Diffusion Schrödinger Bridge with Applications to Score-Based Generative Modeling

Anonymous Author(s)
Affiliation
Address
email

Abstract

Progressively applying Gaussian noise transforms complex data distributions to approximately Gaussian. Reversing this dynamic defines a generative model. When the forward noising process is given by a Stochastic Differential Equation (SDE), Song et al. (2021) demonstrate how the time inhomogeneous drift of the associated reverse-time SDE may be estimated using score-matching. A limitation of this approach is that the forward-time SDE must be run for a sufficiently long time for the final distribution to be approximately Gaussian. In contrast, solving the Schrödinger Bridge (SB) problem, *i.e.* an entropy-regularized optimal transport problem on path spaces, yields diffusions which generate samples from the data distribution in finite time. We present Diffusion SB (DSB), an original approximation of the Iterative Proportional Fitting (IPF) procedure to solve the SB problem, and provide theoretical analysis along with generative modeling experiments. The first DSB iteration recovers the methodology proposed by Song et al. (2021), with the flexibility of using shorter time intervals, as subsequent DSB iterations reduce the discrepancy between the final-time marginal of the forward (resp. backward) SDE with respect to the prior (resp. data) distribution. Beyond generative modeling, DSB offers a widely applicable computational optimal transport tool as the continuous state-space analogue of the popular Sinkhorn algorithm (Cuturi, 2013).

1 Introduction

Score-Based Generative Modeling (SGM) is a recently developed approach to probabilistic generative modeling that exhibits state-of-the-art performance on several audio and image synthesis tasks; see *e.g.* Song and Ermon (2019); Cai et al. (2020); Chen et al. (2021a); Kong et al. (2021); Gao et al. (2020); Jolicoeur-Martineau et al. (2021); Ho et al. (2020); Song and Ermon (2020); Song et al. (2020, 2021); Niu et al. (2020); Durkan and Song (2021); Hoogeboom et al. (2021); Saharia et al. (2021); Luhman and Luhman (2021, 2020); Nichol and Dhariwal (2021); Popov et al. (2021); Dhariwal and Nichol (2021). Existing SGMs generally consist of two parts. Firstly, noise is incrementally added to the data in order to obtain a perturbed data distribution approximating an easy-to-sample *prior* density *e.g.* Gaussian. Secondly, a neural network is used to learn the reverse-time denoising dynamics, which when initialized at this prior distribution, defines a generative model (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song and Ermon, 2019; Song et al., 2021). Song et al. (2021) have shown that one could fruitfully think of the noising process as a Stochastic Differential Equation (SDE) progressively perturbing the initial data distribution into an approximately Gaussian one. The corresponding reverse-time SDE is an inhomogeneous diffusion whose drift depends on the logarithmic gradients of the perturbed data distributions, *i.e.* the scores. In practice, these scores are approximated using neural networks and score-matching techniques (Hyvärinen and Dayan, 2005; Vincent, 2011) while numerical SDE integrators are used for the sampling procedure.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

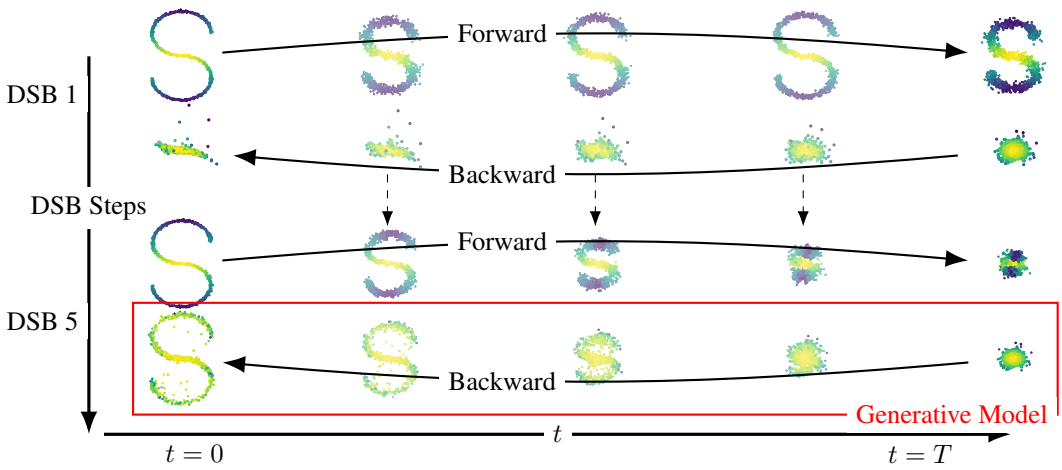

Figure 1: The reference forward diffusion initialized from the 2-dimensional data distribution fails to converge to the Gaussian prior in $T = 0.2$ diffusion-time ($N = 20$ discrete time steps), and the reverse diffusion initialized from the Gaussian prior does not converge to the data distribution. However, convergence does occur after 5 DSB iterations.

Although SGM provides state-of-the-art results (Dhariwal and Nichol, 2021), sample generation is computationally expensive. In order to learn the reverse-time SDE from the prior, *i.e.* the generative model, the forward noising SDE must be run for sufficiently long to converge to the prior and the step size must be sufficiently small for a good SDE approximation. By reformulating generative modeling as a Schrödinger bridge (SB) problem we alleviate this issue and propose a novel algorithm to solve SB problems. Our detailed contributions are as follows.

Generative modeling as a Schrödinger bridge problem. The SB problem is a famous entropy-regularized Optimal Transport (OT) problem introduced by Schrödinger (1932); see *e.g.* (Léonard, 2014b; Chen et al., 2021b) for reviews. Given a reference diffusion with finite time horizon T, a data distribution and a prior distribution, solving the SB amounts to finding the closest diffusion to the reference (in terms of Kullback–Leibler divergence on path spaces) which admits the data distribution as marginal at time $t = 0$ and the prior at time $t = T$. The reverse-time diffusion solving this Schrödinger bridge problem provides a new SGM algorithm which enables approximate sample generation from the data distribution using shorter time intervals compared to the original SGM methods. This approach differs from the entropy-regularized OT formulation proposed in (Genevay et al., 2018) which deals with discrete distributions and relies on a static formulation of SB, as opposed to our dynamical approach for continuous distributions which operates on path spaces.

Solving the Schrödinger bridge problem using score-based diffusions. The Schrödinger bridge problem can be solved using Iterative Proportional Fitting (IPF) (Fortet, 1940; Kullback, 1968; Chen et al., 2021b). We propose Diffusion SB (DSB), a novel implementation of IPF using score-based diffusion techniques and an original mean-matching loss. DSB does not require discretizing the state-space (Chen et al., 2016; Reich, 2019), approximating potential functions using regression (Bernton et al., 2019; Dessein et al., 2017; Pavon et al., 2021), nor performing kernel density estimation (Pavon et al., 2021). The first DSB iteration recovers the method proposed by Song et al. (2021), with the flexibility of using shorter time intervals, as additional DSB iterations reduce the discrepancy between the final-time marginal of the forward (resp. backward) SDE w.r.t. the prior (resp. data) distribution; see Figure 1 for an illustration.

Theoretical analysis. We provide quantitative convergence results for the methodology of Song et al. (2021), which elucidates the need for long diffusion times in existing SGM methods. Additionally, we derive novel quantitative convergence results for IPF in continuous state-space which do not rely on classical compactness assumptions (Chen et al., 2016; Ruschendorf et al., 1995) and improve on the recent results of Léger (2020). Finally, we show that our methodology may be viewed as the time discretization of a dynamic version of IPF on path spaces.

Experiments. We validate our methodology by generating image datasets such as MNIST and CelebA. In particular, we show that using multiple steps of DSB always improve the generative model. We also show how DSM can be used to interpolate between two data distributions.

Notation. In the continuous-time setting, we set $\mathcal{C} = \mathrm{C}([0,T], \mathbb{R}^d)$ the space of continuous functions from $[0,T]$ to $\mathbb{R}^d$ and $\mathcal{B}(\mathcal{C})$ the Borel sets on $\mathcal{C}$. For any measurable space $(\mathsf{E}, \mathcal{E})$, we denote by $\mathscr{P}(\mathsf{E})$ the space of probability measures on $(\mathsf{E}, \mathcal{E})$. For any $\ell \in \mathbb{N}$, let $\mathscr{P}_\ell = \mathscr{P}((\mathbb{R}^d)^\ell)$. When it is defined, we denote $\mathrm{H}(p) = -\int_{\mathbb{R}^d} p(x) \log p(x) \mathrm{d}x$ as the entropy of p and $\mathrm{KL}(p|q)$ as the Kullback–Leibler divergence between p and q. When there is no ambiguity we use the same notation for the distributions and their densities. All proofs are postponed to the supplementary.

2 Denoising Diffusion, Score-Matching and Reverse-Time SDEs

2.1 Discrete-Time: Markov Chains and Time Reversal

Consider a data distribution with positive density p_{data}[1], a positive prior density p_{prior} w.r.t. Lebesgue measure both with support on $\mathbb{R}^d$ and a Markov chain with initial density $p_0 = p_{\text{data}}$ on $\mathbb{R}^d$ evolving according to positive transition densities $p_{k+1|k}$ for $k \in \{0, \dots, N-1\}$. Hence for any $x_{0:N} = \{x_k\}_{k=0}^N \in \mathcal{X} = (\mathbb{R}^d)^{N+1}$, the joint density may be expressed

$$p(x_{0:N}) = p_0(x_0) \prod_{k=0}^{N-1} p_{k+1|k}(x_{k+1}|x_k). \tag{1}$$

This joint density also admits the backward decomposition

$$p(x_{0:N}) = p_N(x_N) \prod_{k=0}^{N-1} p_{k|k+1}(x_k|x_{k+1}), \text{with } p_{k|k+1}(x_k|x_{k+1}) = \frac{p_k(x_k) p_{k+1|k}(x_{k+1}|x_k)}{p_{k+1}(x_{k+1})}, \tag{2}$$

where $p_k(x_k) = \int p_{k|k-1}(x_k|x_{k-1}) p_{k-1}(x_{k-1}) \mathrm{d}x_{k-1}$ is the marginal density at step $k \geq 1$. For the purpose of generative modeling, we will choose transition densities such that $p_N(x_N) \approx p_{\text{prior}}(x_N)$ for large N, where p_{prior} is an easy-to-sample *prior* density. One may sample approximately from p_{data} using ancestral sampling with the reverse-time decomposition (2), *i.e.* first sample $X_N \sim p_{\text{prior}}$ followed by $X_k \sim p_{k|k+1}(\cdot|X_{k+1})$ for $k \in \{N-1, \dots, 0\}$. This idea is at the core of all recent SGM. The reverse-time transitions in (2) cannot be simulated exactly but may be approximated if we consider a forward transition density of the form

$$p_{k+1|k}(x_{k+1}|x_k) = \mathcal{N}(x_{k+1}; x_k + \gamma_{k+1} f(x_k), 2\gamma_{k+1}\mathbf{I}), \tag{3}$$

with drift $f: \mathbb{R}^d \to \mathbb{R}^d$ and stepsize $\gamma_{k+1} > 0$. We first make the following approximation from (2)

$$\begin{aligned} p_{k|k+1}(x_k|x_{k+1}) &= p_{k+1|k}(x_{k+1}|x_k) \exp[\log p_k(x_k) - \log p_{k+1}(x_{k+1})] \\ &\approx \mathcal{N}(x_k; x_{k+1} - \gamma_{k+1} f(x_{k+1}) + 2\gamma_{k+1} \nabla \log p_{k+1}(x_{k+1}), 2\gamma_{k+1}\mathbf{I}), \end{aligned} \tag{4}$$

using that $p_k \approx p_{k+1}$, a Taylor expansion of $\log p_{k+1}$ at x_{k+1} and $f(x_k) \approx f(x_{k+1})$. In practice, the approximation holds if $\|x_{k+1} - x_k\|$ is small which is ensured by choosing γ_{k+1} small enough. Although $\nabla \log p_{k+1}$ is not available, one may obtain an approximation using denoising score-matching methods (Hyvärinen and Dayan, 2005; Vincent, 2011; Song et al., 2021). Assume that the conditional density $p_{k+1|0}(x_{k+1}|x_0)$ is available analytically as in (Ho et al., 2020; Song et al., 2021). We have $p_{k+1}(x_{k+1}) = \int p_0(x_0) p_{k+1|0}(x_{k+1}|x_0) \mathrm{d}x_0$ and elementary calculations show that $\nabla \log p_{k+1}(x_{k+1}) = \mathbb{E}_{p_{0|k+1}}[\nabla_{x_{k+1}} \log p_{k+1|0}(x_{k+1}|X_0)]$. We can therefore formulate score estimation as a regression problem and use a flexible class of functions, *e.g.* neural networks, to parametrize an approximation $s_{\theta^\star}(k, x_k) \approx \nabla \log p_k(x_k)$ such that

$$\theta^\star = \arg\min_\theta \sum_{k=1}^N \mathbb{E}_{p_{0,k}}[||s_\theta(k, X_k) - \nabla_{x_k} \log p_{k|0}(X_k|X_0)||^2],$$

where $p_{0,k}(x_0, x_k) = p_0(x_0) p_{k|0}(x_k|x_0)$ is the joint density at steps 0 and k. If $p_{k|0}$ is not available, we use $\theta^\star = \arg\min_\theta \sum_{k=1}^N \mathbb{E}_{p_{k-1,k}}[||s_\theta(k, X_k) - \nabla_{x_k} \log p_{k|k-1}(X_k|X_{k-1})||^2]$. In summary, SGM involves first estimating the score function $s_{\theta^\star}$ from noisy data, and then sampling X_0 using $X_N \sim p_{\text{prior}}$ and the approximation (4), *i.e.*

$$X_k = X_{k+1} - \gamma_{k+1} f(X_{k+1}) + 2\gamma_{k+1} s_{\theta^\star}(k+1, X_{k+1}) + \sqrt{2\gamma_{k+1}} Z_{k+1}, \; Z_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \mathbf{I}). \tag{5}$$

The random variable X_0 is approximately $p_0 = p_{\text{data}}$ distributed if $p_N(x_N) \approx p_{\text{prior}}(x_N)$. In what follows, we let $\{Y_k\}_{k=0}^N = \{X_{N-k}\}_{k=0}^N$ and remark that $\{Y_k\}_{k=0}^N$ satisfies a forward recursion.

[1] In this presentation, we assume that all distributions admit a density w.r.t. the Lebesgue measure for simplicity. However, the algorithms presented here only require having access to samples from p_{data} and p_{prior}.

2.2 Continuous-Time: SDEs, Reverse-Time SDEs and Theoretical results

For appropriate transition densities, Song et al. (2021) showed that the forward and reverse-time Markov chains may be viewed as discretized diffusions. We derive the continuous-time limit of the procedure presented in Section 2.1 and establish convergence results. The Markov chain with kernel (3) corresponds to an Euler–Maruyama discretization of $(\mathbf{X}_t)_{t\in[0,T]}$, solving the following SDE

$$\mathrm{d}\mathbf{X}_t = f(\mathbf{X}_t)\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t\ , \quad \mathbf{X}_0 \sim p_0 = p_{\text{data}}, \tag{6}$$

where $(\mathbf{B}_t)_{t\in[0,T]}$ is a Brownian motion and $f:\ \mathbb{R}^d \to \mathbb{R}^d$ is regular enough so that (strong) solutions exist. Under conditions on f, it is well-known (see Haussmann and Pardoux (1986); Föllmer (1985); Cattiaux et al. (2021) for instance) that the reverse-time process $(\mathbf{Y}_t)_{t\in[0,T]} = (\mathbf{X}_{T-t})_{t\in[0,T]}$ satisfies

$$\mathrm{d}\mathbf{Y}_t = \{-f(\mathbf{Y}_t) + 2\nabla \log p_{T-t}(\mathbf{Y}_t)\}\,\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t, \tag{7}$$

with initialization $\mathbf{Y}_0 \sim p_T$, where p_t denotes the marginal density of $\mathbf{X}_t$. The reverse-time Markov chain $\{Y_k\}_{k=0}^N$ associated with (5) corresponds to an Euler–Maruyama discretization of (7), where the score functions $\nabla \log p_t(x)$ are approximated by $s_{\theta^\star}(t,x)$.

In what follows, we consider $f(x) = -\alpha x$ for $\alpha \geq 0$. This framework includes the one of Song and Ermon (2019) ($\alpha = 0$, $p_{\text{prior}}(x) = \mathcal{N}(x; 0, 2T)$) for which $(\mathbf{X}_t)_{t\in[0,T]}$ is simply a Brownian motion and Ho et al. (2020) ($\alpha > 0$, $p_{\text{prior}}(x) = \mathcal{N}(x; 0, 1/\alpha)$) for which it is an Ornstein–Uhlenbeck process, see Section S3.3 for more details. Contrary to Song et al. (2021) we consider time homogeneous diffusions. Both approaches approximate (5) using distinct discretizations but our setting leverages the ergodic properties of the Ornstein-Uhlenbeck process to establish Theorem 1.

Theorem 1. *Assume that there exists* $\mathtt{M} \geq 0$ *such that for any* $t \in [0,T]$ *and* $x \in \mathbb{R}^d$

$$\|s_{\theta^\star}(t,x) - \nabla \log p_t(x)\| \leq \mathtt{M}, \tag{8}$$

with $s_{\theta^\star} \in \mathrm{C}([0,T]\times\mathbb{R}^d, \mathbb{R}^d)$. *Assume that* $p_{\text{data}} \in \mathrm{C}^3(\mathbb{R}^d, (0,+\infty))$ *is bounded and that there exist* $d_1, A_1, A_2, A_3 \geq 0$, $\beta_1, \beta_2, \beta_3 \in \mathbb{N}$ *and* $\mathtt{m}_1 > 0$ *such that for any* $x \in \mathbb{R}^d$ *and* $i \in \{1,2,3\}$

$$\|\nabla^i \log p_{\text{data}}(x)\| \leq A_i(1 + \|x\|^{\beta_i}), \quad \langle \nabla \log p_{\text{data}}(x), x\rangle \leq -\mathtt{m}_1 \|x\|^2 + d_1 \|x\|\,,$$

with $\beta_1 = 1$. *Then for any* $\alpha \geq 0$, *there exist* $B_\alpha, C_\alpha, D_\alpha \geq 0$ *such that for any* $N \in \mathbb{N}$ *and* $\{\gamma_k\}_{k=1}^N$ *with* $\gamma_k > 0$ *for any* $k \in \{1,\dots,N\}$, *the following hold:*

(a) if $\alpha > 0$, *we have* $\|\mathcal{L}(X_0) - p_{\text{data}}\|_{\mathrm{TV}} \leq B_\alpha \exp[-\alpha^{1/2}T] + C_\alpha(\mathtt{M} + \bar{\gamma}^{1/2})\exp[D_\alpha T]$;

(b) if $\alpha = 0$, *we have* $\|\mathcal{L}(X_0) - p_{\text{data}}\|_{\mathrm{TV}} \leq B_0(T^{-1} + T^{-1/2}) + C_0(\mathtt{M} + \bar{\gamma}^{1/2})\exp[D_0 T]$;

where $T = \sum_{k=1}^N \gamma_k$, $\bar{\gamma} = \sup_{k\in\{1,\dots,N\}} \gamma_k$ *and* $\mathcal{L}(X_0)$ *is the distribution of* X_0 *given in* (5).

Condition (8) ensures that the neural network approximates the score with a given precision $\mathtt{M} \geq 0$. Under this assumption and conditions on p_{data}, Theorem 1 states how the Markov chain defined by (5) approximates p_{data} in the total variation norm $\|\cdot\|_{\mathrm{TV}}$. In both cases, $\alpha = 0$ and $\alpha > 0$, the error consists of two terms. The first term decreases with $T \geq 0$ and corresponds to the error between p_T and p_{prior}. The second term stems from the error between the continuous-time process (7) with initialization $\mathbf{Y}_0 \sim p_{\text{prior}}$ and its discrete-time approximation (5). Those bounds show that there is a trade-off between the mixing properties of the Markov chain which increases with α, and the quality of the discrete-time approximation which deteriorates as α and T increase. To the best of our knowledge Theorem 1 is the first result assessing the convergence of SGM methods.

3 Diffusion Schrödinger Bridge and Generative Modeling

3.1 Schrödinger Bridges

The Schrödinger Bridge (SB) problem is a classical problem appearing in applied mathematics, optimal control and probability; see *e.g.* Föllmer (1988); Léonard (2014b); Chen et al. (2021b). In the discrete-time setting, it takes the following (dynamic) form. Consider as *reference* density $p(x_{0:N})$ given by (1), describing the process adding noise to the data. We aim to find $\pi^\star \in \mathscr{P}_{N+1}$ such that

$$\pi^\star = \arg\min\{\mathrm{KL}(\pi|p)\ :\ \pi \in \mathscr{P}_{N+1},\ \pi_0 = p_{\text{data}},\ \pi_N = p_{\text{prior}}\}\,. \tag{9}$$

Assuming $\pi^\star$ is available, a generative model can be obtained by sampling $X_N \sim p_{\text{prior}}$, followed by the reverse-time dynamics $X_k \sim \pi^\star_{k|k+1}(\cdot|X_{k+1})$ for $k \in \{N-1, \dots, 0\}$. Before deriving a method to approximate $\pi^\star$ in Section 3.2, we highlight some desirable features of Schrödinger bridges.

Static Schrödinger bridge problem. First, we recall that the dynamic formulation (9) admits a static analogue. Using *e.g.* Léonard (2014a, Theorem 2.4), the following decomposition holds for any $\pi \in \mathscr{P}_{N+1}$, $\mathrm{KL}(\pi|p) = \mathrm{KL}(\pi_{0,N}|p_{0,N}) + \mathbb{E}_{\pi_{0,N}}[\mathrm{KL}(\pi_{|0,N}|p_{|0,N})]$, where for any $\mu \in \mathscr{P}_{N+1}$ we have $\mu = \mu_{0,N}\mu_{|0,N}$ with $\mu_{|0,N}$ the conditional distribution of $X_{1:N-1}$ given X_0, X_N[2]. Hence we have $\pi^\star(x_{0:N}) = \pi^{\mathrm{s},\star}(x_0, x_N)p_{|0,N}(x_{1:N-1}|x_0, x_N)$ where $\pi^{\mathrm{s},\star} \in \mathscr{P}_2$ with marginals $\pi_0^{\mathrm{s},\star}$ and $\pi_N^{\mathrm{s},\star}$ is the solution of the static SB problem

$$\pi^{\mathrm{s},\star} = \arg\min \left\{\mathrm{KL}(\pi^{\mathrm{s}}|p_{0,N}) \,:\, \pi^{\mathrm{s}} \in \mathscr{P}_2,\ \pi_0^{\mathrm{s}} = p_{\text{data}},\ \pi_N^{\mathrm{s}} = p_{\text{prior}}\right\}. \tag{10}$$

Link with optimal transport. Under mild assumptions, the static SB problem can be seen as an entropy-regularized optimal transport problem since (10) is equivalent to

$$\pi^{\mathrm{s},\star} = \arg\min \left\{-\mathbb{E}_{\pi^{\mathrm{s}}}[\log p_{N|0}(X_N|X_0)] - \mathrm{H}(\pi^{\mathrm{s}}) \,:\, \pi^{\mathrm{s}} \in \mathscr{P}_2,\ \pi_0^{\mathrm{s}} = p_{\text{data}},\ \pi_N^{\mathrm{s}} = p_{\text{prior}}\right\}.$$

If $p_{k+1|k}(x_{k+1}|x_k) = \mathcal{N}(x_{k+1}; x_k, \sigma^2_{k+1})$ as in Song and Ermon (2019), then $p_{N|0}(x_N|x_0) = \mathcal{N}(x_N; x_0, \sigma^2)$ with $\sigma^2 = \sum_{k=1}^N \sigma_k^2$ which induces a quadratic cost and

$$\pi^{\mathrm{s},\star} = \arg\min \left\{\mathbb{E}_{\pi^{\mathrm{s}}}[||X_0 - X_N||^2] - 2\sigma^2\mathrm{H}(\pi^{\mathrm{s}}) \,:\, \pi^{\mathrm{s}} \in \mathscr{P}_2,\ \pi_0^{\mathrm{s}} = p_{\text{data}},\ \pi_N^{\mathrm{s}} = p_{\text{prior}}\right\}.$$

Mikami (2004) showed that $\pi^{\mathrm{s},\star} \to \pi^\star_{\mathcal{W}}$ weakly and $2\sigma^2\,\mathrm{KL}(\pi^{\mathrm{s},\star}|p_{0,N}) \to \mathcal{W}_2^2(p_{\text{data}}, p_{\text{prior}})$ as $\sigma \to 0$, where $\pi^\star_{\mathcal{W}}$ is the optimal transport plan between p_{data} and p_{prior} and $\mathcal{W}_2$ is the 2-Wasserstein distance. Note that the transport cost $c(x, x') = -\log p_{N|0}(x'|x)$ is not necessarily symmetric.

3.2 Iterative Proportional Fitting and Time Reversal

In all but trivial cases, the SB problem does not admit a closed-form solution. However, it can be solved using Iterative Proportional Fitting (IPF) (Fortet, 1940; Kullback, 1968; Ruschendorf et al., 1995) which is defined by the following recursion for $n \in \mathbb{N}$ with initialization $\pi^0 = p$ given in (1):

$$\begin{aligned}\pi^{2n+1} &= \arg\min \left\{\mathrm{KL}(\pi|\pi^{2n}) \,:\, \pi \in \mathscr{P}_{N+1},\ \pi_N = p_{\text{prior}}\right\}, \\ \pi^{2n+2} &= \arg\min \left\{\mathrm{KL}(\pi|\pi^{2n+1}) \,:\, \pi \in \mathscr{P}_{N+1},\ \pi_0 = p_{\text{data}}\right\}.\end{aligned} \tag{11}$$

This sequence is well-defined if there exists $\tilde{\pi} \in \mathscr{P}_{N+1}$ such that $\tilde{\pi}_0 = p_{\text{data}}$, $\tilde{\pi}_N = p_{\text{prior}}$ and $\mathrm{KL}(\tilde{\pi}|p) < +\infty$. A standard representation of π^n is obtained by updating the joint density p using potential functions, see Section S4.2 for details. However, this representation of the IPF iterates is difficult to numerically approximate as it not only requires approximating the potentials but also requires evaluating π_N^n pointwise. Our methodology builds upon an alternative representation that is better suited to numerical approximations in the context of generative modeling where one has access to samples of p_{data} and p_{prior}.

Proposition 2. *Assume that* $\mathrm{KL}(p_{\text{data}} \otimes p_{\text{prior}}|p_{0,N}) < +\infty$. *Then for any* $n \in \mathbb{N}$, π^{2n} *and* π^{2n+1} *admit positive densities w.r.t. the Lebesgue measure denoted as* p^n *resp.* q^n *and for any* $x_{0:N} \in \mathcal{X}$, *we have* $p^0(x_{0:N}) = p(x_{0:N})$ *and*

$$q^n(x_{0:N}) = p_{\text{prior}}(x_N)\prod_{k=0}^{N-1} p^n_{k|k+1}(x_k|x_{k+1}),\ p^{n+1}(x_{0:N}) = p_{\text{data}}(x_0)\prod_{k=0}^{N-1} q^n_{k+1|k}(x_{k+1}|x_k).$$

In practice we have access to $p^n_{k+1|k}$ and $q^n_{k|k+1}$. Hence, to compute $p^n_{k|k+1}$ and $q^n_{k+1|k}$ we use

$$p^n_{k|k+1}(x_k|x_{k+1}) = \frac{p^n_{k+1|k}(x_{k+1}|x_k)p^n_k(x_k)}{p^n_{k+1}(x_{k+1})},\quad q^n_{k+1|k}(x_{k+1}|x_k) = \frac{q^n_{k|k+1}(x_k|x_{k+1})q^n_{k+1}(x_{k+1})}{q^n_k(x_k)}.$$

To the best of our knowledge, this representation of the IPF iterates has surprisingly neither been presented nor explored in the literature. One may interpret these formulas as follows. At iteration $2n$, we have $\pi^{2n} = p^n$ with $p^0 = p$ given by the noising process (1). This forward process initalized with $p_0^n = p_{\text{data}}$ defines reverse-time transitions $p^n_{k|k+1}$, which, when combined with an initialization p_{prior} at step N defines the reverse-time process $\pi^{2n+1} = q^n$. The forward transitions $q^n_{k+1|k}$ associated to q^n are then used to obtain $\pi^{2n+2} = p^{n+1}$. IPF then iterates this procedure.

[2] See Section S4.1 for a rigorous presentation using the disintegration theorem for probability measures.

3.3 Diffusion Schrödinger Bridge as Iterative Mean-Matching Proportional Fitting

To approximate the IPF recursion defined in Proposition 2, we use similar approximations to Section 2.1. If at step $n \in \mathbb{N}$ we have $p^n_{k+1|k}(x_{k+1}|x_k) = \mathcal{N}(x_{k+1}; x_k + \gamma_{k+1} f^n_k(x_k), 2\gamma_{k+1}\mathbf{I})$ where $p^0 = p$ and $f^0_k = f$, then we can approximate the reverse-time transitions in Proposition 2 by

$$\begin{aligned} q^n_{k|k+1}(x_k|x_{k+1}) &= p^n_{k+1|k}(x_{k+1}|x_k) \exp[\log p^n_k(x_k) - \log p^n_{k+1}(x_{k+1})] \\ &\approx \mathcal{N}(x_k; x_{k+1} + \gamma_{k+1} b^n_{k+1}(x_{k+1}), 2\gamma_{k+1}\mathbf{I}), \end{aligned}$$

with $b^n_{k+1}(x_{k+1}) = -f^n_k(x_{k+1}) + 2\nabla \log p^n_{k+1}(x_{k+1})$. Similarly, we can approximate the forward transitions in Proposition 2 by $p^{n+1}_{k+1|k}(x_{k+1}|x_k) \approx \mathcal{N}(x_{k+1}; x_k + \gamma_{k+1} f^{n+1}_k(x_k), 2\gamma_{k+1}\mathbf{I})$ with $f^{n+1}_k(x_k) = -b^n_{k+1}(x_k) + 2\nabla \log q^n_k(x_k)$. Hence we have $f^{n+1}_k(x_k) = f^n_k(x_k) - 2\nabla \log p^n_{k+1}(x_k) + 2\nabla \log q^n_k(x_k)$ (and similarly for b^{n+1}_k). It follows that, one could estimate f^{n+1}_k, b^{n+1}_k using score-matching to approximate $\{\nabla \log p^i_{k+1}(x)\}_{i=0}^n$, $\{\nabla \log q^i_k(x)\}_{i=0}^n$ individually. However, this approach is prohibitively costly in terms of memory and compute, see Section S5 for details. Each of the reverse-time dynamics may also be learnt sequentially through a variational approach as in Ho et al. (2020), see Section S4.4. We follow an alternate approach coined *mean-matching* which avoids the aforementioned difficulties. This approach yields convenient loss terms from Proposition 3.

Proposition 3. *Assume that for any $n \in \mathbb{N}$ and $k \in \{0, \dots, N-1\}$,*

$$q^n_{k|k+1}(x_k|x_{k+1}) = \mathcal{N}(x_k; B^n_{k+1}(x_{k+1}), 2\gamma_{k+1}\mathbf{I}),\ p^n_{k+1|k}(x_{k+1}|x_k) = \mathcal{N}(x_{k+1}; F^n_k(x_k), 2\gamma_{k+1}\mathbf{I}),$$

with $B^n_{k+1}(x) = x + \gamma_{k+1} b^n_{k+1}(x)$, $F^n_k(x) = x + \gamma_{k+1} f^n_k(x)$ for any $x \in \mathbb{R}^d$. Then we have for any $n \in \mathbb{N}$ and $k \in \{0, \dots, N-1\}$

$$B^n_{k+1} = \arg\min_{\mathrm{B} \in \mathrm{L}^2(\mathbb{R}^d, \mathbb{R}^d)} \mathbb{E}_{p^n_{k,k+1}}[\|\mathrm{B}(X_{k+1}) - (X_{k+1} + F^n_k(X_k) - F^n_k(X_{k+1}))\|^2], \quad (12)$$

$$F^{n+1}_k = \arg\min_{\mathrm{F} \in \mathrm{L}^2(\mathbb{R}^d, \mathbb{R}^d)} \mathbb{E}_{q^n_{k,k+1}}[\|\mathrm{F}(X_k) - (X_k + B^n_{k+1}(X_{k+1}) - B^n_{k+1}(X_k))\|^2]. \quad (13)$$

Proposition 3 shows how one can recursively approximate B^n_{k+1} and F^{n+1}_k. In practice, we use neural networks $B_{\beta^n}(k, x) \approx B^n_k(x)$ and $F_{\alpha^n}(k, x) \approx F^n_k(x)$.

Algorithm 1 Diffusion Schrödinger Bridge

for $n \in \{0, \dots, L\}$ **do**
  while not converged **do**
    Sample $\{X^j_k\}_{k,j=0}^{N,M}$, where $X^j_0 \sim p_{\text{data}}$, and $X^j_{k+1} = F_{\alpha^n}(k, X^j_k) + \sqrt{2\gamma_{k+1}} Z^j_{k+1}$
    Compute $\hat{\ell}^b_n(\beta^n)$ approximating (12)
    $\beta^n \leftarrow$ Gradient Step$(\hat{\ell}^b_n(\beta^n))$
  end while
  while not converged **do**
    Sample $\{X^j_k\}_{k,j=0}^{N,M}$, where $X^j_N \sim p_{\text{prior}}$, and $X^j_{k-1} = B_{\beta^n}(k, X^j_k) + \sqrt{2\gamma_k} \tilde{Z}^j_k$
    Compute $\hat{\ell}^f_{n+1}(\alpha^{n+1})$ approximating (13)
    $\alpha^{n+1} \leftarrow$ Gradient Step$(\hat{\ell}^f_{n+1}(\alpha^{n+1}))$
  end while
end for
Output: (α^{L+1}, β^L)

Network parameters α^n, β^n are learnt through gradient descent to minimize empirical versions of the sum over k of the loss functions given by (12) and (13) computed using M samples and denoted $\hat{\ell}^b_n(\beta)$ and $\hat{\ell}^f_{n+1}(\alpha)$. The resulting algorithm approximating L IPF iterations is called Diffusion Schrödinger Bridge (DSB) and is summarized in Algorithm 1 with $Z^j_k, \tilde{Z}^j_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \mathbf{I})$, see Figure 1 for an illustration. This algorithm is initialized using the reference dynamics $f_{\alpha^0}(k, x) = f(x)$. Once β^L is learnt we can easily approximately sample from p_{data} by sampling $X_N \sim p_{\text{prior}}$ and then using $X_{k-1} = B_{\beta^L}(k, X_k) + \sqrt{2\gamma_k} Z_k$ with $Z_k \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \mathbf{I})$. The resulting samples X_0 will be approximately distributed from p_{data}.

Although the DSB requires learning a sequence of network parameters, α^n, β^n, fewer diffusion steps are needed compared to standard SGM and hence the networks are quicker to train. In addition, as detailed in Section S9, the initial β^0 may be trained efficiently similar to previous SGM methods and then $\alpha^{n+1}, \beta^{n+1}$ are refinements of α^n, β^n hence may be fine-tuned from previous iterations.

3.4 Convergence of Iterative Proportional Fitting

In this section, we investigate the theoretical properties of IPF. When the state-space is discrete and finite (Franklin and Lorenz, 1989; Peyré and Cuturi, 2019) or in the case where p_{data} and p_{prior}

are compactly supported (Chen et al., 2016), IPF converges at a geometric rate w.r.t. the Hilbert-Birkhoff metric, see Lemmens and Nussbaum (2014) for a definition. Other than recent work by Léger (2020), only qualitative results exist in the general case where p_{data} or p_{prior} is not compactly supported (Ruschendorf et al., 1995; Rüschendorf and Thomsen, 1993). We establish here quantitative convergence of IPF in this non-compact setting as well as novel monotonicity results. We require only the following mild assumption.

A1. $p_N, p_{\text{prior}} > 0$, $\mathrm{H}(p_{\text{prior}}) < +\infty$, $\int_{\mathbb{R}^d} |\log(p_{N|0}(x_N|x_0))| p_{\text{data}}(x_0) p_{\text{prior}}(x_N) \mathrm{d}x_0 \mathrm{d}x_N < +\infty$.

Assumption **A**1 is satisfied in all of our experimental settings. We recall that for $\mu, \nu \in \mathscr{P}(\mathsf{E})$ with $(\mathsf{E}, \mathcal{E})$ a measurable space, the Jeffreys divergence is given by $\mathrm{J}(\mu, \nu) = \mathrm{KL}(\mu|\nu) + \mathrm{KL}(\nu|\mu)$.

Proposition 4. *Assume* **A**1. *Then* $(\pi^n)_{n\in\mathbb{N}}$ *is well-defined and for any* $n \geq 1$ *we have*

$$\mathrm{KL}(\pi^{n+1}|\pi^n) \leq \mathrm{KL}(\pi^{n-1}|\pi^n), \qquad \mathrm{KL}(\pi^n|\pi^{n+1}) \leq \mathrm{KL}(\pi^n|\pi^{n-1}).$$

In addition, $(\|\pi^{n+1} - \pi^n\|_{\mathrm{TV}})_{n\in\mathbb{N}}$ *and* $(\mathrm{J}(\pi^{n+1}, \pi^n))_{n\in\mathbb{N}}$ *are non-increasing. Finally, we have* $\lim_{n\to+\infty} n \left\{\mathrm{KL}(\pi_0^n|p_{\text{data}}) + \mathrm{KL}(\pi_N^n|p_{\text{prior}})\right\} = 0$.

A more general result with additional monotonicity properties is given in Section S6. Under similar assumptions, Léger (2020, Corollary 1) derives that $\mathrm{KL}(\pi_0^n|p_0) \leq C/n$ with $C \geq 0$ using a Bregman divergence gradient descent perspective. In contrast, our proof relies only on tools from information geometry. In addition, we improve the convergence rate and show that $(\pi^n)_{n\in\mathbb{N}}$ converges in total variation towards π^∞, *i.e.* we not only obtain the convergence of the marginals but the convergence of the joint distribution. Under restrictive conditions on p_{data} and p_{prior}, Ruschendorf et al. (1995) show that π^∞ is the Schrödinger bridge. In the following proposition we avoid this assumption using results on automorphisms of measures (Beurling, 1960).

Proposition 5. *Assume* **A**1. *Then there exists a solution* $\pi^\star \in \mathscr{P}_{N+1}$ *to the SB problem and we have* $\lim_{n\to+\infty} \|\pi^n - \pi^\infty\|_{\mathrm{TV}} = 0$ *with* $\pi^\infty \in \mathscr{P}_{N+1}$. *Let* $h = p_{0,N}/(p_0 \otimes p_N)$ *and assume that* $h \in \mathrm{C}((\mathbb{R}^d)^2, (0, +\infty))$ *and that there exist* $\Phi_0, \Phi_N \in \mathrm{C}(\mathbb{R}^d, (0, +\infty))$ *such that*

$$\int_{\mathbb{R}^d\times\mathbb{R}^d} (|\log h(x_0, x_N)| + |\log \Phi_0(x_0)| + |\log \Phi_N(x_N)|) p_{\text{data}}(x_0) p_{\text{prior}}(x_N) \mathrm{d}x_0 \mathrm{d}x_N < +\infty,$$

with $h(x_0, x_N) \leq \Phi_0(x_0)\Phi_N(x_N)$. *If* p *is absolutely continuous w.r.t.* π^∞ *then* $\pi^\infty = \pi^\star$.

Proposition 5 extends previous IPF convergence results without the assumption that the mapping h is lower bounded, see Ruschendorf et al. (1995); Chen et al. (2016). Our assumption on h can be relaxed and replaced by a tighter condition on π^∞, see Section S6.2. Proposition 4 suggests a convergence of order $o(n)$ for the IPF in the non compact setting. However, in some situations, we recover geometric convergence rates with explicit dependency w.r.t. the problem constants, see Section S7.

3.5 Continuous-time IPF

We describe an IPF algorithm for solving SB problems in continuous-time. We show that DSB proposed in Algorithm 1 can be seen as a discretization of this IPF. Given a reference measure $\mathbb{P} \in \mathscr{P}(\mathcal{C})$, the continuous formulation of the SB involves solving the following problem

$$\Pi^\star = \arg\min \left\{\mathrm{KL}(\Pi|\mathbb{P}) \,:\, \Pi \in \mathscr{P}(\mathcal{C}),\, \Pi_0 = p_{\text{data}},\, \Pi_T = p_{\text{prior}}\right\}, \quad T = \textstyle\sum_{k=0}^{N-1} \gamma_{k+1}.$$

Similarly to (11), we define the IPF $(\Pi^n)_{n\in\mathbb{N}}$ with $\Pi^0 = \mathbb{P}$ associated with (6) and for any $n \in \mathbb{N}$

$$\Pi^{2n+1} = \arg\min \left\{\mathrm{KL}(\Pi|\Pi^{2n}) \,:\, \Pi \in \mathscr{P}(\mathcal{C}),\, \Pi_T = p_{\text{prior}}\right\},$$
$$\Pi^{2n+2} = \arg\min \left\{\mathrm{KL}(\Pi|\Pi^{2n+1}) \,:\, \Pi \in \mathscr{P}(\mathcal{C}),\, \Pi_0 = p_{\text{data}}\right\}.$$

One can show that for any $n \in \mathbb{N}$, $\Pi^n = \pi^{\mathrm{s},n}\mathbb{P}_{|0,T}$, with $(\pi^{\mathrm{s},n})_{n\in\mathbb{N}}$ the IPF for the static SB problem. In particular, Proposition 4 and Proposition 5 extend to the continuous IPF framework. In what follows, for any $\mathbb{P} \in \mathscr{P}(\mathcal{C})$, we define $\mathbb{P}^R$ as the reverse-time measure, *i.e.* for any $\mathsf{A} \in \mathcal{B}(\mathcal{C})$ we have $\mathbb{P}^R(\mathsf{A}) = \mathbb{P}(\mathsf{A}^R)$ where $\mathsf{A}^R = \{t \mapsto \omega(T-t) \,:\, \omega \in \mathsf{A}\}$. The following result is the continuous counterpart of Proposition 2 and states that each IPF iteration is associated with a diffusion, showing that DSB can be seen as a discretization of the continuous IPF.

Proposition 6. *Assume* **A**1 *and that there exist* $\mathbb{M} \in \mathscr{P}(\mathcal{C})$, $U \in \mathrm{C}^1(\mathbb{R}^d, \mathbb{R})$, $C \geq 0$ *such that for any* $n \in \mathbb{N}$, $x \in \mathbb{R}^d$, $\mathrm{KL}(\Pi^n|\mathbb{M}) < +\infty$, $\langle x, \nabla U(x)\rangle \geq -C(1 + \|x\|^2)$ *and* $\mathbb{M}$ *is associated with*

$$\mathrm{d}\mathbf{X}_t = -\nabla U(\mathbf{X}_t)\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t, \tag{14}$$

with $\mathbf{X}_0$ *distributed according to the invariant distribution of* (14). *Then, for any* $n \in \mathbb{N}$ *we have:*

(a) $(\Pi^{2n+1})^R$ *is associated with* $\mathrm{d}\mathbf{Y}_t^{2n+1} = b_{T-t}^n(\mathbf{Y}_t^{2n+1})\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t$ *with* $\mathbf{X}_0^{2n+1} \sim p_{\text{prior}}$*;*

(b) Π^{2n+2} *is associated with* $\mathrm{d}\mathbf{X}_t^{2n+2} = f_t^{n+1}(\mathbf{X}_t^{2n+2})\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t$ *with* $\mathbf{X}_0^{2n+2} \sim p_{\text{data}}$*;*
where for any $n \in \mathbb{N}$*,* $t \in [0, T]$ *and* $x \in \mathbb{R}^d$*,* $b_t^n(x) = -f_t^n(x) + 2\nabla \log p_t^n(x)$*,* $f_t^{n+1}(x) = -b_t^n(x) + 2\nabla \log q_t^n(x)$*, with* $f_t^0(x) = f(x)$*, see (6), and* p_t^n*,* q_t^n *the densities of* Π_t^{2n} *and* Π_t^{2n+1}*.*

4 Experiments

Two dimensional toy experiments. We evaluate the validity of our approach on toy two dimensional examples. Contrary to existing SGM approaches we do *not* require that the number of steps is large enough for $p_N \approx p_{\text{prior}}$ to hold. We use a fully connected network with positional encoding (Vaswani et al., 2017) to approximate B_k^n and F_k^n, see Section S10.1 for details about our implementation. Animated plots of the DSB iterations may be found online on our project webpage[3]. In Figure 2, we illustrate the benefits of DSB over classical SGM. We fix $f(x) = \alpha x$ and choose $p_{\text{prior}} = \mathcal{N}(0, \sigma_{\text{data}}^2 \mathbf{I})$, hence $\alpha = 1/\sigma_{\text{data}}^2$ where σ_{data}^2 is the variance of the dataset. We let $N = 20$ and $\gamma_k = 0.01$, *i.e.* $T = 0.2$. Since T is low, the noising process does not satisfy $p_N \approx p_{\text{prior}}$ and the reverse-time process obtained after the first DSB iteration (corresponding to original SGM methods) does not yield a satisfactory generative model. However, multiple iterations of DSB improve the quality of the synthesis. We emphasize that even though in theory Schrödinger bridges allow for T to be arbitrary small, we observe that decreasing values of T require an increasing number of DSB iterations to obtain valid generative models. We discuss this trade-off and present additional experiments in Section S10.1.

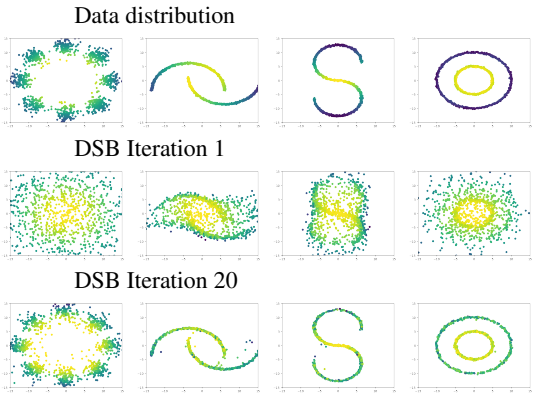

Figure 2: Data distributions p_{data} vs distribution at final time $T = 0.2$ after 1 and 20 DSB iterations.

Generative modeling. DSB is the first practical algorithm for approximating the solution to the SB problem in high dimension. Whilst our implementation does not yet compete with state-of-the-art methods, we show promising results with fewer diffusion steps compared to initial SGMs (Song and Ermon, 2019) and demonstrate its performance on MNIST (LeCun and Cortes, 2010) and CelebA (Liu et al., 2015). A reduced U-net architecture based on Nichol and Dhariwal (2021) is used to approximate B_k^n and F_k^n. Further details are given in Section S10.2. Our method is validated on downscaled CelebA in Figure 3. Figure 4 illustrates qualitative improvement over 8 DSB iterations with as few as $N = 12$ diffusion steps. Note, as shown in the Section S10.2, we obtain better results with higher N yet still significantly fewer steps than in the original SGM procedures (Song and Ermon, 2020, 2019) which use $N = 100$. Figure 4 also shows good diversity of generated samples (red) and coverage of the original dataset (blue) as shown by the two dimensional representation in the latent space of a pre-trained Variational Auto-Encoder (VAE). Figure 5 illustrates how the sample quality, measured quantitatively in terms of Fréchet Inception Distance (FID) (Heusel et al., 2017), improves with the number of DSB iterations for various numbers of steps N.

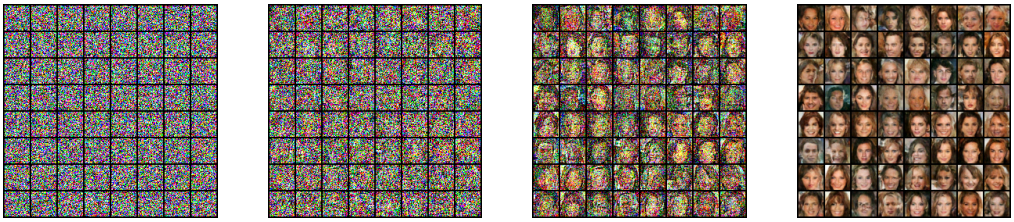
$t = 0$ $t = 0.31$ $t = 0.60$ $t = 0.63$

Figure 3: Generative model for CelebA 32×32 after 10 DSB iterations with $N = 50$ ($T = 0.63$)

[3] https://anon284.github.io/

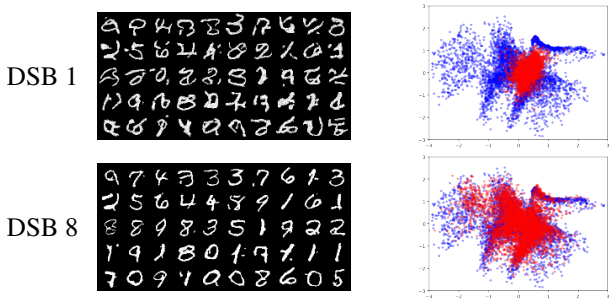

Figure 4: Generated samples ($N = 12$) and two-dimensional visualization of samples (red) compared to original MNIST data (blue) using pre-trained VAE

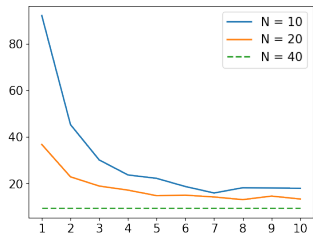

Figure 5: FID vs DSB Iterations. Green dashed line: baseline FID obtained with 1 DSB iteration and $N = 40$.

Datasets interpolation. Schrödinger bridges not only allow us to reduce the number of iterations in SGM methods but also enable flexibility in the choice of the prior density p_{prior} which is not necessarily Gaussian contrary to previous works on SGM. In particular, our approach is still valid if p_{prior} is any other data distribution p'_{data}. In this case DSB converges towards a bridge between p_{data} and p'_{data}, see Figure 6. These experiments pave the way towards high-dimensional optimal transport between arbitrary data distributions.

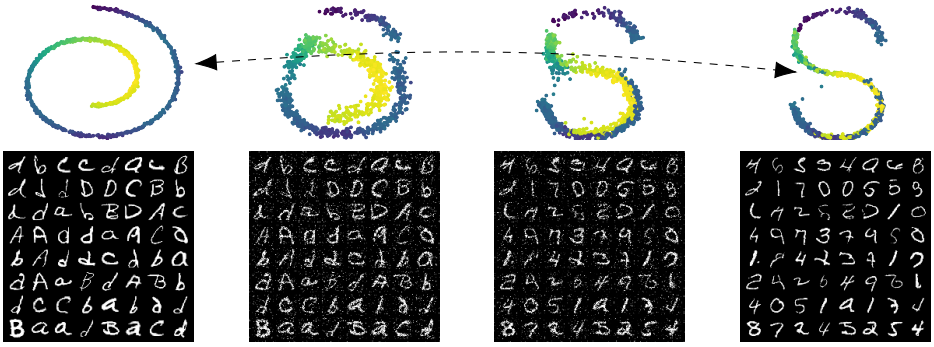

Figure 6: First row: Swiss-roll to S-curve (2D). Iteration 9 of DSB with $T = 1$ ($N = 50$). From left to right: $t = 0, 0.4, 0.6, 1$. Second row: EMNIST (Cohen et al., 2017) to MNIST. Iteration 10 of DSB with $T = 1.5$ ($N = 30$). From left to right: $t = 0, 0.4, 1.25, 1.5$.

5 Discussion

Score-based generative modeling (SGM) may be viewed as the first stage in approximating a solution to the Schrödinger bridge problem. Through this interpretation we have developed a novel methodology, the Diffusion Schrödinger Bridge (DSB), that extends initial SGM approaches and allows one to perform generative modeling with fewer diffusion steps. DSB complements recent techniques used to speed up existing SGM algorithms which rely on either different noise schedules (Nichol and Dhariwal, 2021; San-Roman et al., 2021) or knowledge distillation (Luhman and Luhman, 2021). Additionally, as the solution of the Schrödinger problem is a diffusion, it is possible as in Song et al. (2021, Section 4.3) to obtain an equivalent neural ordinary differential equation that admits the same marginals as the diffusion but enables exact likelihood computation. From a theoretical point of view, we have provided the first convergence result for SGM methods and derived new state-of-the-art convergence bounds for IPF as well as novel monotonicity results. We have demonstrated DSB on generative modeling and interpolation tasks. Finally, while motivated by generative modeling, DSB is much more widely applicable as it can be thought of as the continuous state-space counterpart of the celebrated Sinkhorn algorithm (Cuturi, 2013; Peyré and Cuturi, 2019). For example, DSB could be used to solve multi-marginal Schrödinger bridges problems (Di Marino and Gerolin, 2020), compute Wasserstein barycenters, find the minimizers of entropy-regularized Gromov-Wasserstein problems (Mémoli, 2011) or perform domain adaptation in continuous state-spaces.

References

Bernton, E., Heng, J., Doucet, A., and Jacob, P. E. (2019). Schrödinger bridge samplers. *arXiv preprint arXiv:1912.13170*.

Beurling, A. (1960). An automorphism of product measures. *Annals of Mathematics*, 72(1):189–200.

Cai, R., Yang, G., Averbuch-Elor, H., Hao, Z., Belongie, S., Snavely, N., and Hariharan, B. (2020). Learning gradient fields for shape generation. *European Conference on Computer Vision*.

Cattiaux, P., Conforti, G., Gentil, I., and Léonard, C. (2021). Time reversal of diffusion processes under a finite entropy condition. *arXiv preprint arXiv:2104.07708*.

Chen, N., Zhang, Y., Zen, H., Weiss, R. J., Norouzi, M., and Chan, W. (2021a). Wavegrad: Estimating gradients for waveform generation. *International Conference on Learning Representations*.

Chen, Y., Georgiou, T., and Pavon, M. (2016). Entropic and displacement interpolation: a computational approach using the Hilbert metric. *SIAM Journal on Applied Mathematics*, 76(6):2375–2396.

Chen, Y., Georgiou, T. T., and Pavon, M. (2021b). Optimal transport in systems and control. *Annual Review of Control, Robotics, and Autonomous Systems*, 4.

Cohen, G., Afshar, S., Tapson, J., and van Schaik, A. (2017). EMNIST: an extension of MNIST to handwritten letters. *arXiv preprint arXiv:1702.05373*.

Cuturi, M. (2013). Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*.

Dessein, A., Papadakis, N., and Deledalle, C.-A. (2017). Parameter estimation in finite mixture models by regularized optimal transport: A unified framework for hard and soft clustering. *arXiv preprint arXiv:1711.04366*.

Dhariwal, P. and Nichol, A. (2021). Diffusion models beat GAN on image synthesis. *arXiv preprint arXiv:2105.05233*.

Di Marino, S. and Gerolin, A. (2020). An optimal transport approach for the Schrödinger bridge problem and convergence of Sinkhorn algorithm. *Journal of Scientific Computing*, 85(2):1–28.

Durkan, C. and Song, Y. (2021). On maximum likelihood training of score-based generative models. *arXiv preprint arXiv:2101.09258*.

Föllmer, H. (1985). An entropy approach to the time reversal of diffusion processes. In *Stochastic Differential Systems: Filtering and Control*, pages 156–163. Springer.

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

Fortet, R. (1940). Résolution d'un système d'équations de M. Schrödinger. *Journal de Mathématiques Pures et Appliqués*, 1:83–105.

Franklin, J. and Lorenz, J. (1989). On the scaling of multidimensional matrices. *Linear Algebra and Its Applications*, 114:717–735.

Gao, R., Song, Y., Poole, B., Wu, Y. N., and Kingma, D. P. (2020). Learning energy-based models by diffusion recovery likelihood. *arXiv preprint arXiv:2012.08125*.

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

Haussmann, U. G. and Pardoux, E. (1986). Time reversal of diffusions. *The Annals of Probability*, 14(4):1188–1205.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). GANs trained by a two time-scale update rule converge to a local Nash equilibrium. *arXiv preprint arXiv:1706.08500*.

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

Hoogeboom, E., Nielsen, D., Jaini, P., Forré, P., and Welling, M. (2021). Argmax flows and multinomial diffusion: Towards non-autoregressive language models. *arXiv preprint arXiv:2102.05379*.

Hyvärinen, A. and Dayan, P. (2005). Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4).

Jolicoeur-Martineau, A., Piché-Taillefer, R., Tachet des Combes, R., and Mitliagkas, I. (2021). Adversarial score matching and improved sampling for image generation. *International Conference on Learning Representations*.

Kong, Z., Ping, W., Huang, J., Zhao, K., and Catanzaro, B. (2021). Diffwave: A versatile diffusion model for audio synthesis. *International Conference on Learning Representations*.

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

LeCun, Y. and Cortes, C. (2010). MNIST handwritten digit database.

Léger, F. (2020). A gradient descent perspective on Sinkhorn. *Applied Mathematics & Optimization*, pages 1–13.

Lemmens, B. and Nussbaum, R. D. (2014). Birkhoff's version of Hilbert's metric and its applications in analysis. *Handbook of Hilbert Geometry*, pages 275–303.

Léonard, C. (2014a). Some properties of path measures. In *Séminaire de Probabilités XLVI*, pages 207–230. Springer.

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

Liu, Z., Luo, P., Wang, X., and Tang, X. (2015). Deep learning face attributes in the wild. In *International Conference on Computer Vision*.

Luhman, E. and Luhman, T. (2021). Knowledge distillation in iterative generative models for improved sampling speed. *arXiv preprint arXiv:2101.02388*.

Luhman, T. and Luhman, E. (2020). Diffusion models for handwriting generation. *arXiv preprint arXiv:2011.06704*.

Mémoli, F. (2011). Gromov-Wasserstein distances and the metric approach to object matching. *Foundations of Computational Mathematics*, 11(4):417–487.

Mikami, T. (2004). Monge's problem with a quadratic cost by the zero-noise limit of h-path processes. *Probability Theory and Related Fields*, 129(2):245–260.

Nichol, A. and Dhariwal, P. (2021). Improved denoising diffusion probabilistic models. *arXiv preprint arXiv:2102.09672*.

Niu, C., Song, Y., Song, J., Zhao, S., Grover, A., and Ermon, S. (2020). Permutation invariant graph generation via score-based generative modeling. In *International Conference on Artificial Intelligence and Statistics*.

Pavon, M., Trigila, G., and Tabak, E. G. (2021). The data-driven Schrödinger bridge. *Communications on Pure and Applied Mathematics*, 74:1545–1573.

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

Popov, V., Vovk, I., Gogoryan, V., Sadekova, T., and Kudinov, M. (2021). Grad-tts: A diffusion probabilistic model for text-to-speech. *arXiv preprint arXiV:2105.06337*.

Reich, S. (2019). Data assimilation: the Schrödinger perspective. *Acta Numerica*, 28:635–711.

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

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

Saharia, C., Ho, J., Chan, W., Salimans, T., Fleet, D. J., and Norouzi, M. (2021). Image super-resolution via iterative refinement. *arXiv preprint arXiv:2104.07636*.

San-Roman, R., Nachmani, E., and Wolf, L. (2021). Noise estimation for generative diffusion models. *arXiv preprint arXiv:2104.02600*.

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

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

Song, J., Meng, C., and Ermon, S. (2020). Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

Song, Y. and Ermon, S. (2019). Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*.

Song, Y. and Ermon, S. (2020). Improved techniques for training score-based generative models. *Advances in Neural Information Processing Systems*.

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

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *arXiv preprint arXiv:1706.03762*.

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

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] The Schrödinger bridge methodology is introduced in Section 3 and our main algorithm in Algorithm 1.
   (b) Did you describe the limitations of your work? [Yes] In Section 4 we discuss the limitations of our algorithm.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] Our contribution is primarily theoretical and methodological. Our infrastructure and compute resources do not permit us to compete with state-of-the-art for generative modeling and hence presents little negative societal impact beyond existing methods in the literature.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] All proofs and additional results for the sake of completeness are postponed to the supplementary
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See our project webpage.

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] We give full details in the supplementary material.

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] For FID values we report the standard deviation in the supplementary material. Since, the standard deviation is negligible w.r.t the mean value of the FID score we do not report it in the main document.

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes] We cite Nichol and Dhariwal (2021) as we use both of their model implementation and Jolicoeur-Martineau et al. (2021); Song and Ermon (2020) for data loaders

(b) Did you mention the license of the assets? [Yes] Licences are included on our anonymous project github which will be shared in supplementary material.

(c) Did you include any new assets either in the supplemental material or as a URL? [Yes] See our anonymous project webpage and github link will be shared in the supplementary material.

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] The code and datasets are open source.

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] The datasets are either toy 2 dimensional datasets or classical image datasets such as MNIST and CelebA.

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]