Deep Generative Learning via Schrödinger Bridge

Gefei Wang [1] **Yuling Jiao** [2] **Qian Xu** [3] **Yang Wang** [1 4] **Can Yang** [1 4]

Abstract

We propose to learn a generative model via entropy interpolation with a Schrödinger Bridge. The generative learning task can be formulated as interpolating between a reference distribution and a target distribution based on the Kullback-Leibler divergence. At the population level, this entropy interpolation is characterized via an SDE on $[0, 1]$ with a time-varying drift term. At the sample level, we derive our Schrödinger Bridge algorithm by plugging the drift term estimated by a deep score estimator and a deep density ratio estimator into the Euler-Maruyama method. Under some mild smoothness assumptions of the target distribution, we prove the consistency of both the score estimator and the density ratio estimator, and then establish the consistency of the proposed Schrödinger Bridge approach. Our theoretical results guarantee that the distribution learned by our approach converges to the target distribution. Experimental results on multimodal synthetic data and benchmark data support our theoretical findings and indicate that the generative model via Schrödinger Bridge is comparable with state-of-the-art GANs, suggesting a new formulation of generative learning. We demonstrate its usefulness in image interpolation and image inpainting.

1. Introduction

Deep generative models have achieved enormous success in learning the underlying high-dimensional data distribution from samples. They have various applications in machine learning, like image-to-image translation (Zhu et al., 2017; Choi et al., 2020), semantic image editing (Zhu et al., 2016; Shen et al., 2020) and audio synthesis (Van Den Oord et al., 2016; Prenger et al., 2019). Most of existing generative models seek to learn a nonlinear function to transform a simple reference distribution to the target distribution as data generating mechanisms. They can be categorized as either likelihood-based models or implicit generative models.

Likelihood-based models, such as variational auto-encoders (VAEs) (Kingma & Welling, 2014) and flow-based methods (Dinh et al., 2015), optimize the negative log-likelihood or its surrogate loss, which is equivalent to minimize the Kullback–Leibler (KL) divergence between the target distribution and the generated distribution. Although their ability to learn flexible distributions is restricted by the way to model the probability density, many works have been established to alleviate this problem and achieved appealing results (Makhzani et al., 2016; Tolstikhin et al., 2018; Razavi et al., 2019; Dinh et al., 2017; Papamakarios et al., 2017; Kingma & Dhariwal, 2018; Behrmann et al., 2019). As a representative of implicit generative models, generative adversarial networks (GANs) use a min-max game objective to learn the target distribution. It has been shown that vanilla GAN (Goodfellow et al., 2014) minimizes the Jensen-Shannon (JS) divergence between the target distribution and the generated distribution. To generalize vanilla GAN, researchers consider some other criterions including more general f-divergences (Nowozin et al., 2016), 1-Wasserstein distance (Arjovsky et al., 2017) and maximum mean discrepancy (MMD) (Bińkowski et al., 2018). Meanwhile, recent progress on designing network architectures (Radford et al., 2016; Zhang et al., 2019) and training techniques (Karras et al., 2018; Brock et al., 2019) has enabled GANs to produce impressive high-quality images.

Despite the extraordinary performance of generative models (Razavi et al., 2019; Kingma & Dhariwal, 2018; Brock et al., 2019; Karras et al., 2019), there still exists a gap between the empirical success and the theoretical justification of these methods. For likelihood-based models, consistency results require that the data distribution is within the model family, which is often hard to hold in practice (Kingma & Welling, 2014). Recently, new generative models have been developed from different perspectives, such as gradient flow in a measure space in which GAN can be covered as a special case (Gao et al., 2019; Arbel et al., 2019) and

[1]Department of Mathematics, The Hong Kong University of Science and Technology, Hong Kong, China [2]School of Mathematics and Statistics, Wuhan University, Wuhan, China [3]AI Group, WeBank Co., Ltd., Shenzhen, China [4]Guangdong-Hong Kong-Macao Joint Laboratory for Data-Driven Fluid Mechanics and Engineering Applications, The Hong Kong University of Science and Technology, Hong Kong, China. Correspondence to: Yuling Jiao <yulingjiaomath@whu.edu.cn>, Can Yang <macyang@ust.hk>.

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

stochastic differential equations (SDE) (Song & Ermon, 2019; 2020; Song et al., 2021). To push a simple initial distribution to the target one, however, these methods (Gao et al., 2019; Arbel et al., 2019; Liutkus et al., 2019; Song & Ermon, 2019; 2020; Block et al., 2020) require the evolving time to go to infinity at the population level. Therefore, these methods require a strong assumption to achieve model consistency: the target must be log-concave or satisfy the log-Sobolev inequality.

To fill the gap, we propose a Schrödinger Bridge approach to learn generative models. Schrödinger Bridge tackles the problem by interpolating a reference distribution to a target distribution based on the Kullback-Leibler divergence. The Schrödinger Bridge can be formulated via an SDE on a finite time interval $[0, 1]$ with a time-varying drift term. At the population level, we can solve the SDE using the standard Euler-Maruyama method. At the sample level, we derive our Schrödinger Bridge algorithm by plugging the drift term into the Euler-Maruyamma method, where the drift term can be accurately estimated by a deep score network. The major contributions of this work are as follows:

- From the theoretical perspective, we prove the consistency of the Schrödinger Bridge approach under the some mild smoothness assumptions of the target distribution. Our theory guarantees that the learned distribution converges to the target. To achieve model consistency, existing theories rely on strong assumptions, e.g., the target must be log-concave or satisfy some error bound conditions, such as the log-Sobolev inequality. These assumptions may not hold in practice.

- From the algorithmic perspective, we develop a novel two-stage approach to make the theory of Schrödinger Bridge work in practice, where the first stage effectively learns a smoothed version of the target distribution and the second stage drives the smoothed one to the target distribution. Figure 1 gives an overview of our two-stage algorithm.

- Through synthetic data, we demonstrate that our Schrödinger Bridge approach can stably learn multimodal distribution, while GANs are often highly unstable and prone to miss modes (Che et al., 2017). We also show that the proposed approach achieves comparable performance with state-of-the-art GANs on benchmark data.

In summary, we believe that our work suggests a new formulation of generative models.

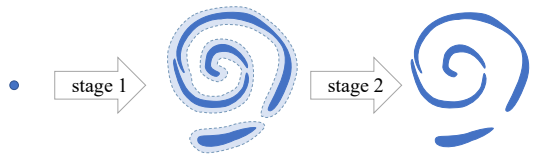


Figure 1. Overview of our two-stage algorithm. Stage 1 drives samples at $\mathbf{0}$ (left) to a smoothed data distribution (middle), and stage 2 learns the underlying target data distribution (right) with samples produced by stage 1. Stage 1 and stage 2 are achieved through the two different Schrodinger Bridges with theoretically guaranteed performance.

2. Background

Let's first recall some background on Schrödinger Bridge problem adopted from (Léonard, 2014; Chen et al., 2020).

Let $\Omega = C([0,1], \mathbb{R}^d)$ be the space of $\mathbb{R}^d$-valued continuous functions on time interval $[0, 1]$. Denote $X = (X_t)_{t\in[0,1]}$ as the canonical process on Ω, where $X_t(\omega) = \omega_t$, $\omega = (\omega_s)_{s\in[0,1]} \in \Omega$. The canonical σ-field on Ω is then generated as $\mathscr{F} = \sigma(X_t, t \in [0,1]) = \big\{\{\omega : (X_t(\omega))_{t\in[0,1]} \in H\} : H \in \mathscr{B}(\mathbb{R}^d)\big\}$. Denote $\mathcal{P}(\Omega)$ as the space of probability measures on the path space Ω, and $\mathbf{W}_\tau^{\mathbf{x}} \in \mathcal{P}(\Omega)$ as the Wiener measure with variance τ whose initial marginal is $\delta_{\mathbf{x}}$. The law of the reversible Brownian motion, is then defined as $\mathbf{P}_\tau = \int \mathbf{W}_\tau^{\mathbf{x}} \mathrm{d}\mathbf{x}$, which is an unbounded measure on Ω. One can observe that, $\mathbf{P}_\tau$ has a marginal coincides with the Lebesgue measure $\mathscr{L}$ at each t.

Schrödinger (1932) studied the problem of finding the most likely random evolution between two continuous probability distributions $\mu, \nu \in \mathcal{P}(\mathbb{R}^d)$. Nowadays, people call the study of Schrödinger as the Schrödinger Bridge problem (SBP). In fact, SBP can be further formulated as seeking a probability law on a path space that interpolates between μ and ν, such that the probability law is close to the prior law of the Brownian diffusion in the sense of relative entropy (Jamison, 1975; Léonard, 2014), i.e., finding a path measure $\mathbf{Q}^* \in \mathcal{P}(\Omega)$ with marginal $\mathbf{Q}_t^* = (X_t)_\# \mathbf{Q}^* = \mathbf{Q}^* \circ X_t^{-1}, t \in [0,1]$ such that

$$\mathbf{Q}^* \in \arg\min_{\mathbf{Q}\in\mathcal{P}(\Omega)} \mathbb{D}_{\mathrm{KL}}(\mathbf{Q}||\mathbf{P}_\tau),$$

and

$$\mathbf{Q}_0 = \mu, \mathbf{Q}_1 = \nu,$$

where $\mu, \nu \in \mathcal{P}(\mathbb{R}^d)$, relative entropy $\mathbb{D}_{\mathrm{KL}}(\mathbf{Q}||\mathbf{P}_\tau) = \int \log(\frac{d\mathbf{Q}}{d\mathbf{P}_\tau}) d\mathbf{Q}$ if $\mathbf{Q} \ll \mathbf{P}_\tau$ (i.e. $\mathbf{Q}$ is absolutely continuous w.r.t. $\mathbf{P}_\tau$), and $\mathbb{D}_{\mathrm{KL}}(\mathbf{Q}||\mathbf{P}_\tau) = \infty$ otherwise. The following results characterize the solution to SBP.

Theorem 1 *(Léonard, 2014) If $\mu, \nu \ll \mathscr{L}$, then SBP admits a unique solution $\mathbf{Q}^* = f^*(X_0)g^*(X_1)\mathbf{P}_\tau$,*

where f^*, g^* are $\mathscr{L}$-measurable nonnegative functions on $\mathbb{R}^d$ satisfying the Schrödinger system $\begin{cases} f^*(\mathbf{x})\mathbb{E}_{\mathbf{P}_\tau}\left[g^*\left(X_1\right) \mid X_0=\mathbf{x}\right]=\frac{\mathrm{d}\mu}{\mathrm{d}\mathscr{L}}(\mathbf{x}), & \mathscr{L}-a.e. \\ g^*(\mathbf{y})\mathbb{E}_{\mathbf{P}_\tau}\left[f^*\left(X_0\right) \mid X_1=\mathbf{y}\right]=\frac{\mathrm{d}\nu}{\mathrm{d}\mathscr{L}}(\mathbf{y}), & \mathscr{L}-a.e. \end{cases}$

Besides $\mathbf{Q}^*$, we can also characterize the density of the time-marginals of $\mathbf{Q}^*$, i.e. $\frac{d\mathbf{Q}_t^*}{d\mathscr{L}}(\mathbf{x})$.

Let $q(\mathbf{x})$ and $p(\mathbf{y})$ be the density of μ and ν respectively, and $h_\tau(s,\mathbf{x},t,\mathbf{y})=[2\pi\tau(t-s)]^{-d/2}\exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\tau(t-s)}\right)$ be the transition density of the Wiener process. Then we have $\mathbb{E}_{\mathbf{P}_\tau}\left[f^*\left(X_0\right) \mid X_1=\mathbf{y}\right]=\int h_\tau(0,\mathbf{x},1,\mathbf{y})f_0(\mathbf{x})\mathrm{d}\mathbf{x}$, $\mathbb{E}_{\mathbf{P}_\tau}\left[g^*\left(X_1\right) \mid X_0=\mathbf{x}\right]=\int h_\tau(0,\mathbf{x},1,\mathbf{y})g_1(\mathbf{y})\mathrm{d}\mathbf{y}$. The above Schrödinger system is equivalent to

$$\begin{cases} f^*(\mathbf{x})\int h_\tau(0,\mathbf{x},1,\mathbf{y})g_1(\mathbf{y})\mathrm{d}\mathbf{y}=q(\mathbf{x}), \\ g^*(\mathbf{y})\int h_\tau(0,\mathbf{x},1,\mathbf{y})f_0(\mathbf{x})\mathrm{d}\mathbf{x}=p(\mathbf{y}). \end{cases}$$

Denote $f_0(\mathbf{x})=f^*(\mathbf{x})$, $g_1(\mathbf{y})=g^*(\mathbf{y})$,

$$f_1(\mathbf{y})=\int h_\tau(0,\mathbf{x},1,\mathbf{y})f_0(\mathbf{x})\mathrm{d}\mathbf{x},$$
$$g_0(\mathbf{x})=\int h_\tau(0,\mathbf{x},1,\mathbf{y})g_1(\mathbf{y})\mathrm{d}\mathbf{y}.$$

The Schrödinger system in Theorem 1 can also be characterized by

$$q(\mathbf{x})=f_0(\mathbf{x})g_0(\mathbf{x}),\ \ p(\mathbf{y})=f_1(\mathbf{y})g_1(\mathbf{y})$$

with the following forward and backward time harmonic equations (Chen et al., 2020)

$$\begin{cases} \partial_t f_t(\mathbf{x})=\frac{\tau\Delta}{2}f_t(\mathbf{x}), \\ \partial_t g_t(\mathbf{x})=-\frac{\tau\Delta}{2}g_t(\mathbf{x}), \end{cases} \qquad \text{on } (0,1)\times\mathbb{R}^d.$$

Let q_t denote marginal density of $\mathbf{Q}_t^*$, then it can be represented (Chen et al., 2020) by the product of g_t and f_t defined as $q_t(\mathbf{x})=\frac{d\mathbf{Q}_t^*}{d\mathscr{L}}(\mathbf{x})$, and $q_t(\mathbf{x})=f_t(\mathbf{x})g_t(\mathbf{x})$.

There are also dynamic formulations of SBP. Let $\mathcal{U}$ consist of admissible Markov controls with finite energy. The following theorem shows that, the vector field

$$\begin{aligned} \mathbf{u}_t^*&=\tau\mathbf{v}_t^*=\tau\nabla_{\mathbf{x}}\log g_t(\mathbf{x}) \\ &=\tau\nabla_{\mathbf{x}}\log\int h_\tau(t,\mathbf{x},1,\mathbf{y})g_1(\mathbf{y})\mathrm{d}\mathbf{y} \end{aligned} \tag{1}$$

solves such a stochastic control problem:

Theorem 2 *(Dai Pra, 1991)*

$$\mathbf{u}_t^*(\mathbf{x})\in\arg\min_{\mathbf{u}\in\mathcal{U}}\mathbb{E}\left[\int_0^1\frac{1}{2}\|\mathbf{u}_t\|^2\mathrm{d}t\right]$$

s.t.

$$\begin{cases} \mathrm{d}\mathbf{x}_t=\mathbf{u}_t\mathrm{d}t+\sqrt{\tau}\mathrm{d}\mathbf{w}_t, \\ \mathbf{x}_0\sim q(\mathbf{x}), \quad \mathbf{x}_1\sim p(\mathbf{x}). \end{cases} \tag{2}$$

According to Theorem 2, the dynamics determined by the SDE in (2) with a time-varying drift term $\mathbf{u}_t^*$ in (1) will make the particles sampled from the initial distribution μ evolve to the particles drawn from the target distribution ν in the unit time interval. This nice property is what we need in generative learning because we want to learn the underlying target distribution ν via pushing forward a simple reference distribution μ. Theorem 2 also indicates that such a solution has minimum energy in terms of quadratic cost.

3. Generative Learning via Schrödinger Bridge

In generative learning, we observe i.i.d. data $\mathbf{x}_1,...,\mathbf{x}_n$ from an unknown distribution $p_{\text{data}}\in\mathcal{P}(\mathbb{R}^d)$. The underlying distribution p_{data} often has multi-modes or lies on a low-dimensional manifold, which may cause difficulty to learn from simple distribution such as Gaussian or Dirac measure supported on a single point. To make the generative learning task easy to handle, we can first learn a smoothed version of p_{data} from the simple reference distribution, say

$$q_\sigma(\mathbf{x})=\int p_{\text{data}}(\mathbf{y})\Phi_\sigma(\mathbf{x}-\mathbf{y})\mathrm{d}\mathbf{y},$$

where $\Phi_\sigma(\cdot)$ is the density of $\mathcal{N}(\mathbf{0},\sigma^2\mathbf{I})$, the variance of Gaussian noise σ^2 controls the smoothness of q_σ. Then we learn p_{data} starting from q_σ. At the population level, this idea can be done via Schrödinger Bridge from the point of view of the stochastic control problem (Theorem 2). To be precise, we have the following theorem.

Theorem 3 *Define the density ratio $f(\mathbf{x})=\frac{q_\sigma(\mathbf{x})}{\Phi_{\sqrt{\tau}}(\mathbf{x})}$. Then for the SDE*

$$\mathrm{d}\mathbf{x}_t=\tau\nabla\log\mathbb{E}_{\mathbf{z}\sim\Phi_{\sqrt{\tau}}}[f(\mathbf{x}_t+\sqrt{1-t}\mathbf{z})]\mathrm{d}t+\sqrt{\tau}\mathrm{d}\mathbf{w}_t \tag{3}$$

with initial condition $\mathbf{x}_0=\mathbf{0}$, we have $\mathbf{x}_1\sim q_\sigma(\mathbf{x})$.

And, for the SDE

$$\mathrm{d}\mathbf{x}_t=\sigma^2\nabla\log q_{\sqrt{1-t}\sigma}(\mathbf{x}_t)\mathrm{d}t+\sigma\mathrm{d}\mathbf{w}_t \tag{4}$$

with initial condition $\mathbf{x}_0\sim q_\sigma(\mathbf{x})$, we have $\mathbf{x}_1\sim p_{\text{data}}(\mathbf{x})$.

According to Theorem 3, at the population level, the target p_{data} can be learned from the Dirac mass supported at $\mathbf{0}$ through two SDEs (3) and (4) in the unit time interval [0,1]. The main feature of the SDEs (3) and (4) is that both drift terms are time-varying, which is different from classical Langevin SDEs with time-invariant drift terms (Song & Ermon, 2019; 2020). The benefit of time-varying drift terms is that the dynamics in (3) and (4) will push the initial distributions to the target distributions in a unit time interval, while the classical Langevin SDE needs time to go to infinity.

3.1. Estimation of the drift terms

Based on Theorem 3, we can run the Euler-Maruyama method to solve the SDEs (3) and (4) and get particles approximately drawn from the targets (Higham, 2001). However, the drift terms in Theorem 3 depend on the underlying target. To make the Euler-Maruyama method practical, we need to estimate the two drift terms in (3) and (4). In Eq. (3), some calculation shows that

$$\begin{aligned}&\nabla \log \mathbb{E}_{\mathbf{z}\sim\Phi_{\sqrt{\tau}}}[f(\mathbf{x}+\sqrt{1-t}\mathbf{z})]\\ =&\frac{\mathbb{E}_{\mathbf{z}\sim\Phi_{\sqrt{\tau}}}\left[f(\mathbf{x}+\sqrt{1-t}\mathbf{z})\nabla \log f(\mathbf{x}+\sqrt{1-t}\mathbf{z})\right]}{\mathbb{E}_{\mathbf{z}\sim\Phi_{\sqrt{\tau}}}[f(\mathbf{x}+\sqrt{1-t}\mathbf{z})]},\end{aligned} \tag{5}$$

and

$$\nabla \log f(\mathbf{x}) = \nabla \log q_\sigma(\mathbf{x}) + \mathbf{x}/\tau.$$

Let $\hat{f}$ and $\widehat{\nabla \log q_\sigma}$ be the estimators of the density ratio f and the score of $q_\sigma(\mathbf{x})$, respectively. After plugging them into (5), we can obtain an estimator of the drift term in (3) by computing the expectation with Monte Carlo approximation.

Now we consider obtaining the estimator of density ratio $\hat{f}$, via minimizing the logistic regression loss $\mathcal{L}_{\text{logistic}}(r) = \mathbb{E}_{q_\sigma(\mathbf{x})}\log(1+\exp(-r(\mathbf{x})))+\mathbb{E}_{\Phi_{\sqrt{\tau}}(\mathbf{x})}\log(1+\exp(r(\mathbf{x})))$. By setting the first variation to zero, the optimal solution is given by

$$r^*(\mathbf{x}) = \log\frac{q_\sigma(\mathbf{x})}{\Phi_{\sqrt{\tau}}(\mathbf{x})}.$$

Therefore, given samples $\widetilde{\mathbf{x}}_1, ..., \widetilde{\mathbf{x}}_n$ from $q_\sigma(\mathbf{x})$, which can be obtained by adding Gaussian noise drawn from Φ_σ on $\mathbf{x}_1, ..., \mathbf{x}_n \sim p_{\text{data}}$, and samples $\mathbf{z}_1, ..., \mathbf{z}_n$ from $\Phi_{\sqrt{\tau}}$, we can estimate the density ratio $f(\mathbf{x})$ by

$$\hat{f}(\mathbf{x}) = \exp(\hat{r}_\phi(\mathbf{x})), \tag{6}$$

where $\hat{r}_\phi \in \mathcal{NN}_\phi$ is the neural network that minimizes the empirical loss:

$$\begin{aligned}\hat{r}_\phi \in \arg\min_{r_\phi\in\mathcal{NN}_\phi}\frac{1}{n}\sum_{i=1}^{n}[&\log(1+\exp(-r_\phi(\widetilde{\mathbf{x}}_i)))\\ &+\log(1+\exp(r_\phi(\mathbf{z}_i)))].\end{aligned} \tag{7}$$

Next, we consider estimating the time-varying drift term in (4), i.e., $\nabla \log q_{\sqrt{1-t}\sigma}(\mathbf{x})$ for $t \in [0,1]$. To do so, we build a deep network as the score estimator for $\nabla \log q_{\tilde{\sigma}}(\mathbf{x})$ with $\tilde{\sigma}$ varying in $[0,\sigma]$. Vincent (2011) showed that, explicitly matching the score by minimizing the objective

$$\frac{1}{2}\mathbb{E}_{q_{\tilde{\sigma}}(\mathbf{x})}\|\mathbf{s}_\theta(\mathbf{x},\tilde{\sigma}) - \nabla_{\mathbf{x}}\log q_{\tilde{\sigma}}(\mathbf{x})\|^2$$

is equivalent to minimizing the denoising score matching objective

$$\begin{aligned}&\frac{1}{2}\mathbb{E}_{p_{\text{data}}(\mathbf{x})}\mathbb{E}_{\mathcal{N}(\tilde{\mathbf{x}};\mathbf{x},\tilde{\sigma}^2\mathbf{I})}\|\mathbf{s}_\theta(\tilde{\mathbf{x}},\tilde{\sigma}) - \nabla_{\tilde{\mathbf{x}}}\log q_{\tilde{\sigma}}(\tilde{\mathbf{x}}|\mathbf{x})\|^2\\ =&\frac{1}{2}\mathbb{E}_{p_{\text{data}}(\mathbf{x})}\mathbb{E}_{\mathcal{N}(\tilde{\mathbf{x}};\mathbf{x},\tilde{\sigma}^2\mathbf{I})}\left\|\mathbf{s}_\theta(\tilde{\mathbf{x}},\tilde{\sigma}) + \frac{\tilde{\mathbf{x}}-\mathbf{x}}{\tilde{\sigma}^2}\right\|^2.\end{aligned}$$

Thus we build the score estimator following Song & Ermon (2019; 2020) as

$$\hat{\mathbf{s}}_\theta(\cdot,\cdot) \in \arg\min_{\mathbf{s}_\theta\in\mathcal{NN}_\theta}\mathcal{L}(\theta), \tag{8}$$

$$\mathcal{L}(\theta) = \frac{1}{m}\sum_{j=1}^{m}\lambda(\tilde{\sigma}_j)\mathcal{L}_{\tilde{\sigma}_j}(\theta), \tag{9}$$

$$\mathcal{L}_{\tilde{\sigma}_j}(\theta) = \sum_{i=1}^{n}\left\|\mathbf{s}_\theta(\mathbf{x}_i+\mathbf{z}_i,\tilde{\sigma}) + \frac{\mathbf{z}_i}{\tilde{\sigma}_j^2}\right\|^2/n,$$

variance terms $\tilde{\sigma}_j^2, j = 1,\dots,m$ are i.i.d. samples from $\text{Uniform}[0,\sigma^2]$ with sample size m, $\lambda(\tilde{\sigma}) = \tilde{\sigma}^2$ is a non-negative scaling factor to ensure all the summands in (9) have the same scale, and $\mathbf{z}_i, i = 1, ..., n$ are i.i.d. from $\Phi_{\tilde{\sigma}}$.

At last, we establish the consistencies of the deep density ratio estimator $\hat{f}(\mathbf{x}) = \exp(\hat{r}_\phi(\mathbf{x}))$ and the deep score estimator $\widehat{\nabla \log q_{\tilde{\sigma}}}(\mathbf{x}) = \hat{\mathbf{s}}_\theta(\mathbf{x};\tilde{\sigma})$ in Theorem 4 and Theorem 5, respectively.

Theorem 4 *Assume that the support of $p_{\text{data}}(\mathbf{x})$ is contained in a compact set, and $f(\mathbf{x})$ is Lipschitz continuous and bounded. Set the depth $\mathcal{D}$, width $\mathcal{W}$, and size $\mathcal{S}$ of $\mathcal{NN}_\phi$ as*

$$\mathcal{D} = \mathcal{O}(\log(n)), \mathcal{W} = \mathcal{O}(n^{\frac{d}{2(2+d)}}/\log(n)),$$

$$\mathcal{S} = \mathcal{O}(n^{\frac{d-2}{d+2}}\log(n)^{-3}).$$

Then $\mathbb{E}[\|\hat{f}(\mathbf{x}) - f(\mathbf{x})\|_{L^2(p_{\text{data}})}] \to 0$ as $n \to \infty$.

Theorem 5 *Assume that $p_{\text{data}}(\mathbf{x})$ is differentiable with bounded support, and $\nabla \log q_{\tilde{\sigma}}(\mathbf{x})$ is Lipschitz continuous and bounded for $(\tilde{\sigma},\mathbf{x}) \in [0,\sigma]\times\mathbb{R}^d$. Set the depth $\mathcal{D}$, width $\mathcal{W}$, and size $\mathcal{S}$ of $\mathcal{NN}_\theta$ as*

$$\mathcal{D} = \mathcal{O}(\log(n)), \mathcal{W} = \mathcal{O}(\max\{n^{\frac{d}{2(2+d)}}/\log(n), d\}),$$

$$\mathcal{S} = \mathcal{O}(dn^{\frac{d-2}{d+2}}\log(n)^{-3}).$$

Then $\mathbb{E}[\|\|\widehat{\nabla \log q_{\tilde{\sigma}}}(\mathbf{x}) - \nabla \log q_{\tilde{\sigma}}(\mathbf{x})\|_2\|_{L^2(q_{\tilde{\sigma}})}] \to 0$ as $m, n \to \infty$.

Algorithm 1 Sampling

Input: $\hat{f}(\cdot)$, $\hat{\mathbf{s}}_\theta(\cdot,\cdot)$, τ, σ, N_1, N_2, N_3

Initialize particles as $\mathbf{x}_0 = \mathbf{0}$ stage 1
for $k = 0$ **to** $N_1 - 1$ **do**
  Sample $\{\mathbf{z}_i\}_{i=1}^{2N_3}, \boldsymbol{\epsilon}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\tilde{\mathbf{x}}_i = \mathbf{x}_k + \sqrt{\tau\left(1 - \frac{k}{N_1}\right)}\mathbf{z}_i$, $i = 1, ..., N_3$
  $\mathbf{b}(\mathbf{x}_k) = \frac{\sum_{i=1}^{N_3} \hat{f}(\tilde{\mathbf{x}}_i)[\hat{\mathbf{s}}_\theta(\tilde{\mathbf{x}}_i, \sigma) + \sqrt{\left(1-\frac{k}{N_1}\right)/\tau}\mathbf{z}_i]}{\sum_{i=N_3+1}^{2N_3} \hat{f}(\tilde{\mathbf{x}}_i)} + \frac{\mathbf{x}_k}{\tau}$.
  $\mathbf{x}_{k+1} = \mathbf{x}_k + \frac{\tau}{N_1}\mathbf{b}(\mathbf{x}_k) + \sqrt{\frac{\tau}{N_1}}\boldsymbol{\epsilon}_k$.
end for
Set $\mathbf{x}_0 = \mathbf{x}_{N_1}$ stage 2
for $k = 0$ **to** $N_2 - 1$ **do**
  Sample $\boldsymbol{\epsilon}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathbf{b}(\mathbf{x}_k) = \hat{\mathbf{s}}_\theta(\mathbf{x}_k, \sqrt{1 - \frac{k}{N_2}}\sigma)$
  $\mathbf{x}_{k+1} = \mathbf{x}_k + \frac{\sigma^2}{N_2}\mathbf{b}(\mathbf{x}_n) + \frac{\sigma}{\sqrt{N_2}}\boldsymbol{\epsilon}_k$
end for

return $\mathbf{x}_{N_2}$

3.2. Schrödinger Bridge Algorithm

With the two estimators $\hat{f}$ and $\widehat{\nabla \log q_{\tilde{\sigma}}}$, we can use the Euler-Maruyama method to approximate numerical solutions of SDEs (3) and (4). Let N_1 and N_2 be the number of uniform grids on the time interval $[0, 1]$. In stage 1, we start from $\mathbf{0}$ and run Euler-Maruyama for (3) with the estimated $\hat{f}$ and $\widehat{\nabla \log q_\sigma}$ in the drift term to obtain samples that follow q_σ approximately. In stage 2, we start with the samples from q_σ and run another Euler-Maruyama for (4) with the estimated time-varying drift term $\widehat{\nabla \log q_{\tilde{\sigma}}}$. We summarize our two-stage Schrödinger Bridge algorithm in 1.

Interestingly, the second stage of our proposed Schrödinger Bridge algorithm 1 recovers the reverse-time Variance Exploding (VE) SDE algorithm proposed in Song et al. (2021), if their annealing scheme is chosen to be linear as $\sigma^2(t) = \sigma^2 \cdot t$. From this point of view, our Schrödinger Bridge algorithm also provides deeper understanding of annealing score based sampling, i.e., the reverse-time VE SDE algorithm (with a proper annealing scheme) proposed by Song et al. (2021) is equivalent to the Schrödinger Bridge SDE (4).

3.3. Consistency of Schrödinger Bridge Algorithm

Let

$$D_1(t, \mathbf{x}) = \nabla \log \mathbb{E}_{\mathbf{z}\sim\Phi_{\sqrt{\tau}}}[f(\mathbf{x} + \sqrt{1-t}\mathbf{z})],$$

$$D_2(t, \mathbf{x}) = \nabla \log q_{\sqrt{1-t}\sigma}(\mathbf{x})$$

be the drift terms. Denote

$$h_{\sigma,\tau}(\mathbf{x}_1, \mathbf{x}_2) = \exp\left(\frac{\|\mathbf{x}_1\|^2}{2\tau}\right) p_{\text{data}}(\mathbf{x}_1 + \sigma\mathbf{x}_2).$$

Now we establish the consistency of our Schrödinger Bridge algorithm which can drive a simple distribution to the target one. To this end, we need the following assumptions:

Assumption 1 $\text{supp}(p_{\text{data}})$ *is contained in a ball with radius* R*, and* $p_{\text{data}} > c > 0$ *on its support.*

Assumption 2 $\|D_i(t, \mathbf{x})\|^2 \le C_1(1 + \|\mathbf{x}\|^2)$, $\forall \mathbf{x} \in \text{supp}(p_{\text{data}})$, $t \in [0, 1]$, *where* $C_1 \in \mathbb{R}$ *is a constant.*

Assumption 3 $\|D_i(t_1, \mathbf{x}_1) - D_i(t_2, \mathbf{x}_2)\| \le C_2(\|\mathbf{x}_1 - \mathbf{x}_2\| + |t_1 - t_2|^{1/2})$, $\forall \mathbf{x}_1, \mathbf{x}_2 \in \text{supp}(p_{\text{data}}), t_1, t_2 \in [0, 1]$. $C_2 \in \mathbb{R}$ *is another constant.*

Assumption 4 $h_{\sigma,\tau}(\mathbf{x}_1, \mathbf{x}_2)$, $\nabla_{\mathbf{x}_1} h_{\sigma,\tau}(\mathbf{x}_1, \mathbf{x}_2)$, p_{data} *and* ∇p_{data} *are* L*-Lipschitz functions.*

Theorem 6 *Under Assumptions 1-4,*

$$\mathbb{E}[\mathcal{W}_2(\text{Law}(\mathbf{x}_{N_2}), p_{\text{data}})] \to 0, \text{ as } n, N_1, N_2, N_3 \to \infty,$$

where $\mathcal{W}_2$ *is the 2-Wasserstein distance between two distributions.*

The consistency of the proposed Schrödinger Bridge algorithm is mainly based on mild assumptions (such as smoothness and boundedness) without some restricted technical requirements that the target distribution has to be log-concave or fulfill the log-Sobolev inequality (Gao et al., 2021; Arbel et al., 2019; Liutkus et al., 2019; Block et al., 2020).

4. Related Work

We discuss connections and differences between our Schrödinger Bridge approach and existing related works.

Most of existing generative models, such as VAEs, GANs and flow-based methods, parameterize a transform map with a neural network G that minimizes an integral probability metric. Clearly, they are quite different from our proposal.

Recently, particle methods derived in the perspective of gradient flows in measure spaces or SDEs have been studied (Johnson & Zhang, 2018; Gao et al., 2019; Arbel et al., 2019; Song & Ermon, 2019; 2020; Song et al., 2021). Here we clarify the main differences of our Schrödinger Bridge approach and the above mentioned particle methods. The proposals in (Johnson & Zhang, 2018; Gao et al., 2019; Arbel et al., 2019) are derived based on the surrogate of the geodesic interpolation (Gao et al., 2021; Liutkus et al., 2019; Song & Ermon, 2019). They utilize the invariant measure of SDEs to model the generative task, resulting in an iteration scheme that looks similar to our Schrödinger Bridge.

However, the main difference lies that the drift terms of the Langevin SDEs in (Song & Ermon, 2019; 2020; Block et al., 2020) are time-invariant in contrast to the time-varying drift term in our formulation. As shown in Theorem 3, the benefit of the time-varying drift term is essential: the SDE of Schrödinger Bridge runs on a unit time interval $[0, 1]$ will recover the target distribution at the terminal time. However, the evolution measures of the above mentioned methods (Gao et al., 2019; Arbel et al., 2019; Song & Ermon, 2019; 2020; Block et al., 2020; Gao et al., 2021) only converge to the target when the time goes to infinity. Hence, some technical requirements are imposed to the target distribution, such as log-concave or the log-Sobolev inequality, to guarantee the consistency of Euler-Maruyama discretization. However, these assumptions may often be too strong to hold in real data analysis. We proposed a two-stage approach to make the Schrödinger Bridge formulation work in practice. We drive the Dirac distribution to a smoothed version of underlying distribution p_{data} in stage 1 and then learn p_{data} from the smoothed version in stage 2. Interestingly, the second stage of the proposed Schrödinger Bridge algorithm recovers the reverse-time Variance Exploding SDE algorithm (VE SDE) (Song et al., 2021) when their annealing scheme is linear, i.e., $\sigma^2(t) = \sigma^2 \cdot t$. Therefore, the analysis developed here also provides a theoretical justification of why the reverse-time VE SDE algorithm works well. However, their setting is $\sigma^2(t) = (\sigma^2_{\max})^t \cdot (\sigma^2_{\min})^{1-t}$. This implies that the end-time distribution of the reverse-time VE SDE is still a smoothed one (with noise level $\sigma_{\min}$), resulting in a barrier of establishing the consistency. Another fundamental difference between our approach and reverse-time VE SDE is that, the reverse-time VE SDE also need a smoothed distribution as the input of theoretically, but they only approximately use large Gaussian noises as the initialization of the denoising process. Stage 1 ensures our algorithm to learn samples from the smoothed data distribution in unit time, which is necessary for model consistency.

5. Experiments

In this section, we first employ two-dimensional toy examples to show the ability of our algorithm to learn multimodal distributions which may not satisfy log-Sobolev inequality. Next, we show that our algorithm is able to generate realistic image samples. We also demonstrate the effective of our approach by image interpolation and image inpainting. We use two benchmark datasets including CIFAR-10 (Krizhevsky et al., 2009) and CelebA (Liu et al., 2015). For CelebA, the images are center-cropped and resized to 64×64. Both of the datasets are normalized by first rescaling the pixel values to $[0, 1]$, and then substracting a mean vector $\bar{\mathbf{x}}$ estimated using 50,000 samples to center the data distributions at the origin. In our algorithm, the particles start from $\delta_{\mathbf{0}}$. To improve the performance, it is helpful to align the sample mean to the origin. After generation, we add the image mean $\bar{\mathbf{x}}$ back to the generated samples. More details on the hyperparameter settings and network architectures, and some additional experiments are provided in the supplementary material. The code for reproducing all our experiments is available at https://github.com/YangLabHKUST/DGLSB.

5.1. Setup

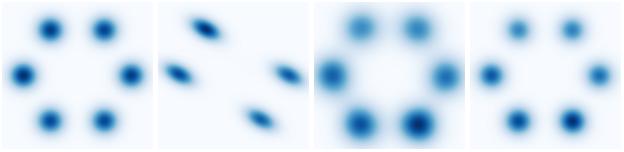
For the noise level σ, we set $\sigma = 1.0$ in this paper for generative tasks including both 2D example and CIFAR-10. In fact, the performance of our algorithm is insensitive to the choice of σ when σ is given in a reasonable range (the results with other σ values are shown in the supplementary material). We find that the performance of our algorithm is often among the best by setting $\sigma = 1.0$ for 32×32 images. The reason is that a very small σ can not make q_σ smooth enough and harms the performance of stage 1 while a very large σ brings more difficulty for our stage 2 to anneal the noise level down. For larger images like CelebA, as the dimensionality of samples is higher, we increase the noise level σ to 2.0. We also compare the results by varying the value of the variance of the Wiener measure τ for image generation. The numbers of grids are chosen as $N_1 = N_2 = 1,000$ for stage 1 and stage 2. We use sample size $N_3 = 1$ to estimate the drift term in stage 1 for both 2D toy examples and real images. In general, we find that a larger sample size N_3 does not significantly improve sample quality.

5.2. Learning 2D Multimodal Distributions

We demonstrate that our algorithm can effectively learn multimodal distributions. The distribution we adopt is a mixture of Gaussians with 6 components. Each of the components has a mean with a distance equaling to 5.0 from the origin, and a variance 0.01, as shown in Fig. 2(a). The components are relatively far away from each other. It is a very challenging task for GANs to learn this multimodal distribution because this distribution may not satisfy the log-Sobolev inequality. Fig. 2(b) shows the failure of vanilla GAN, where several modes are missed. However, Fig. 2(c) and 2(d) show that our algorithm is able to stably generate samples from the multimodal distribution without ignoring any of the modes. In Fig. 3, we compare the ground truth velocity fields induced by drift terms $D_1(t, \mathbf{x}), D_2(t, \mathbf{x})$ with the estimated velocity fields at the end of each stage. Our estimated drift terms are close to the ground truth except for the region with nearly zero probability density.

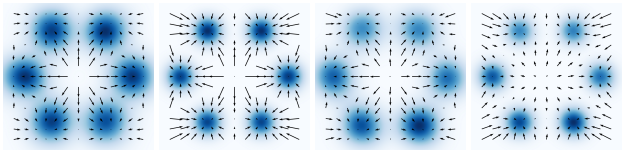
5.3. Effectiveness of Two Stages for Image Generation

Fig. 4 shows the particle evolution on CIFAR-10 in our algorithm, where the two stages are annotated with corresponding colors. It shows that our two-stage approach



(a) (b) (c) (d)

Figure 2. KDE plots for mixture of Gaussians with 5,000 samples. (a). Ground truth. (b). Distribution learned by vanilla GAN. (c). Distribution learned by the proposed method after stage 1 ($\tau = 5.0$). (d). Distribution learned by the proposed method after stage 2.



(a) (b) (c) (d)

Figure 3. Velocity fields. (a) and (b). Ground truth velocity fields at the end of stages 1 and 2. (c) and (d). Estimated velocity fields at the end of stages 1 and 2.

provides a valid path for the particles to move from the origin to the target distribution. A natural question is: what are the roles of stage 1 and stage 2 in the generative modeling, respectively? In this subsection, we design experiments to answer this question.

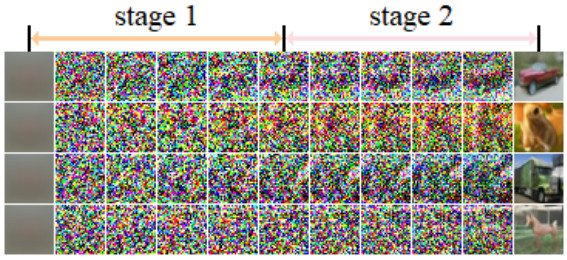
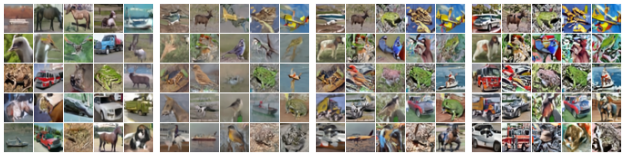


Figure 4. Particle evolution on CIFAR-10. The column in the center indicates particles obtained after stage 1.

We first evaluate the role of stage 1. For this purpose, we skip stage 1 but simply run stage 2 using non-informative Gaussian noises as the initial condition. Fig. 5 shows that the approach only using stage 2 generates worse image samples than the proposed two-stage approach. These results indicate that the role of stage 1 is to provide a better initial reference for stage 2. The role of stage 2 is easier to check: it is a Schrödinger Bridge from $q_\sigma(\mathbf{x})$ to the target distribution $p_{\text{data}}(\mathbf{x})$. In Fig. 6, we perturb real images with Gaussian noises of variance $\sigma^2 = 1.0$. Our stage 2 anneals the noise level to zero and drives the particles to the data distribution. Moreover, Fig. 6 also indicates that stage 2 not only recover the original images, but also generate images with some extent of diversity.



(a) (b) (c) (d)

Figure 5. Comparison with random image samples. (a). Samples produced by our algorithm with $\tau = 2.0$ (FID = 12.32). (b), (c), (d). Samples produced by stage 2 taking Gaussian noises with variance 1.0 (FID = 32.60), 1.5 (FID = 24.76), 2.0 (FID = 51.21) as input respectively.

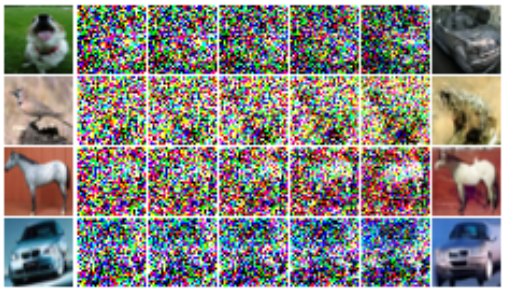


Figure 6. Denoising with stage 2 for perturbed real images.

5.4. Results

In this subsection, we evaluate our proposed approach on benchmark datasets. Fig. 7 presents the generated samples of our algorithm on CIFAR-10 and CelebA. Visually, our algorithm produces high-fidelity image samples which are competitive with real images. For quantitive evaluation, we employ Fréchet Inception Distance (FID) (Heusel et al., 2017) and Inception Score (IS) (Salimans et al., 2016) to compare our method with other benchmark methods.

We first compare the FID and IS on CIFAR-10 dataset, with τ increasing from 1.0 to 4.0 using 50,000 generated samples. Note that τ is the variance of the prior Wiener measure in stage 1, so it controls the behavior of the particle evolution from $\delta_{\mathbf{0}}$ to q_σ, and has an impact on the numerical results. To make the prior reasonable, we let $\tau_{\min} = \sigma^2 = 1.0$. The reason is that, if the particles strictly follow the prior law of the Brownian diffusion with variance τ in stage 1, the end time marginal will be $\mathcal{N}(\mathbf{0}, \tau\mathbf{I})$. A good choice of the prior should make $\mathcal{N}(\mathbf{0}, \tau\mathbf{I})$ close to the end time marginal q_σ which we are interested about. As shown in Table 1, our

Table 1. FID and Inception Score on CIFAR-10 for $\tau \in [1, 4]$.

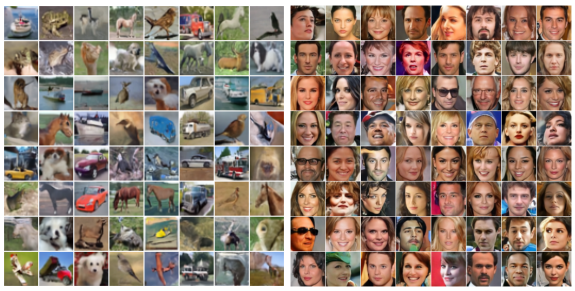
τ	1.0	1.5	2.0	2.5
FID	37.20	20.49	**12.32**	12.90
IS	6.52	7.65	**8.14**	7.99
τ	3.0	3.5	4.0	
FID	13.97	14.49	14.67	
IS	7.98	8.03	8.10	

Table 2. FID and Inception Scores on CIFAR-10.

MODELS	FID	IS
WGAN-GP	36.4	7.86±0.07
SN-SMMDGAN	25.0	7.3±0.1
SNGAN	21.7	8.22±0.05
NCSN	25.32	8.87±0.12
NCSNV2	10.87	8.40±0.07
OURS	**12.32**	**8.14±0.07**

algorithm achieves the best performance at $\tau = 2.0$. The results also indicate that our algorithm is stable with respect to the value of variance of the prior Wiener measure τ when $\tau \geq 2.0$. In general, reasonable choices of τ would result in relatively good generating performance.

Table 2 presents the FID and IS of our algorithm evaluating with 50,000 samples, as well as other state-of-the-art generative models including WGAN-GP (Gulrajani et al., 2017), SN-SMMDGAN(Arbel et al., 2018), SNGAN (Miyato et al., 2018), NCSN (Song & Ermon, 2019) and NCSNv2 (Song & Ermon, 2020) on CIFAR-10. Our algorithm attains an FID score of 12.32 and an Inception Score of 8.14, which are competitive with the referred baseline methods. The quantitative results demonstrate the effectiveness of our algorithm.



(a) (b)

Figure 7. Random samples on CIFAR-10 ($\sigma = 1.0$, $\tau = 2.0$) and CelebA ($\sigma = 2.0$, $\tau = 8.0$).

5.5. Image Interpolation and Inpainting with Stage 2

To demonstrate usefulness of the proposed algorithm, we consider image interpolation and inpainting tasks.

Interpolating images linearly in the data distribution p_{data} would induce artifacts. However, if we perturb the linear interpolation using a Gaussian noise with variance σ^2, and then use our stage 2 to denoise, we are able to obtain an interpolation without such artifacts. We find $\sigma^2 = 0.4$ is suitable for the image interpolation task for CelebA. Fig. 8 lists the image interpolation results. Our algorithm produces smooth image interpolation by gradually changing facial attributes.

Figure 8. Image interpolation on CelebA. The first and last columns correspond to real images.

The second stage can also be utilized for image inpainting with a little modification, inspired by the image inpainting algorithm with annealed Langevin dynamics in (Song & Ermon, 2019). Let $\mathbf{m}$ be a mask with entries in $\{0, 1\}$ where 0 corresponds to missing pixels. The idea for inpainting is very similar to interpolation. We treat $\mathbf{x} \odot \mathbf{m} + \sigma\boldsymbol{\epsilon}$ as a sample from q_σ, where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Thus, we can use stage 2 to obtain samples from p_{data}. The image inpainting procedure is given in algorithm 2, and the results are presented in Fig. 9. Notice that we perturb $\mathbf{y}$ with $\sqrt{1 - \frac{k+1}{N_2}}\sigma\mathbf{z}$ at the end of each iteration. This is because the k-th iteration in stage 2 can be regarded as one-step Schrödinger Bridge from $q_{\sqrt{1-k/N_2}\sigma}$ to $q_{\sqrt{1-(k+1)/N_2}\sigma}$. Thus, the particles are supposed to follow $q_{\sqrt{1-(k+1)/N_2}\sigma}(\mathbf{x})$ after the k-th iteration.

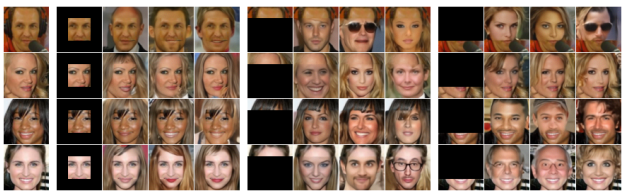


Figure 9. Image inpainting on CelebA. The leftmost column contains real images. Each occluded image is followed by three inpainting samples.

Algorithm 2 Inpainting with stage 2

Input: $\mathbf{y} = \mathbf{x} \odot \mathbf{m}, \mathbf{m}$
Sample $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
$\mathbf{x}_0 = \mathbf{y} + \sigma\mathbf{z}$
for $k = 0$ **to** $N_2 - 1$ **do**
  Sample $\boldsymbol{\epsilon}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathbf{b}(\mathbf{x}_k) = \mathbf{s}_\theta(\mathbf{x}_k, \sqrt{1 - \frac{k}{N_2}}\sigma)$
  $\mathbf{x}_{k+1} = \mathbf{x}_k + \frac{\sigma^2}{N_2}\mathbf{b}(\mathbf{x}_k) + \frac{\sigma}{\sqrt{N_2}}\boldsymbol{\epsilon}_n$
  $\mathbf{x}_{k+1} = \mathbf{x}_{k+1} \odot (1 - \mathbf{m}) + (\mathbf{y} + \sqrt{1 - \frac{k+1}{N_2}}\sigma\mathbf{z}) \odot \mathbf{m}$
end for
return $\mathbf{x}_{N_2}$

6. Conclusion

We propose to learn a generative model via entropy interpolation with a Schrödinger Bridge. At the population level, this entropy interpolation can be characterized via an SDE on $[0, 1]$ with a time varying drift term. We derive a two-stage Schrödinger Bridge algorithm by plugging the drift term estimated by a deep score estimator and a deep density estimator in the Euler-Maruyama method. Under some smoothness assumptions of the target distribution, we prove the consistency of the proposed Schrödinger Bridge approach, guaranteeing that the learned distribution converges to the target distribution. Experimental results on multimodal synthetic data and benchmark data support our theoretical findings and demonstrate that the generative model via Schrödinger Bridge is comparable with state-of-the-art GANs, suggesting a new formulation of generative learning.

7. Acknowledgement

We thank the reviewers for their valuable comments. This work is supported in part by the National Key Research and Development Program of China [grant 208AAA0101100], the National Science Foundation of China [grant 11871474], the research fund of KLATASDSMOE, Hong Kong Research Grant Council [grants 16307818, 16301419, 16308120], the Guangdong-Hong Kong-Macao Joint Laboratory [grant 2020B1212030001], Hong Kong Innovation and Technology Fund [PRP/029/19FX], Hong Kong University of Science and Technology (HKUST) [startup grant R9405, Z0428 from the Big Data Institute] and the HKUST-WeBank Joint Lab project. The computational task for this work was partially performed using the X-GPU cluster supported by the RGC Collaborative Research Fund: C6021-19EF.

References

Arbel, M., Sutherland, D., Bińkowski, M., and Gretton, A. On gradient regularizers for MMD GANs. In *Advances in Neural Information Processing Systems*, pp. 6701–6711, 2018.

Arbel, M., Korba, A., Salim, A., and Gretton, A. Maximum mean discrepancy gradient flow. In *Advances in Neural Information Processing Systems*, pp. 6481–6491, 2019.

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pp. 214–223, 2017.

Behrmann, J., Grathwohl, W., Chen, R. T. Q., Duvenaud, D., and Jacobsen, J.-H. Invertible residual networks. In *International Conference on Machine Learning*, pp. 573–582, 2019.

Bińkowski, M., Sutherland, D. J., Arbel, M., and Gretton, A. Demystifying MMD GANs. In *International Conference on Learning Representations*, 2018.

Block, A., Mroueh, Y., and Rakhlin, A. Generative modeling with denoising auto-encoders and langevin sampling. *arXiv preprint arXiv:2002.00107*, 2020.

Brock, A., Donahue, J., and Simonyan, K. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019.

Che, T., Li, Y., Jacob, A. P., Bengio, Y., and Li, W. Mode regularized generative adversarial networks. In *International Conference on Learning Representations*, 2017.

Chen, Y., Georgiou, T. T., and Pavon, M. Stochastic control liasons: Richard sinkhorn meets gaspard monge on a schroedinger bridge. *arXiv preprint arXiv:2005.10963*, 2020.

Choi, Y., Uh, Y., Yoo, J., and Ha, J.-W. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8188–8197, 2020.

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

Dinh, L., Krueger, D., and Bengio, Y. NICE: Non-linear independent components estimation. In *International Conference on Learning Representations*, 2015.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using Real NVP. In *International Conference on Learning Representations*, 2017.

Gao, Y., Jiao, Y., Wang, Y., Wang, Y., Yang, C., and Zhang, S. Deep generative learning via variational gradient flow. In *International Conference on Machine Learning*, pp. 2093–2101, 2019.

Gao, Y., Huang, J., Jiao, Y., Liu, J., Lu, X., and Yang, Z. Generative learning with euler particle transport. In *Annual Conference on Mathematical and Scientific Machine Learning*, volume 145, pp. 1–33, 2021.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pp. 2672–2680, 2014.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. Improved training of Wasserstein gans. In *Advances in Neural Information Processing Systems*, pp. 5769–5779, 2017.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pp. 6629–6640, 2017.

Higham, D. J. An algorithmic introduction to numerical simulation of stochastic differential equations. *SIAM review*, 43(3):525–546, 2001.

Jamison, B. The markov processes of schrödinger. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 32(4):323–331, 1975.

Johnson, R. and Zhang, T. Composite functional gradient learning of generative adversarial models. In *International Conference on Machine Learning*, pp. 2371–2379, 2018.

Karras, T., Aila, T., Laine, S., and Lehtinen, J. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018.

Karras, T., Laine, S., and Aila, T. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4401–4410, 2019.

Kingma, D. P. and Dhariwal, P. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, pp. 10236–10245, 2018.

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

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Léonard, C. A survey of the schrodinger problem and some of its connections with optimal transport. *DYNAMICAL SYSTEMS*, 34(4):1533–1574, 2014.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision*, pp. 3730–3738, 2015.

Liutkus, A., Simsekli, U., Majewski, S., Durmus, A., Stöter, F.-R., Chaudhuri, K., and Salakhutdinov, R. Sliced-Wasserstein flows: Nonparametric generative modeling via optimal transport and diffusions. In *International Conference on Machine Learning*, pp. 4104–4113, 2019.

Makhzani, A., Shlens, J., Jaitly, N., and Goodfellow, I. Adversarial autoencoders. In *ICLR Workshop*, 2016.

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018.

Nowozin, S., Cseke, B., and Tomioka, R. f-GAN: Training generative neural samplers using variational divergence minimization. In *Advances in Neural Information Processing Systems*, pp. 271–279, 2016.

Papamakarios, G., Pavlakou, T., and Murray, I. Masked autoregressive flow for density estimation. In *Advances in Neural Information Processing Systems*, pp. 2335–2344, 2017.

Prenger, R., Valle, R., and Catanzaro, B. Waveglow: A flow-based generative network for speech synthesis. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3617–3621, 2019.

Radford, A., Metz, L., and Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Learning Representations*, 2016.

Razavi, A., van den Oord, A., and Vinyals, O. Generating diverse high-fidelity images with vq-vae-2. In *Advances in Neural Information Processing Systems*, pp. 14837–14847, 2019.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pp. 2226–2234, 2016.

Schrödinger, E. Sur la théorie relativiste de l'électron et l'interprétation de la mécanique quantique. In *Annales de l'institut Henri Poincaré*, volume 2, pp. 269–310, 1932.

Shen, Y., Gu, J., Tang, X., and Zhou, B. Interpreting the latent space of gans for semantic face editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9243–9252, 2020.

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, pp. 11895–11907, 2019.

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

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

Tolstikhin, I., Bousquet, O., Gelly, S., and Schölkopf, B. Wasserstein auto-encoders. In *International Conference on Learning Representations*, 2018.

Van Den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A. W., and Kavukcuoglu, K. WaveNet: A generative model for raw audio. In *9th ISCA Speech Synthesis Workshop*, pp. 125–125, 2016.

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

Zhang, H., Goodfellow, I., Metaxas, D., and Odena, A. Self-attention generative adversarial networks. In *International Conference on Machine Learning*, pp. 7354–7363, 2019.

Zhu, J.-Y., Krähenbühl, P., Shechtman, E., and Efros, A. A. Generative visual manipulation on the natural image manifold. In *European Conference on Computer Vision*, pp. 597–613, 2016.

Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2223–2232, 2017.