift function $f_{0,\theta}$, iterations L, discretization steps K, number of particles N, observation noise schedule $\kappa(l)$

Output: Learned forward and backward drift (f_θ, b_ϕ)

for $l = 1$ **to** L **do**

— Forward process —

Initialize forward particles $\{\mathbf{x}_0^i\}_{i=1}^N \sim \pi_0$

for $k = 1$ **to** K **do**

Generate $\{\mathbf{x}_k^i\}_{i=1}^N$ using $\{\mathbf{x}_{k-1}^i\}_{i=1}^N$ ▷ Eq. (5)

if Observations at t_k **then**

$\{\mathbf{x}_k^i\}_{i=1}^N \leftarrow$ **DiffResample**$(\{\mathbf{x}_k^i\}_{i=1}^N, \kappa(l))$

end if

end for

Optimize the forward loss function w.r.t. ϕ ▷ Eq. (11)

— Backward process —

Initialize backward particles $\{\mathbf{z}_K^i\}_{i=1}^N \sim \pi_T$

for $k = K$ **to** 1 **do**

Generate $\{\mathbf{z}_{k-1}^i\}_{i=1}^N$ using $\{\mathbf{z}_k^i\}_{i=1}^N$ ▷ Eq. (6)

if Observations at t_k **then**

$\{\mathbf{z}_{k-1}^i\}_{i=1}^N \leftarrow$ **DiffResample**$(\{\mathbf{z}_{k-1}^i\}_{i=1}^N, \kappa(l))$

end if

end for

Optimize the backwards loss function w.r.t. θ ▷ Eq. (15)

end for

3.2 Computational Considerations

The ISB algorithm is a generic approach to learn data-conditional diffusion bridges under various choices of, *e.g.*, the particle filter proposal density or the reference drift. Next, we cover practical considerations for the implementation of the method and highlight the model choices in the experiments.

Multiple observations per time step Naturally, we can make more than one observation at a single point in time t_k, denoted as $\mathcal{D}_{t_k} = \{(t_j, \mathbf{y}_j) \in \mathcal{D} \,|\, t_j = t_k\}$. To compute particle weights $w_{t_k}^i$ for the ith particle we consider only the H-nearest neighbours of $\mathbf{x}_{t_k}^i$ in $\mathcal{D}_{t_k}$ instead of all observations in $\mathcal{D}_{t_k}$. By restricting to the H-nearest neighbours, denoted as $\mathcal{D}_{t_k}^H$, we introduce an additional locality to the proposal density computation, which can be helpful in the case of multimodality. On the other hand, letting $H > 1$ results in weights which take into account the local density of the observations, not only the distance to the nearest neighbour. In experiments with few observations, we set $H = 1$, the choice of H is discussed when we have set the value higher.

Particle filtering proposal The proposal density chosen for the ISB is the bootstrap filter, where the proposal matches the Gaussian transition density $p(\mathbf{x}_{t_k} \,|\, \mathbf{x}_{t_{k-1}})$. Assuming a Gaussian noise model $\mathrm{N}(\mathbf{0}, \sigma^2\mathbf{I})$, the unnormalized log-weights for the ith particle at time t_k are given by $\log w_{t_k}^i = -1/2\sigma^2 \sum_{\mathbf{y}_j \in \mathcal{D}_{t_k}^H} \|\mathbf{x}_{t_k}^i - \mathbf{y}_j\|^2$. While we restrict our approach in practice to the bootstrap filter, applying more sophisticated proposals such as in the auxiliary particle filter (Pitt & Shephard, 1999) could improve the results in some cases, although restricting the problem to a linear observation model.

Observational noise schedule In practice, using a constant observation noise variance σ^2 can result in an iterative scheme which does not have a stationary point as $L \to \infty$. Even if the learned drift function $f_{l,\theta}$ was optimal, the filtering steps 1 and 3 would alter the trajectories unless all particles would have uniform weights. Thus, we introduce a noise schedule $\kappa(l)$ which ensures that the observation noise increases in the number of ISB iterations, causing ISB to converge to the IPFP (De Bortoli et al., 2021) as $L \to \infty$. We found that letting the observation noise first decrease and then increase (in the spirit of simulated annealing) often outperformed a strictly increasing observation noise schedule. The noise schedule is studied in App. C, where we derive the property that letting $L \to \infty$ yields IPFP.

Drift initialization Depending on the application, one may choose to incorporate additional information by selecting an appropriate initial drift. A possible choice includes a pre-trained neural network drift learned to

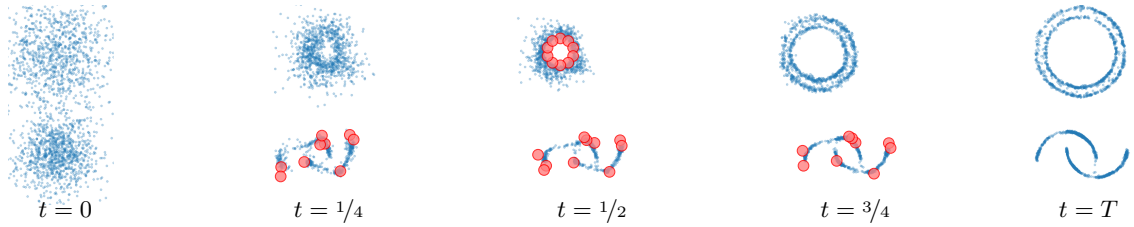


Figure 3: 2D toy experiments from scikit-learn with both cases starting from a Gaussian: The TWO CIRCLES (top) and TWO MOONS (bottom) data sets, with observations (red markers) constraining the problem. For the circles, the 10 circular observations at $t = 0.5$ first force the method to create a circle that then splits into two; in the lower plot the observations at $t \in [0.25, 0.5, 0.75]$ split the data into clusters before joining them into two moons. See Fig. 6 in the Appendix for the IPFP result.

transport π_0 to π_T without accounting for observations. However, starting from a drift for the unconstrained SBP can be problematic in cases where the observations are far away from the unconstrained bridge. To encourage exploration, one may choose $f_0 = 0$ for the initial drift. In various problem settings, we found a zero drift and starting from the SBP to be successful in the experiments. See App. C for discussion.

4 Experiments

To assess the properties and performance of the ISB, we present a range of experiments that demonstrate how the iterative learning procedure can incorporate both observational data and terminal constraints. We start with simple examples that build intuition (*cf.* Fig. 1 and Fig. 2) and show standard ML benchmark tasks. For quantitative assessment, we design an experiment with a non-linear SDE for which the marginal distributions are available in closed-form. Finally, we demonstrate our model both in a highly multimodal bird migration task, conditioned image generation, and in a single-cell embryo RNA modelling problem. Ablation studies are found in App. C.

Experiment setup In all experiments, the forward and backward drift functions f_θ and b_ϕ are parametrized as neural networks. For low-dimensional experiments, we apply the MLP block design as in De Bortoli et al. (2021), and for the image experiment an U-Net as in Nichol & Dhariwal (2021). The latent state SDE was simulated by Euler–Maruyama with a fixed time-step of 0.01 over 100 steps and 1000 particles if not otherwise stated. All low-dimensional (at most $d = 5$) experiments were run on a MacBook Pro laptop CPU, whereas the image experiments used a single NVIDIA A100 GPU and ran for 5 h 10 min. Notice that since ISB only performs particle filtering outside the stochastic gradient training loop, the training runtime is in the same order as in the earlier Schrödinger bridge image generation experiments of De Bortoli et al. (2021). Thus we omit any wall-clock timings. Full details for all the experiments are included in App. B.

All experiment settings include a number of hyperparameter choices, some typical to all diffusion problems and some specific to particle filtering and smoothing. The diffusion $g(t)$ is a pre-determined function not optimized during training. We divide the experiments into two main subsets: problems of 'sharpening to achieve a data distribution' and 'optimal transport problems'. In the former, the initial distribution has a support overlapping with the terminal distribution and the process noise level $g(t)$ goes from high to low as time progresses. Conversely, in the latter setting, the particles sampled from the initial distribution must travel to reach the support of the terminal distribution, and we chose to use a constant process noise level. Perhaps the most significant choice of hyperparameter is the observational noise level, as it imposes a preference on how closely should the observational points be followed, see App. C.1 for details.

2D toy examples We show illustrative results for the TWO MOONS and CIRCLES from scikit-learn. We add artificial observation data to bias the processes. For the circles, the observational data consists of 10 points, spaced evenly on the circle. The points are all observed simultaneously, at halfway through the process, forcing the marginal density of the generating SDE to collapse to the small circle, and then to expand. For the TWO MOONS, the observational data is collected from 10 trajectories of a diffusion model, which generates the two moons from

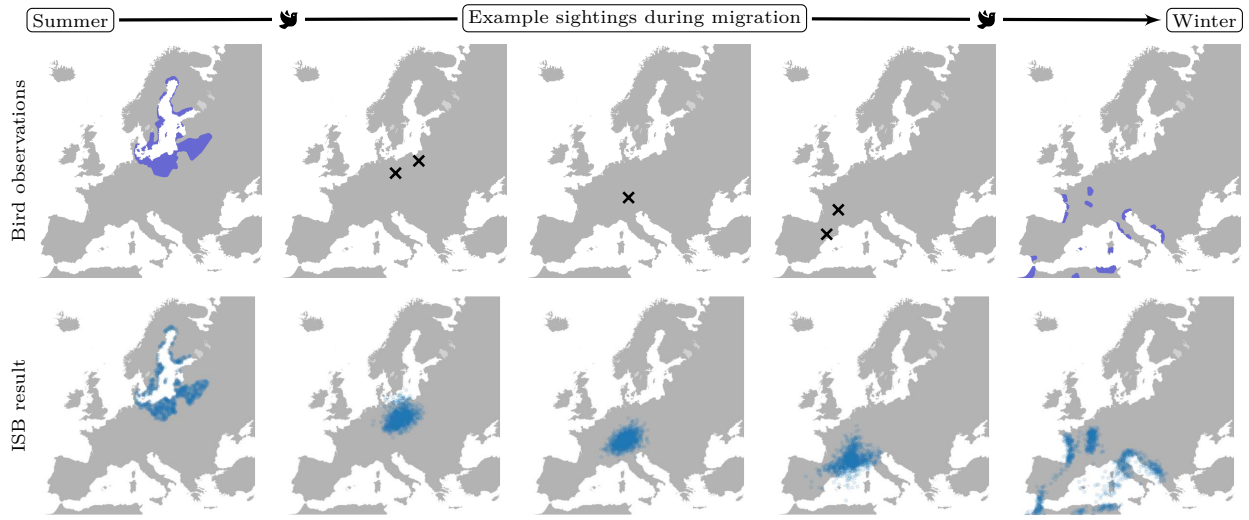


Figure 4: Bird migration example. The top row describes nesting and wintering areas and example sightings during migration. The bottom shows the marginal densities of the ISB model from the initial to terminal distribution, matching bird sightings along the migration.

noise, and these 10 trajectories are then observed at three points in time. Results are visualized in Fig. 3 (see videos in supplement). For reference, we have included plots of the IPFP dynamics in the supplement, see Fig. 6.

Quantitative comparison on the Beneš SDE In order to quantify how observing a process in between its initial and terminal states steers the ISB model to areas with higher likelihood, we test its performance on a Beneš SDE model (see, *e.g.* Särkkä & Solin, 2019). The Beneš SDE is a non-linear one-dimensional SDE of form $\mathrm{d}x_t = \tanh(x_t)\,\mathrm{d}t + \mathrm{d}\beta_t$ with $x_0 = 0$, but its marginal density is available in closed-form, allowing for negative log-likelihood evaluation. We simulate trajectories from the Beneš SDE and from the reverse drift and stack the reversed trajectories. The terminal distribution is shifted and scaled so that the Beneš SDE itself does not solve the transport problem from π_0 to π_T, see App. B.2 for details and visualizations of the processes.

We fit a Schrödinger bridge model with no observational data as a baseline, using the Beneš SDE drift as the reference model. The ISB model is initialized with a zero-drift model (not with the Beneš as reference), thus making learning more challenging. We compare the models in terms of negative log predictive density in Table 1, where we see that the ISB model captures the process well on average (over the entire time-horizon) and at selected marginal times.

Bird migration Bird migration can be seen as a regular seasonal transport problem, where birds move (typically North–South) along a flyway, between breeding and wintering grounds. We take this as a motivating example of constrained optimal transport, where the geographical and constraints and preferred routes are accounted for by bird sighting data (see Fig. 4 top). By adapting data from Ambrosini et al. (2014) and Pellegrino et al. (2015), we propose a simplified data set for geese migration in Europe (OIBMD: ornithologically implausible bird migration data; available in the supplement). We applied the ISB for 12 iterations, with a linear observation noise schedule from 1 to 0.2, and constant diffusion noise 0.05. The drift function was initialized as a zero-function, and thus the method did not rely on a separately fit model optimized for generating the wintering distribution based on the breeding distribution. For comparison, we include the Schrödinger bridge results in App. B.3.

Constraining an image generation process We demonstrate that the ISB approach scales well to high-dimensional inputs by studying a proof-of-concept image generation task. We modify the diffusion generative process of the MNIST (LeCun et al., 1998) digit 8 by artificial observations steering the dynamical system in the middle of the generation process. While the concept of observations in case of image generation is somewhat unnatural, it showcases the scalability of the method to high-dimensional data spaces. Here, the

Table 1: Results for the Beneš experiment. We report the negative log predictive density (NLPD, lower better) of the Beneš marginal likelihood over generated particles at the initial and terminal distributions and at the middle of the transport process.

Method	NLPD Average	Middle	End
Schrödinger B	4.787	3.565	0.1919
Iterative smoothing B	**3.557**	**2.985**	**0.1567**

Table 2: Results for the single-cell embryo RNA experiment. We compare ISB to TrajectoryNet, IPML, and our implementation of IPFP. Unlike the other methods, our model is able to utilize the intermediate data distributions while training.

Method	Earth mover's distance t=0	t=1	t=2	t=3	t=T
TrajectoryNet	0.62	1.15	1.49	1.26	0.99
IPML	**0.34**	1.13	1.35	1.01	**0.49**
IPFP (no obs)	0.57	1.53	1.86	1.32	0.85
ISB (single-cell obs)	0.57	**1.04**	**1.24**	**0.94**	0.83

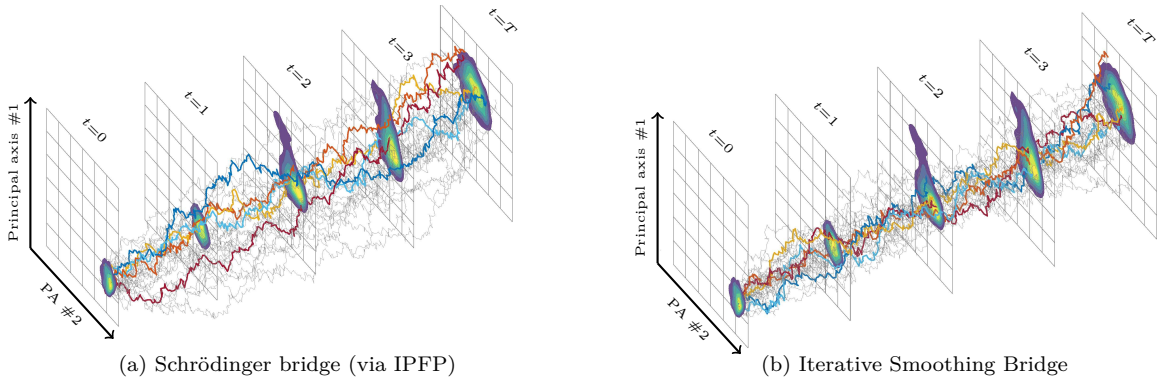
drift is initialized using a pre-trained neural network obtained by first running a Schrödinger bridge model for image generation. The process is then given an observation in the form of a bottom-half of a MNIST digit 8 in the middle of the dynamical process. As the learned model uses information from the observation both before and after the observation time, the lower half of the image is sharper than the upper half. We provide further details on this experiment and sampled trajectories in App. B.4, and an ablation of multi-modal MNIST image generation in App. B.5

Single-cell embryo RNA-seq Lastly, we evaluated our approach on an Embryoid body scRNA-seq time course (Tong et al., 2020). The data consists of RNA measurements collected over five time ranges from a developing human embryo system. No trajectory information is available, instead we only have access to snapshots of RNA data. This leads to a data set over 5 time ranges, the first from days 0–3 and the last from days 15–18. In the experiment, we followed the protocol by Tong et al. (2020), reduced the data dimensionality to $d = 5$ using PCA, and used the first and last time ranges as the initial and terminal constraints. All other time ranges are considered observational data. Contrary to the other experiments, intermediate data are imprecise (only a time range of multiple days is known) but abundant.

We learned the ISB using a zero drift and compared it against an unconditional bridge obtained through the IPFP (De Bortoli et al., 2021)—see Fig. 5. The ISB learns to generate trajectories with marginals closer to the observed data while performing comparably to the IPFP at the initial and terminal stages. This improvement is also verified numerically in Table 2, showing that the ISB obtains a lower Earth mover's distance between the generated marginals and the observational data than IPFP. Additionally, Table 2 lists the performance of previous works that do not use the intermediate data during training (Tong et al., 2020) or only use it to construct an informative reference drift (Vargas et al., 2021), see App. B.6 for details. In both cases, ISB outperforms the other approaches w.r.t. the intermediate marginal distributions ($t = 1, 2, 3$), while IPML (Vargas et al., 2021) outperforms ISB at the initial and terminal stages due to its data-driven reference drift. Notice that while we reduced the dimensionality via PCA to 5 for fair comparisons to Vargas et al. (2021), the ISB model would also allow modelling the full state-space model, with observations in the high-dimensional gene space and a latent SDE.

5 Discussion and Conclusion

The dynamic Schrödinger bridge problem provides an appealing setting for posing optimal transport problems as learning non-linear diffusion processes and enables efficient iterative solvers. However, while recent works have state-of-the-art performance in many complex application domains, they are typically limited to learning bridges with only initial and terminal constraints dependent on observed data. In this work, we have extended this paradigm and introduced the Iterative Smoothing Bridge (ISB), an iterative algorithm for generating data-conditional smoothing bridges. For this, we leveraged the strong connections between the constrained bridging problem and particle filtering in sequential Monte Carlo, extending them from pure inference to learning. We thoroughly assessed the applicability and flexibility of our approach in various experimental settings, including synthetic data sets and complex real-world scenarios (*e.g.*, bird migration, conditional image generation, and modelling single-cell RNA-sequencing time-series). Our experiments showed that ISB



(a) Schrödinger bridge (via IPFP) (b) Iterative Smoothing Bridge

Figure 5: Illustration of the trajectories of the single-cell experiment for the Schrödinger bridge (a) and the ISB (b), projected onto the first two principal components. The first five trajectories are highlighted in colour, and intermediate observation densities visualized as slices.

generalizes well to high-dimensional data, is computationally efficient, and provides accurate estimates of the marginals at initial, terminal, and intermediate times.

Accurately modelling the dynamics of complex systems under both path constraints induced by sparse observations and initial and terminal constraints is a key challenge in many application domains. These include biomedical applications, demographic modelling, and environmental dynamics, but also machine learning specific applications such as reinforcement learning, planning, and time-series modelling. All these applications have in common that the dynamic nature of the problem is driven by the progression of time, and not only the progression of a generative process as often is the case in, *e.g.*, generative image models. Thus, constraints over intermediate stages have a natural role and interpretation in this wider set of dynamic diffusion modelling applications. We believe the proposed ISB algorithm opens up new avenues for diffusion models in relevant real-world modelling tasks and will be stimulating for future work. Recent work suggest close connections between sequential DA, commonly applied in many real-world scientific domains, and the Schrödinger bridge problem (Reich, 2019) further emphasising the potential for future scientific applications of our work by exploiting these links. Moreover, in the future more sophisticated observational models, alternative strategies to account for multiple observations, and different noise schedules could be explored. Lastly, the proposed approach could naturally be extended to other types of optimal transport problems, such as the Wasserstein barycenter, a frequently employed case of the multi-marginal optimal transport problem.

A reference implementation of the ISB model can be found at `https://github.com/AaltoML/iterative-smoothing-bridge`.

Acknowledgements and Disclosure of Funding

Authors acknowledge funding from the Academy of Finland (grants 339730, 324345, and 347279). We also acknowledge the computational resources provided by the Aalto Science-IT project and CSC – IT Center for Science, Finland. We wish to thank Adrien Corenflos for sharing an implementation of differentiable resampling in PyTorch, and Prakhar Verma for comments on the manuscript.

References

Roberto Ambrosini, Riccardo Borgoni, Diego Rubolini, Beatrice Sicurella, Wolfgang Fiedler, Franz Bairlein, Stephen R. Baillie, Robert A. Robinson, Jacquie A. Clark, Fernando Spina, and Nicola Saino. Modelling the progression of bird migration with conditional autoregressive models applied to ringing data. *PLoS ONE*, 9(7):1–10, 07 2014.

Mark Asch, Marc Bocquet, and Maëlle Nodet. *Data Assimilation: Methods, Algorithms, and Applications*. SIAM, 2016.

Florian Beiser, Håvard Heitlo Holm, and Jo Eidsvik. Comparison of ensemble-based data assimilation methods for sparse oceanographic data. *arXiv preprint arXiv:2302.07197*, 2023.

Julius Berner, Lorenz Richter, and Karen Ullrich. An optimal control perspective on diffusion-based generative modeling. In *NeurIPS 2022 Workshop on Score-Based Methods*, 2022.

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

Yoram Bresler. Two-filter formulae for discrete-time non-linear Bayesian smoothing. *International Journal of Control*, 43(2):629–641, 1986.

Mark Briers, Arnaud Doucet, and Simon Maskell. Smoothing algorithms for state–space models. *Annals of the Institute of Statistical Mathematics*, 62:61–89, 02 2009.

Charlotte Bunne, Ya-Ping Hsieh, Marco Cuturi, and Andreas Krause. Recovering stochastic dynamics via Gaussian Schrödinger bridges. *arXiv preprint arXiv:2202.05722*, 2022.

Kenneth F. Caluya and Abhishek Halder. Reflected Schrödinger bridge: Density control with path constraints. In *American Control Conference, ACC 2021*, pp. 1137–1142. IEEE, 2021.

Kenneth F. Caluya and Abhishek Halder. Wasserstein proximal algorithms for the Schrödinger bridge problem: Density control with nonlinear drift. *IEEE Transactions on Automatic Control*, 67(3):1163–1178, 2022.

Yongxin Chen, Tryphon T. Georgiou, and Michele Pavon. Stochastic control liaisons: Richard Sinkhorn meets Gaspard Monge on a Schrödinger Bridge. *SIAM Review*, 63(2):249–313, 2021.

Nicolas Chopin and Omiros Papaspiliopoulos. *An Introduction to Sequential Monte Carlo*. Springer, 2020.

Adrien Corenflos, James Thornton, George Deligiannidis, and Arnaud Doucet. Differentiable particle filtering via entropy-regularized optimal transport. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021*, volume 139 of *Proceedings of Machine Learning Research*, pp. 2100–2111. PMLR, 2021.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems 26 (NIPS)*, pp. 2292–2300. Curran Associates, Inc., 2013.

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 34 (NeurIPS)*, pp. 17695–17709. Curran Associates, Inc., 2021.

Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems 35 (NeurIPS)*, pp. 8780–8794. Curran Associates, Inc., 2021.

Arnaud Doucet, Simon Godsill, and Christophe Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000.

Arnaud Doucet, Nando De Freitas, and Neil James Gordon. *Sequential Monte Carlo methods in practice*. Statistics for Engineering and Information Science. Springer, 2001.

Arnaud Doucet, Will Sussman Grathwohl, Alexander G. D. G. Matthews, and Heiko Strathmann. Score-based diffusion meets annealed importance sampling. In *Advances in Neural Information Processing Systems 35*, 2022.

David Lopes Fernandes, Francisco Vargas, Carl Henrik Ek, and Neill DF Campbell. Shooting Schrödinger's cat. In *Proceedings of the Fourth Symposium on Advances in Approximate Bayesian Inference (AABI)*, 2021.

Hans Föllmer. Random fields and diffusion processes. In *École d'Été de Probabilités de Saint-Flour XV–XVII, 1985–87*, pp. 101–203. Springer, 1988.

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

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, pp. 6840–6851. Curran Associates, Inc., 2020.

Roland Hostettler. A two filter particle smoother for Wiener state-space systems. In *2015 IEEE Conference on Control Applications (CCA)*, pp. 412–417, 2015.

Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4):695–709, 2005.

Michael Janner, Yilun Du, Joshua B. Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato (eds.), *International Conference on Machine Learning (ICML)*, volume 162, pp. 9902–9915. PMLR, 2022.

Juho Kokkala, Arno Solin, and Simo Särkkä. Expectation maximization based parameter estimation by sigma-point and particle smoothing. In *Proceedings of the 17th International Conference on Information Fusion (FUSION)*, pp. 1–8, 2014.

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

Yann LeCun, Corinna Cortes, and Christopher J.C. Burges. The MNIST database of handwritten digits, 1998. URL `http://yann.lecun.com/exdb/mnist/`.

Xuechen Li, Ting-Kam Leonard Wong, Ricky T. Q. Chen, and David Duvenaud. Scalable gradients for stochastic differential equations. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 108 of *Proceedings of Machine Learning Research*, pp. 3870–3882. PMLR, 2020.

Guan-Horng Liu, Tianrong Chen, Oswin So, and Evangelos Theodorou. Deep generalized schrödinger bridge. In *Advances in Neural Information Processing Systems 35*, 2022.

Christian Léonard. A survey of the Schrödinger problem and some of its connections with optimal transport. *Discrete & Continuous Dynamical Systems*, 34(4):1533–1574, 2014.

Dimitra Maoutsa. Geometric constraints improve inference of sparsely observed stochastic dynamics. *arXiv preprint arXiv:2304.00423*, 2023.

Dimitra Maoutsa and Manfred Opper. Deterministic particle flows for constraining stochastic nonlinear systems. *Physical Review Research*, 4(4), 2022.

Dimitra Maoutsa, Sebastian Reich, and Manfred Opper. Interacting particle solutions of Fokker–Planck equations through gradient–log–density estimation. *Entropy*, 22:802, 07 2020.

Mario J Mendez, Matthew J Hoffman, Elizabeth M Cherry, Christopher A Lemmon, and Seth H Weinberg. Cell fate forecasting: A data-assimilation approach to predict epithelial-mesenchymal transition. *Biophysical Journal*, 118(7):1749–1768, 2020.

S.K. Mitter. Filtering and stochastic control: a historical perspective. *IEEE Control Systems Magazine*, 16 (3):67–76, 1996.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8162–8171. PMLR, 2021.

Bernt Øksendal. *Stochastic Differential Equations: An Introduction with Applications*. Springer, New York, NY, sixth edition, 2003.

Sung Woo Park, Kyungjae Lee, and Junseok Kwon. Neural markov controlled SDE: Stochastic optimization for continuous-time data. In *International Conference on Learning Representations (ICLR)*, 2022.

Irene Pellegrino, Marco Cucco, Arne Follestad, and Mathieu Boos. Lack of genetic structure in greylag goose (Anser anser) populations along the European Atlantic flyway. *PeerJ*, 3:e1161, 2015.

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

Michael K. Pitt and Neil Shephard. Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, 94(446):590–599, 1999.

Kashif Rasul, Abdul-Saboor Sheikh, Ingmar Schuster, Urs M. Bergmann, and Roland Vollgraf. Multivariate probabilistic time series forecasting via conditioned normalizing flows. In *International Conference on Learning Representations (ICLR)*, 2021.

Sebastian Reich. A nonparametric ensemble transform method for Bayesian inference. *SIAM Journal on Scentific Computing*, 35, 2013.

Sebastian Reich. Data assimilation: The Schrödinger perspective. *Acta Numerica*, 28:635–711, 2019.

Yulia Rubanova, Tian Qi Chen, and David K Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. In *Advances in Neural Information Processing Systems 32 (NeurIPS)*, pp. 5321–5331. Curran Associates, Inc., 2019.

Ludger Ruschendorf. Convergence of the iterative proportional fitting procedure. *The Annals of Statistics*, 23 (4):1160–1174, 1995.

L. Rüschendorf and W. Thomsen. Note on the Schrödinger equation and I-projections. *Statistics & Probability Letters*, 17(5):369–375, 1993.

Simo Särkkä. *Bayesian Filtering and Smoothing*. Cambridge University Press, Cambridge, UK, 2013.

Simo Särkkä and Arno Solin. *Applied Stochastic Differential Equations*. Cambridge University Press, Cambridge, UK, 2019.

E. Schrödinger. Sur la théorie relativiste de l'électron et l'interprétation de la mécanique quantique. *Annales de l'institut Henri Poincaré*, 2(4):269–310, 1932.

Thomas B. Schön, Adrian Wills, and Brett Ninness. System identification of nonlinear state-space models. *Automatica*, 47(1):39–49, 2011.

Yuyang Shi, Valentin De Bortoli, George Deligiannidis, and Arnaud Doucet. Conditional simulation using diffusion Schrödinger bridges. In *38th Conference on Uncertainty in Artificial Intelligence*. UAI, 2022.

Yuyang Shi, Valentin De Bortoli, Andrew Campbell, and Arnaud Doucet. Diffusion schrödinger bridge matching. *arXiv preprint arXiv:2303.16852*, 2023.

Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. In *Advances in Neural Information Processing Systems 35 (NeurIPS)*, pp. 1415–1428. Curran Associates, Inc., 2021a.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations (ICLR)*, 2021b.

Yang Song, Liyue Shen, Lei Xing, and Stefano Ermon. Solving inverse problems in medical imaging with score-based generative models. In *International Conference on Learning Representations (ICLR)*, 2022.

Andreas Svensson and Thomas B. Schön. A flexible state–space model for learning nonlinear dynamical systems. *Automatica*, 80:189–199, 2017.

Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. CSDI: Conditional score-based diffusion models for probabilistic time series imputation. In *Advances in Neural Information Processing Systems 35 (NeurIPS)*, pp. 24804–24816. Curran Associates, Inc., 2021.

Evangelos A. Theodorou Tianrong Chen, Guan-Horng Liu. Likelihood training of schrödinger bridge using forward-backward SDEs theory. In *International Conference on Learning Representations (ICLR)*, 2022.

Emanuel Todorov. General duality between optimal control and estimation. In *Proceedings of the 47th IEEE Conference on Decision and Control*, pp. 4286–4292, 2008.

Alexander Tong, Jessie Huang, Guy Wolf, David van Dijk, and Smita Krishnaswamy. Trajectorynet: A dynamic optimal transport network for modeling cellular dynamics. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9526–9536. PMLR, 2020.

Francisco Vargas, Pierre Thodoroff, Austen Lamacraft, and Neil Lawrence. Solving Schrödinger bridges via maximum likelihood. *Entropy*, 23(9):1134, 2021.

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

Bin Wang, Xiaolei Zou, and Jiang Zhu. Data assimilation and its applications. *Proceedings of the National Academy of Sciences*, 97(21):11143–11144, 2000.

Gefei Wang, Yuling Jiao, Qian Xu, Yang Wang, and Can Yang. Deep generative learning via Schrödinger bridge. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pp. 10794–10804. PMLR, 2021.

Jeffrey S Whitaker, Gilbert P Compo, and Jean-Noël Thépaut. A comparison of variational and ensemble-based data assimilation systems for reanalysis of sparse observations. *Monthly Weather Review*, 137(6): 1991–1999, 2009.

Jingdong Zhang, Qunxi Zhu, and Wei Lin. Neural stochastic control. In *Advances in Neural Information Processing Systems 35*, 2022.

A Method Details

We present the details of the objective function derivation in App. A.1 and explain the connection of the backward drift function to Hamilton–Jacobi equations in App. A.2. In App. A.3, we discuss the behaviour of our model at the limit $M \to \infty$, that is, when the observations fully represent the marginal densities of the stochastic process.

A.1 Deriving the Mean-matching Loss at Observation Times

Recall that the forward loss is written as

$$\ell(\phi) = \sum_{i=1}^{N} \left[\sum_{k=1}^{K} \ell^i_{k,\text{obs}}(\phi) \mathbb{I}_{y_{t_k} \neq \emptyset} + \ell^i_{k,\text{nobs}}(\phi) \mathbb{I}_{y_{t_k} = \emptyset} \right], \tag{12}$$

where the loss at observations $\ell^i_{k,\text{obs}}(\phi)$ and loss elsewhere $\ell^i_{k,\text{nobs}}(\phi)$ are

$$\ell^i_{k+1,\text{nobs}} = \| b_{l,\phi}(\mathbf{x}^i_{t_{k+1}}, t_{k+1})\Delta_k - \mathbf{x}^i_{t_{k+1}} - f_{l-1,\theta}(\mathbf{x}^i_{t_{k+1}}, t_k)\Delta_k + \mathbf{x}^i_{t_k} + f_{l-1,\theta}(\mathbf{x}^i_{t_k}, t_k)\Delta_k \|^2, \tag{13}$$

$$\begin{aligned} \ell^i_{k+1,\text{obs}} = \| & b_{l,\phi}(\mathbf{x}^i_{t_{k+1}}, t_{k+1})\Delta_k - \mathbf{x}^i_{t_{k+1}} - f_{l-1,\theta}(\mathbf{x}^i_{t_{k+1}}, t_k)\Delta_k \\ & + \frac{1}{C_{\varepsilon,i}} \textstyle\sum_{n=1}^{N} T_{(\varepsilon),i,n} \left(\mathbf{x}^n_{t_k} + f_{l-1,\theta}(\mathbf{x}^n_{t_k}, t_k)\Delta_k \right) \|^2, \end{aligned} \tag{14}$$

For convenience, we state the backward loss functions which follow similarly to their forward versions. The backward loss is defined as

$$\overleftarrow{\ell(\theta)} = \sum_{i=1}^{N} \left[\sum_{k=1}^{K} \overleftarrow{\ell}^{\,i}_{k,\text{obs}}(\theta) \mathbb{I}_{y_{t_k} \neq \emptyset} + \overleftarrow{\ell}^{\,i}_{k,\text{nobs}}(\theta) \mathbb{I}_{y_{t_k} = \emptyset} \right], \tag{15}$$

where the loss at observations $\overleftarrow{\ell}^{\,i}_{k,\text{obs}}(\theta)$ and loss elsewhere $\overleftarrow{\ell}^{\,i}_{k,\text{nobs}}(\theta)$ are

$$\overleftarrow{\ell}^{\,i}_{k+1,\text{nobs}} = \| f_{l,\theta}(\mathbf{x}^i_{t_{k+1}}, t_{k+1})\Delta_k - \mathbf{x}^i_{t_{k+1}} - b_{l,\theta}(\mathbf{x}^i_{t_{k+1}}, t_k)\Delta_k + \mathbf{x}^i_{t_k} + b_{l,\theta}(\mathbf{x}^i_{t_k}, t_k)\Delta_k \|^2, \tag{16}$$

$$\begin{aligned} \overleftarrow{\ell}^{\,i}_{k+1,\text{obs}} = \| & f_{l,\theta}(\mathbf{x}^i_{t_{k+1}}, t_{k+1})\Delta_k - \mathbf{x}^i_{t_{k+1}} - b_{l,\phi}(\mathbf{x}^i_{t_{k+1}}, t_k)\Delta_k \\ & + \frac{1}{C_{\varepsilon,i}} \textstyle\sum_{n=1}^{N} T_{(\varepsilon),i,n} \left(\mathbf{x}^n_{t_k} + b_{l,\phi}(\mathbf{x}^n_{t_k}, t_k)\Delta_k \right) \|^2. \end{aligned} \tag{17}$$

Proposition 1. *Define the forward SDE as*

$$\mathrm{d}\mathbf{x}_t = f_{l,\theta}(\mathbf{x}_t, t)\,\mathrm{d}t + g(t)\,\mathrm{d}\boldsymbol{\beta}_t, \qquad \mathbf{x}_0 \sim \pi_0, \tag{18}$$

and a backward SDE drift as

$$b_{l,\phi}(\mathbf{x}_{t_{k+1}}, t_{k+1}) = f_{l-1,\theta}(\mathbf{x}_{t_{k+1}}, t_k) - g(t_{k+1})^2 \nabla \ln p_{t_{k+1}}, \tag{19}$$

where $p_{t_{k+1}}$ is the particle filtering density after differential resampling at time t_{k+1}. Then $b_{l,\phi}(\mathbf{x}_{t_{k+1}}, t_{k+1})$ minimizes the loss function

$$\begin{aligned} \ell^i_{k+1,obs} = \| & b_{l,\phi}(\mathbf{x}^i_{t_{k+1}}, t_{k+1})\Delta_k - \mathbf{x}^i_{t_{k+1}} - f_{l-1,\theta}(\mathbf{x}^i_{t_{k+1}}, t_k)\Delta_k \\ & + \frac{1}{C_{\varepsilon,i}} \textstyle\sum_{n=1}^{N} T_{(\varepsilon),i,n} \left(\mathbf{x}^n_{t_k} + f_{l-1,\theta}(\mathbf{x}^n_{t_k}, t_k)\Delta_k \right) \|^2, \end{aligned} \tag{20}$$

where we denote $\mathbf{C}_{\varepsilon,i} = \frac{1}{g(t_{k+1})^2 \Delta_k} \operatorname{Var}\left(\sum_{n=1}^{N} T_{(\varepsilon),i,n} \tilde{\mathbf{x}}^n_{t_{k+1}} \right)$, and $\{\tilde{\mathbf{x}}^i_{t_{k+1}}\}_{i=1}^N$ are the particles before resampling.

Proof sketch. Our objective is to find a backward drift function $b_{l,\phi}(\mathbf{x}_{t_{k+1}}, t_{k+1})$ as in Eq. (19). Notice that at observation times t_k, this is not equivalent to finding the reverse drift of the SDE forward transition and differential resampling combined, since the drift function $f_{l-1,\theta}$ alone does not map the particles $\{\mathbf{x}^i_{t_k}\}_{i=1}^N$ to the particles $\{\mathbf{x}^i_{t_{k+1}}\}_{i=1}^N$. We will derive a loss function for learning the backward drift as in Eq. (19) below, leaving the discussion on why it is a meaningful choice of a backward drift to App. A.2. Our derivation closely follows the proof of Proposition 3 in De Bortoli et al. (2021), but we provide the details here for the sake of completeness.

First, we give the transition density $p_{\mathbf{x}_{t_k} \mid \mathbf{x}^i_{t_{k-1}}}(\mathbf{x}_k)$ and apply it to derive the observation time loss $\ell^i_{k,\text{obs}}$. The derivation for the loss $\ell^i_{k,\text{no obs}}$ is skipped since it is as in the proof of Proposition 3 in De Bortoli et al. (2021). Suppose that at t_k, there are observations. By definition, the particles before resampling $\{\tilde{\mathbf{x}}^i_{t_{k+1}}\}_{i=1}^N$ are generated by the Gaussian transition density

$$p(\tilde{\mathbf{x}}_{t_{k+1}} \mid \mathbf{x}^i_{t_k}) = \mathrm{N}(\tilde{\mathbf{x}}_{t_{k+1}} \mid \mathbf{x}^i_{t_k} + \delta_k f_l(\mathbf{x}^i_{t_k}, t_k), g(t_{k+1})^2 \Delta_k \mathbf{I}). \tag{21}$$

Recall that the resampled particles are defined as a weighted average of all the particles, $\mathbf{x}^i_{t_k} = \sum_{n=1}^N \tilde{\mathbf{x}}^n_{t_k} T_{(\varepsilon),i,n}$. Thus, the transition density from $\{\mathbf{x}^i_{t_k}\}_{i=1}^N$ to the particles $\{\mathbf{x}^i_{t_{k+1}}\}_{i=1}^N$ is also a Gaussian,

$$p(\mathbf{x}_{t^i_{k+1}} \mid \mathbf{x}^i_{t_k}) = \mathrm{N}(\tilde{\mathbf{x}}_{t_{k+1}} \mid \sum_{n=1}^N T_{(\varepsilon),i,n}(\mathbf{x}^n_{t_{k-1}} + \Delta_k f_{l-1,\theta}(\mathbf{x}^n_{t_k}, t_k)), g(t_{k+1})^2 \Delta_k C_{\varepsilon,i} \mathbf{I}_d). \tag{22}$$

We will derive the loss function Eq. (9) by modifying the mean matching proof in De Bortoli et al. (2021) by the transition mean Eq. (22) and the backward drift definition Eq. (19). Using the particle filtering approximation, the marginal density can be decomposed as $p_{t_{k+1}}(\mathbf{x}_{k+1}) = \sum_{i=1}^N p_{t_k}(\mathbf{x}^i_k) p_{\mathbf{x}_{k+1} \mid \mathbf{x}^i_k}(\mathbf{x}_{k+1})$. By substituting the transition density Eq. (22) it follows that

$$p_{t_{k+1}}(\mathbf{x}_{t_{k+1}}) = \frac{1}{Z} \sum_{i=1}^N p_{t_k}(\mathbf{x}^i_{t_k}) \exp\left(- \frac{\| \left(\sum_{n=1}^N T_{(\varepsilon),i,n}(\mathbf{x}^i_{t_k} + f_{l-1,\theta}(\mathbf{x}_{t_k}, t_k)) \right) - \mathbf{x}_{t_{k+1}} \|^2}{2 g(t_{k+1})^2 C_{\varepsilon,i} \Delta_k} \right), \tag{23}$$

where Z is the normalization constant of Eq. (22). As in the proof of Proposition 3 of De Bortoli et al. (2021), we derive an expression for the score function. Since $\nabla \ln p_{t_{k+1}}(\mathbf{x}_{t_{k+1}}) = \frac{\nabla_{\mathbf{x}_{t_{k+1}}} p_{t_{k+1}}(\mathbf{x}_{t_{k+1}})}{p_{t_{k+1}(\mathbf{x}_{t_{k+1}})}}$, we first manipulate $\nabla_{\mathbf{x}_{t_{k+1}}} p_{t_{k+1}}(\mathbf{x}_{t_{k+1}})$,

$$\begin{aligned}
\nabla_{\mathbf{x}_{t_{k+1}}} p_{t_{k+1}}(\mathbf{x}_{t_{k+1}}) &= \frac{1}{Z} \sum_{i=1}^N \nabla_{\mathbf{x}_{t_{k+1}}} p(\mathbf{x}^i_{t_k}) \exp\left(- \frac{\| \left(\sum_{n=1}^N T_{(\varepsilon),i,n}(\mathbf{x}^i_{t_k} + f_{l-1,\theta}(\mathbf{x}_{t_k}, t_k)) \right) - \mathbf{x}_{t_{k+1}} \|^2}{2 g(t_{k+1})^2 C_{\varepsilon,i} \Delta_k} \right) \\
&= \frac{1}{Z} \Bigg(\sum_{i=1}^N p(\mathbf{x}^i_{t_k}) \left(\sum_{n=1}^N \frac{1}{g(t_{k+1})^2 \Delta_k C_{\varepsilon,i}} \left(T_{(\varepsilon),i,n}(\mathbf{x}^i_{t_k} + f_{l-1,\theta}(\mathbf{x}_{t_k}, t_k)) - \mathbf{x}_{t_{k+1}} \right) \right) \\
&\qquad \exp\left(- \frac{\| \left(\sum_{n=1}^N T_{(\varepsilon),i,n}(\mathbf{x}^i_{t_k} + f_{l-1,\theta}(\mathbf{x}_{t_k}, t_k)) \right) - \mathbf{x}_{t_{k+1}} \|^2}{2 g(t_{k+1})^2 C_{\varepsilon,i} \Delta_k} \right) \Bigg).
\end{aligned} \tag{24}$$

Substituting $p_{t_k}(x^i_k) = \frac{p_{t_{k+1}}(\mathbf{x}_{t_{k+1}}) p_{\mathbf{x}_{k+1} \mid \mathbf{x}^i_k}(\mathbf{x}_{k+1})}{p_{\mathbf{x}^i_k \mid \mathbf{x}_{k+1}}(\mathbf{x}^i_k)}$ to the equation above gives

$$\nabla_{\mathbf{x}_{t_{k+1}}} p_{t_{k+1}}(\mathbf{x}_{t_{k+1}}) = p_{t_{k+1}}(\mathbf{x}_{t_{k+1}}) \sum_{i=1}^N p_{\mathbf{x}_{k+1} \mid \mathbf{x}^i_k}(\mathbf{x}^i_k) \left(\sum_{n=1}^N \frac{\left(T_{(\varepsilon),i,n}(\mathbf{x}^i_{t_k} + f_{l-1,\theta}(\mathbf{x}_{t_k}, t_k)) - \mathbf{x}_{t_{k+1}} \right)}{g(t_{k+1})^2 \Delta_k C_{\varepsilon,i}} \right), \tag{25}$$

and dividing by $p_{t_{k+1}(\mathbf{x}_{t_{k+1}})}$ yields

$$\nabla \ln p_{t_{k+1}}(\mathbf{x}_{t_{k+1}}) = \sum_{i=1}^N p_{\mathbf{x}_{t^i_k} \mid \mathbf{x}_{t_{k+1}}}(\mathbf{x}_{t^i_k}) \left(\sum_{n=1}^N \frac{\left(T_{(\varepsilon),i,n}(\mathbf{x}^i_{t_k} + f_{l-1,\theta}(\mathbf{x}_{t_k}, t_k)) - \mathbf{x}_{t_{k+1}} \right)}{g(t_{k+1})^2 \Delta_k C_{\varepsilon,i}} \right). \tag{26}$$

Substituting Eq. (26) to the definition of the optimal backward drift Eq. (19) gives

$$\begin{aligned} b_{l,\phi}(\mathbf{x}_{t_{k+1}}, t_{k+1}) &= f_{l-1,\theta}(\mathbf{x}_{t_{k+1}}, t_k) - g(t_{k+1})^2 \nabla \ln p_{t_{k+1}}(\mathbf{x}_{k+1}) \\ &= f_{l-1,\theta}(\mathbf{x}_{t_{k+1}}, t_k) \\ &\quad - g(t_{k+1})^2 \sum_{i=1}^{N} p_{\mathbf{x}_{t_k^i} \mid \mathbf{x}_{t_{k+1}}}(\mathbf{x}_{t_{k+1}}) \left(\sum_{n=1}^{N} \frac{\left(T_{(\varepsilon),i,n}(\mathbf{x}_{t_k}^i + f_{l-1,\theta}(\mathbf{x}_{t_k}, t_k)) - \mathbf{x}_{t_{k+1}}\right)}{g(t_{k+1})^2 \Delta_k C_{\varepsilon,i}} \right), \end{aligned} \tag{27}$$

where taking $f_{l-1,\theta}(\mathbf{x}_{t_{k+1}}, t_k)$ inside the sum yields

$$\begin{aligned} b_{l,\phi}(\mathbf{x}_{t_{k+1}}, t_{k+1}) = &\sum_{i=1}^{N} p_{\mathbf{x}_{t_k^i} \mid \mathbf{x}_{t_{k+1}}}(\mathbf{x}_{t_{k+1}}) \\ &\left(\frac{1}{C_{\varepsilon,i}} \left(\sum_{n=1}^{N} T_{(\varepsilon),i,n}(\mathbf{x}_{t_k}^i + f_{l-1,\theta}(\mathbf{x}_{t_k}, t_k)) \right) - \frac{\mathbf{x}_{t_{k+1}}}{C_{\varepsilon,i}} - \Delta_k f_{l-1,\theta}(\mathbf{x}_{t_{k+1}}, t_k) \right) / \Delta_k). \end{aligned} \tag{28}$$

Multiplying the equation above by Δ_k gives

$$\Delta_k b_{l,\phi}(\mathbf{x}_{t_{k+1}}^i, t_{k+1}) = \left(\sum_{n=1}^{N} T_{(\varepsilon),i,n}(\mathbf{x}_{t_k}^n + f_{l-1,\theta}(\mathbf{x}_{t_k}^n, t_k)) \right) - \frac{\mathbf{x}_{t_{k+1}}^i}{C_{\varepsilon,i}} - \Delta_k f_{l-1,\theta}(\mathbf{x}_{t_{k+1}^i}, t_k). \tag{29}$$

Thus we may set the objective for finding the optimal backward drift $b_{l,\phi}$ as

$$\begin{aligned} \ell_{k+1,\text{no obs}}^i = \| b_{l,\phi}(\mathbf{x}_{t_{k+1}}^i, t_{k+1}) \Delta_k - \frac{\mathbf{x}_{t_{k+1}}^i}{C_{\varepsilon,i}} - f_{l-1,\theta}(\mathbf{x}_{t_{k+1}}^i, t_k) \Delta_k \\ + \frac{1}{C_{\varepsilon,i}} \textstyle\sum_{n=1}^{N} T_{(\varepsilon),i,n} \left(\mathbf{x}_{t_k}^n + f_{l-1,\theta}(\mathbf{x}_{t_k}^n, t_k) \Delta_k \right) \|^2. \end{aligned} \tag{30}$$

□

Notice that if the weights before resampling are uniform, then $T_{(\varepsilon)} = \mathbf{I}_N$, and for all $i \in 1, 2, \ldots, N$ it holds that $C_{\varepsilon,i} = 1$, since all but one of the terms in the sum $\frac{1}{g(t_{k+1})^2} \mathrm{Var}\left(\sum_{n=1}^{N} T_{(\varepsilon),i,n} \tilde{\mathbf{x}}_{t_{k+1}}^n \right)$ vanish. Similarly, for one-hot weights $C_{\varepsilon,i} = 1$. In practice, we set the constant $C_{\varepsilon,i} = 1$ as in Eq. (9) and observe good empirical performance with the simplified loss function.

A.2 Connection to Hamilton–Jacobi Equations

We connect the backward drift function $b_{l,\phi}(\mathbf{x}_{t_{k+1}}, t_{k+1}) = f_{l-1,\theta}(\mathbf{x}_{t_k+1}, t_k) - g(t_{k+1})^2 \nabla \ln p_{t_{k+1}}(\mathbf{x}_{t_{k+1}})$ to the Hamilton–Jacobi equations for stochastic control through following the setting of Maoutsa & Opper (2022), which applies the drift $f_{l-1,\theta}(\mathbf{x}_t, t) - g(t)^2 \nabla \ln p_t(\mathbf{x}_t)$ for a backwards SDE initialized at π_T.

Consider a stochastic control problem with a path constraint $U(\mathbf{x}_t, t)$, optimizing the following loss function,

$$\mathcal{J} = \frac{1}{N} \sum_{i=1}^{N} \int_{t=0}^{T} \frac{1}{2g(t)^2} \| f_\theta(\mathbf{x}_t^i, t) - f(\mathbf{x}_t^i, t) \|^2 + U(\mathbf{x}_t^i, t) \, \mathrm{d}t - \ln \chi(\mathbf{x}_T^i), \tag{31}$$

with the paths, $\mathbf{x}_t^i$ sampled as trajectories from the SDE

$$\mathbf{x}_0 \sim \pi_0, \quad \mathrm{d}\mathbf{x}_t = f_{l-1,\theta}(\mathbf{x}_t, t) \, \mathrm{d}t + g(t) \, \mathrm{d}\boldsymbol{\beta}_t, \tag{32}$$

and the loss $\ln \chi(\mathbf{x}_T^i)$ measures distance from the distribution π_T. Since we set the path constraint via observational data, our method resembles setting $U(\mathbf{x}_t^i, t) = 0$ when t is not an observation time, and $U(\mathbf{x}_t^i) = -\log \mathbf{p}(\mathbf{y} \mid \mathbf{x}_t^i)$, where $\mathbf{p}(\mathbf{y} \mid \mathbf{x}_t^i)$ is the observation model.

Let $q_t(\mathbf{x})$ denote the marginal density of the controlled (drift f_θ) SDE at time t. In Maoutsa & Opper (2022), the marginal density is decomposed as

$$q_t(\mathbf{x}) = \varphi_t(\mathbf{x}) p_t(\mathbf{x}), \tag{33}$$

where $\varphi_t(\mathbf{x})$ is a solution to a backwards Fokker-Planck-Kolmogorov (FPK) partial differential equation starting from $\varphi_T(\mathbf{x}) = \pi_T$, and the density evolves as in

$$\frac{\mathrm{d}\varphi_t(\mathbf{x})}{\mathrm{d}t} = -\mathcal{L}_f^\dagger \varphi_t(\mathbf{x}) + U(\mathbf{x}, t)\varphi_t(\mathbf{x}), \tag{34}$$

where $\mathcal{L}_f^\dagger$ is the adjoint FPK operator to the uncontrolled system. The density $p_t(\mathbf{x})$ corresponds to the forward filtering problem, initialized with π_0,

$$\frac{\mathrm{d}p_t(\mathbf{x})}{\mathrm{d}t} = \mathcal{L}_f(p_t(\mathbf{x})) - U(\mathbf{x}, t)p_t(\mathbf{x}), \tag{35}$$

where $\mathcal{L}_f$ is the FPK operator of the uncontrolled SDE (with drift f). The particle filtering trajectories $\{\mathbf{x}_{t_k}\}^i$ generated in our method are samples from the density defined by Eq. (35). In the context of our method, the path constraint matches the log-weights of particle filtering at observation times and is zero elsewhere.

In Maoutsa & Opper (2022), a backward evolution for q_t is applied, using the backwards time $\tilde{q}_{T-\tau}(\mathbf{x}) = q_\tau(\mathbf{x})$, yielding a backwards SDE starting from $\tilde{q}_0(\mathbf{x}) = \{\mathbf{x}_T^i\}_{i=1}^N$, reweighted according to π_T. The backward samples from $\tilde{q}$ are generated following the SDE dynamics

$$\mathrm{d}\mathbf{x}_\tau^i = (f(\mathbf{x}_\tau^i, T-\tau) + g(t)^2 \nabla \ln p_{T-\tau}(\mathbf{x}_\tau^i)\,\mathrm{d}t + g(t)\,\mathrm{d}\beta_\tau. \tag{36}$$

We have thus selected the backward drift $b_{l,\phi}$ to match the drift of $\tilde{q}_t(x)$, the backward controlled density. Intuitively, our choice of $b_{l,\phi}$ is a drift which generates the smoothed particles when initialized at $\{\mathbf{x}_T^i\}_{i=1}^N$, the terminal state of the forward SDE. The discrepancy between π_T and the distribution induced by $\{\mathbf{x}_T^i\}_{i=1}^N$ then motivates the use of an iterative scheme after learning to simulate from $q_t(x)$.

A.3 Observing the Full Marginal Density

Suppose that at time t_k, we let the number of observations grow unbounded. We analyse the behaviour of our model at the resampling step, at the limit $M \to \infty$ for the number of observations and $\sigma \to 0$ for the observation noise. When applying the bootstrap proposal, recall that we combined the multiple observations to compute the log-weights as

$$\log w_{t_k}^i = -\frac{1}{2\sigma^2} \sum_{\mathbf{y}_j \in \mathcal{D}_{i,t_k}^H} \|\mathbf{x}_{t_k}^i - \mathbf{y}_j\|^2, \tag{37}$$

which works well in practice for the sparse-data settings we have considered. Below we analyse the behaviour of an alternative way to combine the weights and show that given an infinite number of observations, it creates samples from the true underlying distribution.

Proposition 2. *Let $\{\mathbf{x}_{t_k}^i\}_{i=1}^N$ be a set of particles and $\{\mathbf{y}_j\}_{j=1}^M$ the observations at time t_k. Assume that the observations have been sampled from a density ρ_{t_k} and that for all i it holds that $\mathbf{x}_{t_k}^i \in \mathrm{supp}(\rho_{t_k})$. Define the particle weights as*

$$\log w_{t_k,\sigma,M}^i = \log\left(\frac{1}{Z|\mathcal{D}_{i,t_k}^{H(M)}|} \sum_{\mathbf{y}_j \in \mathcal{D}_{i,t_k}^{H(M)}} \exp(-\|\mathbf{x}_{t_k}^i - \mathbf{y}_j\|^2/2\sigma^2)\right), \tag{38}$$

where Z is the normalization constant of the observation model Gaussian $p(\mathbf{y} \,|\, \mathbf{x}_{t_k}^i)$. Then for each particle $\mathbf{x}_{t_k}^i$, its weight satisfies

$$\lim_{\sigma \to 0} \lim_{M \to \infty} w_{t_k,\sigma,M}^i = \rho_{t_k}(x_{t_k}). \tag{39}$$

Proof sketch. We drop the σ and $H(M)$ from the weight notation for simplicity of notation, but remark that the particle filtering weights are dependent on both quantities. Consider the number of particles N fixed, and denote the d-dimensional sphere centered at $\mathbf{x}_{t_k}^i$ as $B(\mathbf{x}_{t_k}^i, r)$. Since each particle $\mathbf{x}_{t_k}^i$ lies in the support of the true underlying marginal density ρ_{t_k}, then for any radius $r > 0$ such that $B(\mathbf{x}_{t_k}^i, r) \in \mathrm{supp}(\rho_{t_k})$, and

$H > 0$, we may choose M high enough so that the points $\mathbf{y}_j \in \mathcal{D}^H_{i,t_k}$ satisfy $\mathbf{y}_j \in B(\mathbf{x}^i_{t_k}, r)$. It follows from Eq. (38) that

$$w^i_{t_k} = \frac{1}{Z|\mathcal{D}^{H(M)}_{i,t_k}|} \sum_{\mathbf{y}_j \in \mathcal{D}^{H(M)}_{i,t_k}} \exp(-\|\mathbf{x}^i_{t_k} - \mathbf{y}_j\|^2/2\sigma^2). \tag{40}$$

For any $r > 0$ and with observation noise $\sigma = cr$, we may set $c, H(M)$ so that the sum above approximates the integral

$$w^i_{r,t_k} \approx \frac{1}{|B(\mathbf{x}^i_{t_k}, r)|} \int_{B(\mathbf{x}^i_{t_k}, r)} p(\mathbf{y} \,|\, \mathbf{x}^i_{t_k}) \rho_t(\mathbf{y}) \, \mathrm{d}\mathbf{y}. \tag{41}$$

By applying the Lebesque differentiation theorem, we obtain that for almost every $\mathbf{x}^i_{t_k}$, we have $\lim_{r\to 0} w^i_{t_k,r} = \rho_{t_k}(\mathbf{x}^i_{t_k})$, since as $\sigma \to 0$, the density $p(\mathbf{y} \,|\, \mathbf{x}^i_{t_k})$ collapses to the Dirac delta of $\mathbf{x}^i_{t_k}$. □

Prop. 2 can be interpreted as the infinite limit of a kernel density estimate of the true underlying distribution. Resampling accurately reweights the particles so that the probability of resampling particle $\mathbf{x}^i_{t_k}$ is proportional to the density ρ_{t_k} compared to the other particles. Notice that the result does not guarantee that the particles will cover the support of ρ_{t_k}, since we did not assume that the drift initialization generates a marginal density at time t_k covering its support.

B Experimental Details

B.1 2D Toy Data Sets

For the constrained transport problem for two-dimensional scikit-learn, the observational data we chose to use was different for each of the three data sets presented; TWO MOONS, TWO CIRCLES and the S-shape. All three experiments had the same discretization ($t \in [0, 0.99]$), $\Delta_k = 0.01$), learning rate 0.001, and differentiable resampling regularization parameter $\varepsilon = 0.01$. The process noise $g(t)^2$ follows a linear schedule from 0.001 to 1, with low noise at time $t = 0$ and high noise at $t = 0.99$, and each iteration of the ISB method trains the forward and backward drift networks each for 5000 iterations, with batch size 256. When running on a Macbook Pro CPU, it took approximately 6 minutes to complete for the TWO CIRCLES experiment for instance, while the exact runtime varies based on factors such as the number of observations and the number of ISB iterations required. Other hyperparameters are explained below.

Two moons The observational data consists of 10 points selected from the Schrödinger bridge trajectories, all observed at $t \in [0.25, 0.5, 0.75]$ with an exponential observation noise schedule $\kappa(l) = 1.25^{l-1}$. The ISB was run for 6 epochs and initialized with a drift from the pre-trained Schrödinger bridge model from the unconstrained problem.

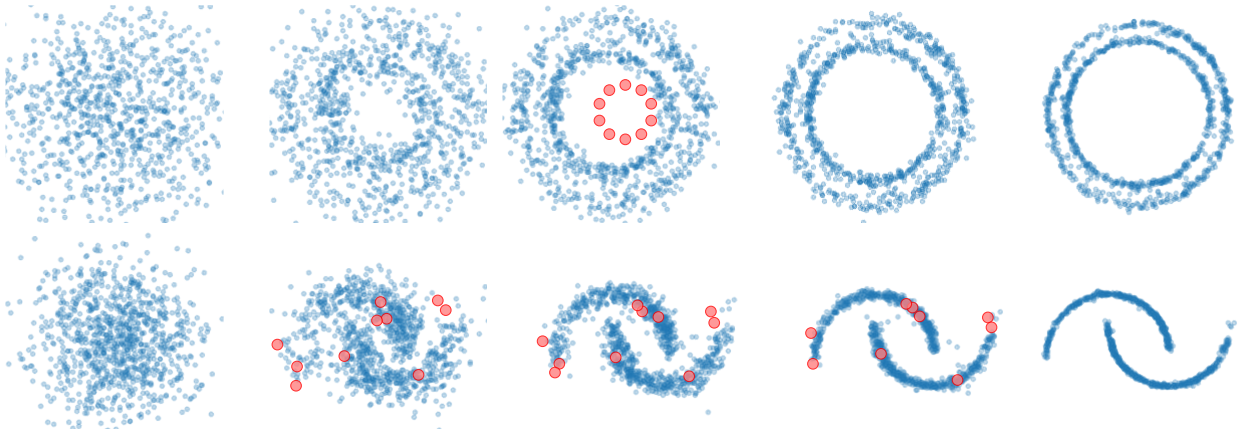
Two circles The observational data consists of 10 points which lie evenly distributed on a circle, observed at $t = 0.5$ with an exponential observational noise schedule $\kappa(l) = 0.5 \cdot 1.25^{l-1}$. The ISB was run for 6 epochs and initialized with a drift from the pre-trained Schrödinger bridge model from the unconstrained problem.

S-shape The observational data consists of 6 points, with pairs being observed at times $t \in [0.4, 0.5, 0.6]$. We used a bilinear observational noise schedule with a linear decay for the first half of the iterations from $\kappa(0)^2 = 4$ to $\kappa(L/2)^2 = 1$ and a linear ascend for the second half of the iterations from $\kappa(L/2)^2 = 1$ to $\kappa(L)^2 = 4$. The ISB ran for 6 epochs, with a zero drift initialization.

B.2 The Beneš SDE

In the Beneš SDE experiment, we obtain the sparse observational data from sampled Beneš SDE trajectories while the terminal state is a shifted and scaled $(3 + 5x_T)$ version of a Beneš marginal density. As the Beneš trajectories were first generated by simulating the SDE until $t = 6$ and then in reverse from $t = 6$ to $t = 0$, we set $T = 11.97$. We apply the analytical expression for the Beneš marginal density for computing $\log p_t(\mathbf{x})$,

$$p_t(\mathbf{x}) = \frac{1}{\sqrt{2\pi t}} \frac{\cosh(\mathbf{x})}{\cosh(\mathbf{x}_0)} \exp\left(-\frac{1}{2}t\right) \exp\left(-\frac{1}{2t}(\mathbf{x} - \mathbf{x}_0)^2\right). \tag{42}$$



(a) $t = 0.00$ (b) $t = 0.25$ (c) $t = 0.50$ (d) $t = 0.75$ (e) $t = 0.99$

Figure 6: The unconditioned Schrödinger bridges for the scikit-learn 2D experiment, corresponding to the Fig. 3. The observations (red markers) were not used in any way during training but are included in the figure for reference, to show that the unconstrained dynamics greatly differ from the learned ISB model dynamics.

See the Beneš SDE trajectories in Fig. 8a. As expected, the transport model with no observations performs well in the generative task, but its trajectories cover also some low-likelihood space around $t = 6$ (in the middle part in Fig. 8b). The observations for the ISB model were sampled from the generated trajectories, 10 observations at 10 random time-instances (see Fig. 8c)

Both the unconstrained Schrödinger bridge model and the ISB model were run for 3 iterations, using a learning rate of 0.001 for the neural networks. Likely due to the fact that the problem was only one-dimensional, the convergence of the Schrödinger bridge to a process which matches the desired terminal state was fast, and we chose not to run the model for a higher number of ISB iterations, see Fig. 7 for a comparison of the trained model marginal densities and the true terminal distribution π_T. We set the observation noise schedule to the constant 0.7, and at each iteration of the ISB or the unconstrained Schrödinger bridge the drift neural networks were trained for 5000 iterations each with the batch size 256, and the trajectories were refreshed every 500 iterations with a cache size of 1000 particles. The number of nearest neighbours to compare to was $H = 10$.

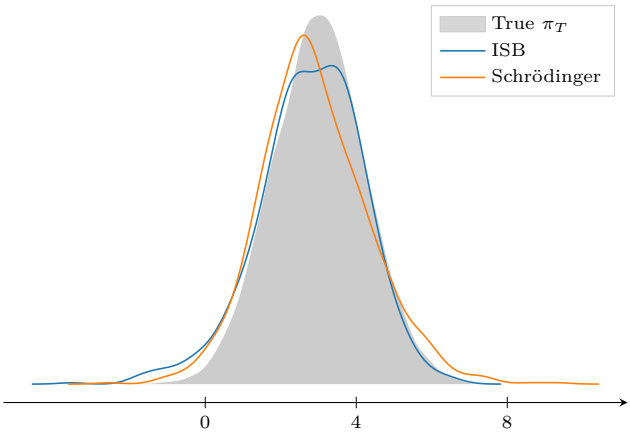
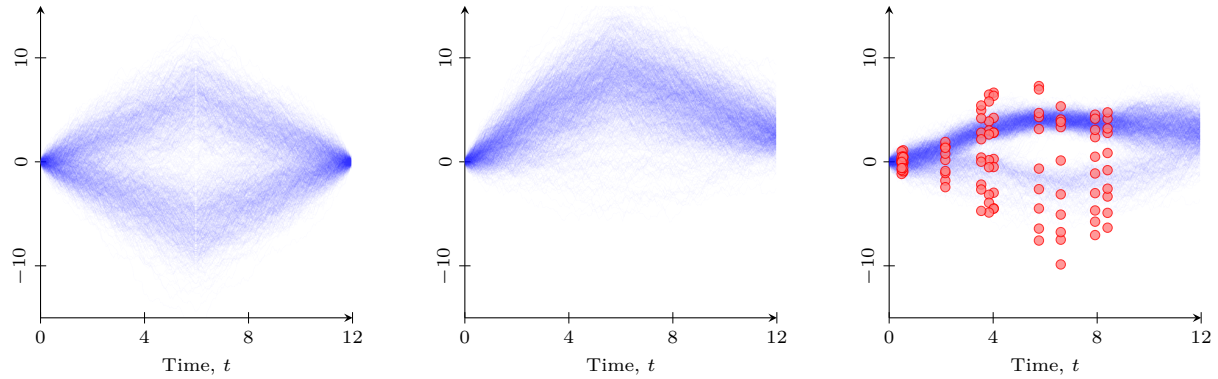


Figure 7: A kernel density estimate of the Beneš SDE terminal state. We compare π_T to the Schrödinger bridge and ISB terminal states. Both unconstrained Schrödinger bridge and ISB terminal states succeed in representing π_T well, with the Schrödinger bridge terminal state more closely matching π_T near its mean.



(a) Trajectories of the Beneš SDE (b) SBP trajectories (c) ISB trajectories (ours)

Figure 8: Comparison of the solution for the SBP (with Beneš SDE reference drift) and the ISB (with zero initial drift) on the Beneš SDE under sparse observations (●). The target distribution π_T is slightly shifted and scaled from the Beneš SDE. Even if the SBP has the true model as reference drift, its trajectories degenerate into a unimodal distribution, while the ISB manages to cover both modes even if only sparse observations are available.

B.3 The Bird Migration Data Set

The ISB model learned bird migration trajectories which transport the particles from the Northern Europe summer habitats to the southern winter habitats, see Fig. 11 for a comparison of a Schrödinger bridge and ISB. Since the problem lies on a sphere, Schrödinger bridge methods adjusted for learning on Riemannian manifolds could have been applied here. For simplicity, we mapped the problem to a two-dimensional plane using a Mercator projection and solved the problem on a $[0, 5] \times [0, 5]$ square. The SDE had the discretization $t \in [0, 0.99]$, $\Delta_k = 0.01$ and a constant process noise $g(t)^2 = 0.05$. The model was trained for 12 iterations, and initialized with a zero drift, while the observational data was chosen by the authors to promote learning trajectories clearly different from the unconstrained transport trajectories. The observation noise schedule was piecewise linear (starting at 2, going to 0.1 at iteration 6, then rising linearly to reach 2 at iteration 12). At each ISB iteration, the neural networks were trained for 5000 iterations each, and the trajectories were refreshed every 1000 iterations. We used a batch size of 256 and a learning rate of 0.001.

B.4 The MNIST Generation Task

Applying state-space model approaches such as particle filtering and smoothing to generative diffusion models directly in the observation space (that is, not in a lower-dimensional latent space) has to our knowledge not been explored before. Some experimental design choices had a great impact on the training objectives sensibility, as the observational data is completely artificial and its timing during the process modifies the filtering distribution significantly. As the MNIST conditional generative model was trained to display the scalability of our method beyond low-dimensional toy examples, we did not further explore optimizing the hyperparameters or the observation model. To avoid the background noise in MNIST images in the middle of the generative process impacting the particle filtering weights excessively, the observation model is a Gaussian with masked inputs equal to zero in pixels where the observation image is black, see Fig. 9 for sampled trajectories. The figure shows the progression of seven samples, where the lower half of the eight resemble the observation target.

The SDE was run for time $t \in [0, 0.5]$, with the digit eight observed at $t = 0.38$. The ISB method was applied for 10 iterations, with a discretization $t \in [0, 0.495]$, $\Delta_k = 0.005$, and the process noise $g(t)^2$ followed a linear schedule from 0.0001 to 1. At each iteration of the method, the forward and backward drift neural networks were trained for 5000 iterations with a batch size of 256, and the trajectory cache regenerated every 1000 iterations. The observational data consisted of a single sample of a lower half of the digit eight, observed at time $t = 0.38$. The observation noise schedule was a constant $\kappa(l) = 0.3$.

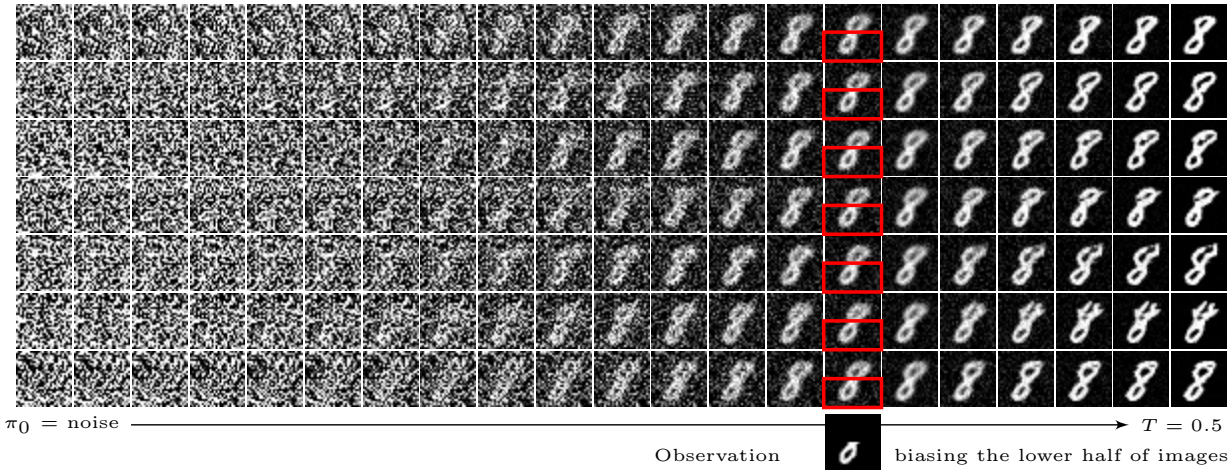


Figure 9: Model trajectories for MNIST digit '8' conditioned on a lower-loop of a single '8' at $t = 0.38$ to bias the lower half of the digits to look alike, with the effect still visible at terminal time T.

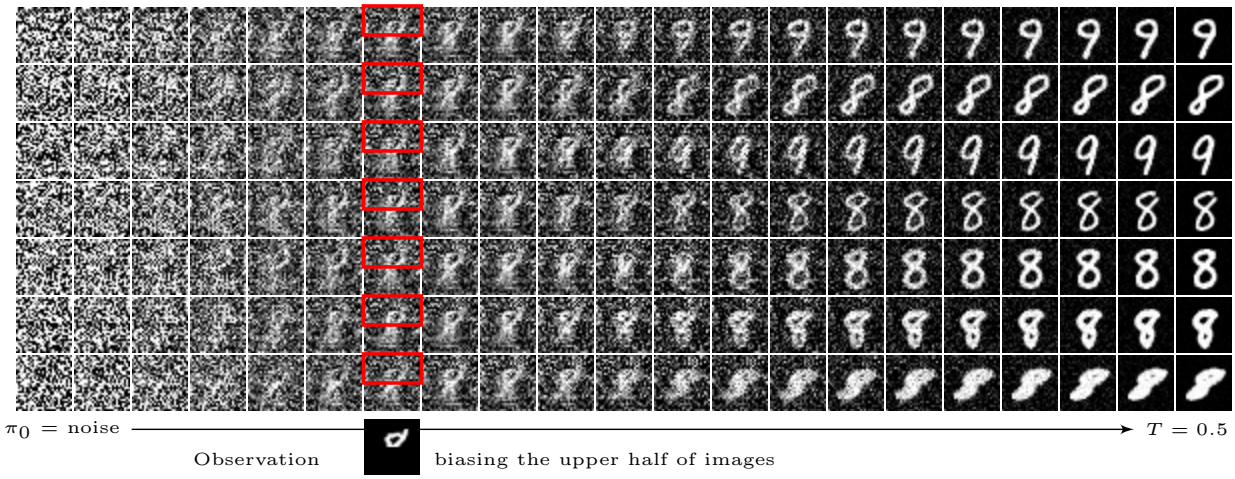


Figure 10: Model trajectories for MNIST digits '8' and '9' conditioned on a upper-loop of a single '8' at $t = 0.38$ to bias the upper half of the digits to look alike, with the effect still visible at terminal time T, while observing both modes at the end of the trajectory.

B.5 MNIST for Multi-modal Data Generation

In addition to the MNIST experiment as explained in App. B.4, we evaluated the performance of our model on a multi-modal generation task. The reference model was a Schrödinger bridge trained via IPFP from normal distribution noise to MNIST digits eight and nine, and the observation was a single upper loop of a figure eight. Our goal was to generate trajectories that match the observation but still generate both digits eight and nine by the end of the trajectory. Based on the results in Fig. 10, ISB succeeds in the task. Most hyperparameters were kept the same as explained in App. B.4, but the cache size if IPFP training was increased to 5000 and the number of trajectories in the particle filtering step of ISB was increased to 1000, to encourage a sufficiently versatile sample of both modes.

B.6 Single-Cell Data Set

We directly use the preprocessed data from the TrajectoryNet (Tong et al., 2020) repository. A major difference between our implementation and Vargas et al. (2021) is the reference drift. We set the reference drift to zero, which means that we utilize the intermediate data only as observations in the state-space model. On the contrary, Vargas et al. (2021) fits a mixture model of 15 Gaussians on the combined data set (across

all measurement times) and sets the reference drift to the gradient of the log-likelihood of the mixture model. Effectively, such a reference drift aids in keeping the SDE trajectories within the support of the combined data set. We remark that if the intermediate observed marginals had clearly disjoint support, combining all the data would cause the mixture model to have 'gaps' and could cause an unstable reference model drift. Thus we consider our approach of setting the reference drift to zero as more generally applicable.

As in Vargas et al. (2021), we set the process noise to $g(t) = 1$ and model the SDE between time $t \in [0, 4]$. The learning rate is set to 0.001 with a batch size of 256 and the number of neural network training iterations equal to 5000. We apply the ISB for 6 iterations. We perform filtering using 1000 points from the intermediate data sets, but compute the Earth mover's distance by comparing it to all available data. As the observational data at $T = 1, 2, 3$ consists of a high number of data points, the parameters H (number of nearest neighbours) and σ (observation noise) need to be carefully set. We set $H = 10$ to only include the close neighbourhood of each particle and set the observation noise schedule as constant 0.7.

C Computational Considerations

In Sec. 3.2, we raised a number of important computational considerations for the constrained transport problem. Below we discuss them in detail, analyzing the limit $L \to \infty$ from the perspective of setting the observation noise schedule in App. C.1, and presenting ablation results on modifying the initial drift in the bird migration experiment in App. C.2. Finally, we study the impact of the observation noise schedule on effective sample size in the filtering step in App. C.3.

C.1 Discussion on Observation Noise

We briefly mentioned in Sec. 3.2 that when letting $L \to \infty$, the choice of observation noise should be carefully planned in order for the ISB procedure to have a stationary point. Here we explain why an unbounded observation noise schedule $\kappa(l)$ implies convergence to the IPF method for uncontrolled Schrödinger bridges (De Bortoli et al., 2021), when using a nearest neighbour bootstrap filter as the proposal density.

Proposition 3. *Let $\Omega \in \mathbb{R}^d$ be a bounded domain where both the observations and SDE trajectories lie, and let the particle filtering weights $\{w^i_{l,t_k}\}_{i=1}^N$ at ISB iteration l be*

$$\log w^i_{l,t_k} = -\frac{1}{2\kappa(l)^2} \sum_{\mathbf{y}_j \in \mathcal{D}^H_{t_k}} \|\mathbf{x}^i_{t_k} - \mathbf{y}_j\|^2. \tag{43}$$

If the schedule $\kappa(l)$ is unbounded with respect to l, then for any δ there exists l' such that for the normalized weights it holds

$$|\hat{w}^i_{l',t_k} - \frac{1}{N}| \leq \delta. \tag{44}$$

Proof sketch. Since $\kappa(l)$ is unbounded, for any $S > 0$ $\exists$ l' such that $\kappa(l') \geq S$. We choose the value of S so that the following derivation yields Eq. (44).

Let $S = \sqrt{0.5R^{-1}|\mathcal{D}^H_{t_k}| \operatorname{diam}(\Omega)^2}$, and apply the property that $\|\mathbf{x}^i_{t_k} - \mathbf{y}_j\|^2 \leq \operatorname{diam}(\Omega)^2$ to Eq. (43),

$$\begin{aligned} \log w^i_{l',t_k} &\geq -\frac{1}{2S^2} \sum_{\mathbf{y}_j \in \mathcal{D}^H_{t_k}} \|\mathbf{x}^i_{t_k} - \mathbf{y}_j\|^2 \\ &\geq -\frac{\sum_{\mathbf{y}_j \in \mathcal{D}^H_{t_k}} \|\mathbf{x}^i_{t_k} - \mathbf{y}_j\|^2}{R^{-1}|\mathcal{D}^H_{t_k}| \operatorname{diam}(\Omega)^2} \geq -\frac{\sum_{\mathbf{y}_j \in \mathcal{D}^H_{t_k}} \operatorname{diam}(\Omega)^2}{R^{-1}|\mathcal{D}^H_{t_k}| \operatorname{diam}(\Omega)^2} \geq -R. \end{aligned} \tag{45}$$

The bound above is for the unnormalized weights, and the normalized log-weights are defined as

$$\log \hat{w}^i_{l',t_k} = \log w^i_{l',t_k} - \log \left(\sum_{j=1}^N \exp(\log w^j_{l',t_k}) \right), \tag{46}$$

where for the normalizing constant it holds that

$$\log\left(\sum_{j=1}^{N}\exp(\log w^j_{l',t_k})\right) \leq \log\left(\sum_{j=1}^{N}1\right) = \log(N), \tag{47}$$

since w^j_{l',t_k} is the value of a probability density and thus always $w^j_{l',t_k} \leq 1$. Combining Eq. (46), Eq. (45) and Eq. (47), it follows that

$$\log \hat{w}^i_{l',t_k} - (-\log(N) \geq -R, \tag{48}$$

where taking exponentials on both sides gives

$$\hat{w}^i_{l',t_k} - \frac{1}{N} \geq -(1-\exp(-R))\frac{1}{N}. \tag{49}$$

Since the weights are normalized, even the largest particle weight $\hat{w}^j_{l',t_k}$ can differ from $\frac{1}{N}$ as much as every smaller weight in total lies under $\frac{1}{N}$,

$$\hat{w}^j_{l',t_k} \leq \frac{1}{N} + (N-1)\left((1-\exp(-R))\frac{1}{N}\right), \tag{50}$$

implying that for any weight $\hat{w}^j_{l',t_k}$, it holds that

$$|\hat{w}^j_{l',t_k} - \frac{1}{N}| \leq (N-1)\left((1-\exp(-R))\frac{1}{N}\right) \leq 1-\exp(-R), \tag{51}$$

and selecting $R = -\log(1-\delta)$ is sufficient for $\delta < 1$. □

Effectively, the above derivation implies that for an unbounded observation noise schedule $\kappa(l)$, the particle weights will converge to uniform weights. Since performing differentiable resampling on uniform weights implies that $\mathbf{T}_{(\varepsilon)} = \mathbf{I}_N$, the ISB method trajectory generation step and the objective in training the backward drift converge to those of the IPF method for solving unconstrained Schrödinger bridges. Intuitively, this means that at the limit $L \to \infty$, our method will focus on reversing the trajectories and matching the terminal distribution while not further utilizing information from the observations.

C.2 Ablation on Initial Drift

We conducted an ablation study on drift initialization for the bird migration problem. As the distributions π_0 and π_T (as pictured in Fig. 11) are complex, we consider the problem setting to be interesting for setting f_0 as the unconstrained transport problem drift. To this end, we trained a Schrödinger bridge model for 10 epochs, and trained an ISB model with the same hyperparameter selections as explained in App. B.3, using the Schrödinger bridge as the initialization. Compare the two bottom rows of Fig. 11 to see a selection of marginal densities of the two processes. Based on a visual analysis of the densities, it seems that the zero drift and pre-trained diffusion model initializations produce similar results around the observations, although the Schrödinger bridge initialization gave slightly sharper results at the terminal time.

C.3 Effective Sample Size

We studied the Effective Sample Size (ESS) of the particles generated during the filtering steps 1 and 3. Ideally, our method would steer the particle trajectories near the observations quickly over the iterations, to allow for efficient training. Furthermore, as we additionally constrain the system to match the initial and terminal distributions, it is desired that the ESS will eventually rise to a high number, indicating that the particle filtering step no longer greatly adjusts the trajectories. In order to assess these properties, we computed the ESS in the scikit-learn TWO CIRCLES experiment, see App. B.1 for other experimental details. In Table 3, we have reported three scenarios: *(i)* an increasing noise schedule without learning of the drift function, *(ii)* an ISB model with constant observation noise, *(iii)* the proposed ISB model with the increasing noise schedule. Note that the first scenario only serves as a control to better isolate the impact of increasing noise in ISB. Table 3 indicates that our approach succeeds in directing the particle trajectories and consistently obtains higher ESS even when comparing the constant noise level ISB to an increasing noise level but no learning.

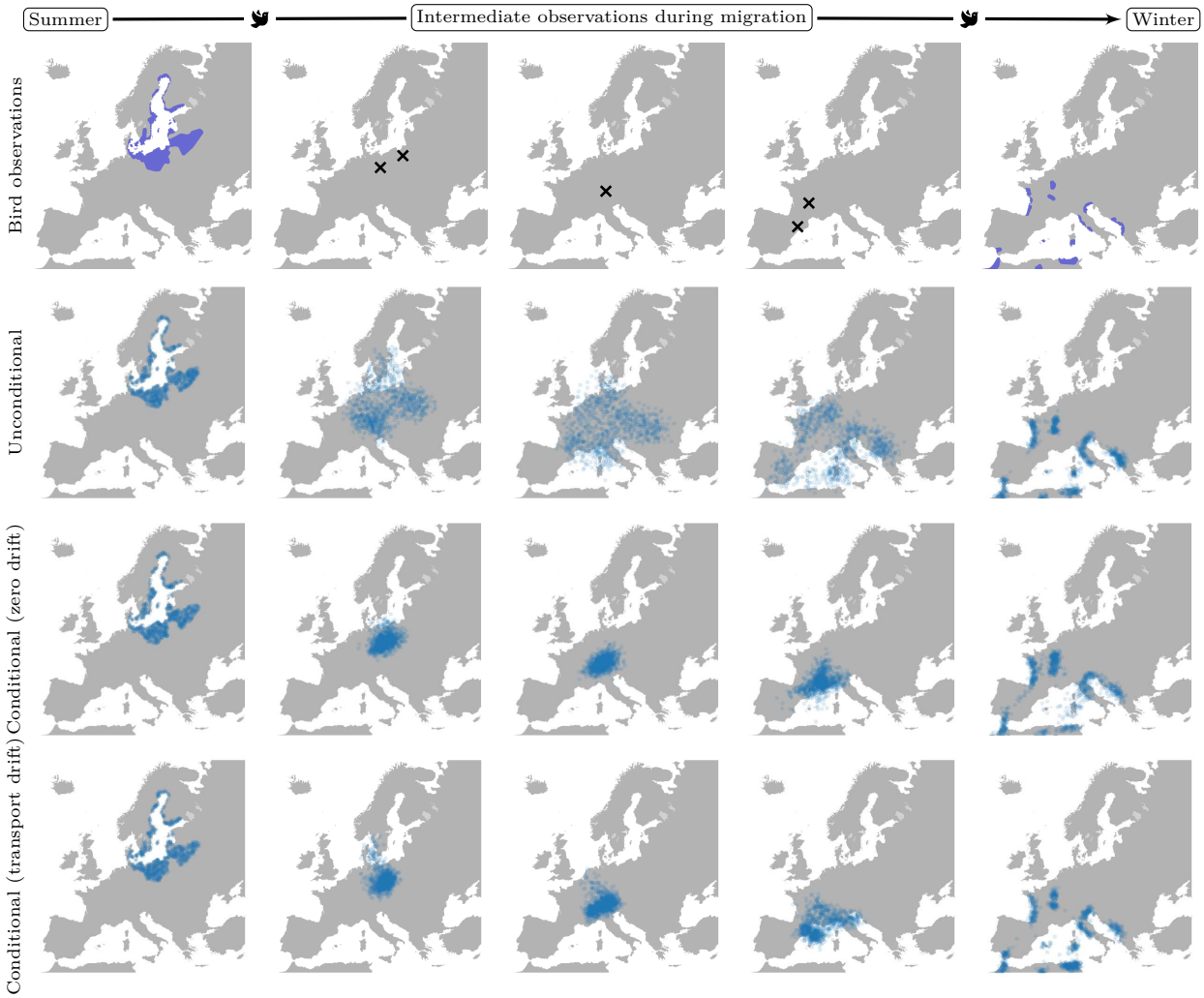


Figure 11: Top row: The first map image on the left describes the initial position of the birds, and the final one on the right depicts their position after migration. The observational data in the middle are bird observations during migration, at given timestamps. Second row: Marginal densities of a Schrödinger bridge model from the initial to terminal distribution, without using the observations. Third row: Marginal densities of our model, using both initial and terminal distributions and observational data and a zero drift initialization. Bottom row: Same as the third row, but with the second-row dynamics as initialization.

Table 3: Average and standard deviation of effective sample size (ESS) results over five seeds for the scikit-learn TWO CIRCLES experiment, developing over six ISB iterations with 1000 particles. We observe a rapid increase in ESS when the trajectories are steered towards the observations in the optimization step of the ISB compared to only increasing the observing noise, and see a diminishing impact of the filtering step as ESS nears 1000.

	ESS / ITERATION					
NOISE SCHEDULE	$l = 1$	$l = 2$	$l = 3$	$l = 4$	$l = 5$	$l = 6$
No learning and noise schedule $0.5 \times 1.25^{l-1}$	6 ± 2	9 ± 2	14 ± 3	27 ± 5	57 ± 8	121 ± 11
ISB with a constant noise schedule 0.5	6 ± 2	66 ± 21	165 ± 9	300 ± 24	456 ± 37	564 ± 46
ISB with a noise schedule $0.5 \times 1.25^{l-1}$	6 ± 2	92 ± 21	250 ± 14	442 ± 39	593 ± 64	700 ± 39

D Differentiable Resampling

In the ISB model steps 1 and 3 presented in Sec. 3.1, we applied differentiable resampling (see Corenflos et al., 2021). Resampling itself is a basic block of particle filtering. A differentiable resampling step transports the particles and weights $(\tilde{\mathbf{x}}^i_{t_k}, w^i_{t_k})$ to a uniform distribution over a set of particles through applying the *differentiable* ensemble transport map $\mathbf{T}_{(\varepsilon)}$, that is

$$(\tilde{\mathbf{x}}^i_{t_k}, w^i_{t_k}) \to (\tilde{\mathbf{X}}^\top_{t_k} \mathbf{T}_{(\varepsilon),i}, {}^1\!/_N) = (\mathbf{x}^i_{t_k}, {}^1\!/_N), \tag{52}$$

where $\tilde{\mathbf{X}}_{t_k} \in \mathbb{R}^{N\times d}$ denotes the stacked particles $\{\tilde{\mathbf{x}}^i_{t_k}\}_{i=1}^N$ at time t_k before resampling and $\mathbf{x}^i_{t_k}$ denotes the particles post resampling. Here we give the definition of the map $\mathbf{T}_{(\varepsilon)}$ and review the regularized optimal transport problem which has to be solved to compute it. We partly follow the presentation in Sections 2 and 3 of Corenflos et al. (2021), but directly apply the notation we use for particles and weights and focus on explaining the transport problem rather than the algorithm used to solve it.

The standard particle filtering resampling step consists of sampling N particles from the categorical distribution defined by the weights $\{w^i_{t_k}\}_{i=1}^N$, resulting in the particles with large weights being most likely to be repeated multiple times. A result from Reich (2013) gives the property that the random resampling step can be approximated by a deterministic ensemble transform $\mathbf{T}$. In heuristic terms, the ensemble transform map will be selected so that the particles $\{\mathbf{x}^i_{t_k}\}_{i=1}^N$ will be transported with minimal cost, while allowing all the weights to be uniform.

Let μ and ν be atomic measures, $\mu = \sum_{i=1}^N w^i_{t_k}\delta_{\tilde{\mathbf{x}}^i_{t_k}}$ and $\nu = \sum_{i=1}^N N^{-1}\delta_{\tilde{\mathbf{x}}^i_{t_k}}$, where δ_x is the Dirac delta at x. Then μ is the particle filtering distribution before resampling. Define the elements of a cost matrix $\mathbf{C} \in \mathbb{R}^{N\times N}$ as $C_{i,j} = \|\tilde{\mathbf{x}}^i_{t_k} - \tilde{\mathbf{x}}^j_{t_k}\|^2$, and the 2-Wasserstein distance between two atomic measures as

$$\mathcal{W}_2^2(\mu,\nu) = \min_{P\in S(\mu,\nu)} \sum_{i=1}^N \sum_{j=1}^N C_{i,j}P_{i,j}. \tag{53}$$

Above the optimal matrix $\mathbf{P}$ is to be found within $S(\mu,\nu)$, which is a space consisting of mixtures of N particles to N particles such that the marginals coincide with the weights of μ and ν, formally

$$S(\mu,\nu) = \left\{ \mathbf{P} \in [0,1]^{N\times N} \mid \sum_{i=1}^N P_{i,j} = w^i_{t_k}, \sum_{j=1}^N P_{i,j} = \frac{1}{N} \right\}. \tag{54}$$

The entropy-regularized Wasserstein distance with regularization parameter ε is then

$$\mathcal{W}_{2,\varepsilon}^2 = \min_{\mathbf{P}\in S(\mu,\nu)} \sum_{i=1}^N \sum_{j=1}^N P_{i,j} \left(C_{i,j} + \varepsilon \log \frac{P_{i,j}}{w^i_{t_k} \cdot \frac{1}{N}} \right). \tag{55}$$

The unique minimizing transport map of the above Wasserstein distance is denoted by $\mathbf{P}_\varepsilon^{\mathrm{OPT}}$, and the ensemble transport map is then set as $\mathbf{T}_{(\varepsilon)} = N\mathbf{P}_\varepsilon^{\mathrm{OPT}}$. This means that we can find the matrix $\mathbf{T}_{(\varepsilon)}$ via minimizing the regularized Wasserstein distance, which is done by applying the iterative Sinkhorn algorithm for entropy-regularized optimal transport (Cuturi, 2013).