LIKELIHOOD TRAINING OF SCHRÖDINGER BRIDGE USING FORWARD-BACKWARD SDEs THEORY

Anonymous authors
Paper under double-blind review

ABSTRACT

Schrödinger Bridge (SB) is an optimal transport problem that has received increasing attention in deep generative modeling for its mathematical flexibility compared to the Scored-based Generative Model (SGM). However, it remains unclear whether the optimization principle of SB relates to the modern training of deep generative models, which often rely on constructing parameterized log-likelihood objectives.This raises questions on the suitability of SB models as a principled alternative for generative applications. In this work, we present a novel computational framework for likelihood training of SB models grounded on *Forward-Backward Stochastic Differential Equations Theory* – a mathematical methodology appeared in stochastic optimal control that transforms the optimality condition of SB into a set of SDEs. Crucially, these SDEs can be used to construct the likelihood objectives for SB that, surprisingly, generalizes the ones for SGM as special cases. This leads to a new optimization principle that inherits the same SB optimality yet without losing applications of modern generative training techniques, and we show that the resulting training algorithm achieves comparable results on generating realistic images on MNIST, CelebA, and CIFAR10.

1 INTRODUCTION

Score-based Generative Model (SGM; Song et al. (2020b)) is an emerging class of generative models that has achieved remarkable results in synthesizing high-fidelity data, *e.g.* images (Dhariwal & Nichol, 2021; Song & Ermon, 2020), audio (Kong et al., 2020a;b), and geometry structures (Luo & Hu, 2021). Like many deep generative models, SGM seeks to learn nonlinear functions that transform simple distributions (typically Gaussian) into complex, often intractable, data distributions. In SGM, this is done by first diffusing data to noise through a stochastic differential equation (SDE); then learning to *reverse* this diffusion process by regressing a network to match the score function (*i.e.* the gradient of the log probability density) at each time step (Hyvärinen & Dayan, 2005; Song et al., 2020a). This reversed process thereby defines the generative mechanism (see Fig. 1).

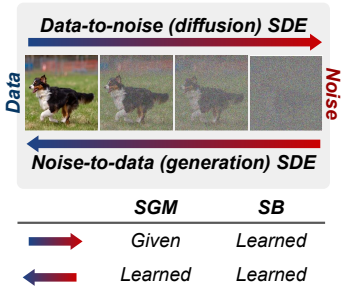

Figure 1: Both Score-based Generative Model (SGM) and Schrödinger Bridge (SB) involve stochastic transformation between two distributions. While SGM requires the data-to-noise diffusion process to be pre-specified, SB instead *learns* the process jointly.

Despite its empirical successes, SGM admits few limitations. First, the diffusion process has to obey a simple form (*e.g.* linear or degenerate drift) in order to compute the analytic score function for the regression purpose. Secondly, the diffusion process needs to run to sufficiently large time steps so that the end distribution is approximate Gaussian (Kong & Ping, 2021). For these reasons, SGM often takes a notoriously long time in generating data (Jolicoeur-Martineau et al., 2021), thereby limiting their practical usages compared to *e.g.* GANs or flow-based models (Ping et al., 2020; Karras et al., 2020b).

In the attempt to lift these restrictions, a line of recent works inspired by Schrödinger Bridge (SB; Schrödinger (1932)) has been proposed (De Bortoli et al., 2021; Wang et al., 2021; Vargas et al., 2021). SB – as an optimal transport problem – seeks a set of optimal policies that transforms back-and-forth between two *arbitrary* distributions in a *finite* horizon. The similarity between the two problems

Table 1: Comparison of related methods. Our method enjoys the *same* SB-inherited flexibility on designing diffusion SDEs while retaining application of SGM training techniques.

		SB-based diffusion SDEs		Connection to SGM training	
		nonlinear drift	fewer time steps	log-likelihood	Langevin sampling
SGM		✗	✗	-	-
SB	prior attempts	✓	✓	✗	✗
	our work	✓	✓	✓	✓

(*i.e.* both involve transforming distributions) is evident, and the additional flexibility from SB is also attractive. To enable SB-inspired generative training, however, previous works require either ad-hoc multi-stage optimization or adopting traditional SB algorithms (Ruschendorf, 1995; Kullback, 1968). The underlying relation between the optimization principle of SB and modern generative training, in particular SGM, remains relatively unexplored, despite their intimately related problem formulations. More importantly, with the recent connection between SGM and log-likelihood estimation (Song et al., 2021; Huang et al., 2021), it is crucial to explore whether there exists an alternative way of training SB that better respects, or perhaps generalizes, modern training of SGM, so as to solidify the suitability of SB as a principled generative model.

In this work, we present a fundamental connection between solving SB and training SGM. The difficulty arises immediately as one notices that the optimality condition of SB and the likelihood objective of SGM are represented by merely two distinct mathematical objects. While the former is characterized by two coupled partial differential equations (PDEs) (Léonard, 2013), the latter integrates over a notably complex SDE that resembles neither its diffusion nor reversed process (Song et al., 2021). Nevertheless, inspired by the recent advance on understanding deep learning through the optimal control perspective (Li & Hao, 2018; Liu et al., 2021a;b), we show that *Forward-Backward SDEs* – a mathematical methodology appeared in stochastic optimal control for solving high-dimensional nonlinear PDEs (Han et al., 2018) – paves an elegant way to connect the two objectives. The implication of our findings is nontrivial: It yields a novel computational framework for SB to compute parameterized log-likelihood objectives that, surprisingly, generalize the ones for SGM. In other words, it is possible to retain the mathematical principle from SB without losing connection to the modern SGM training, *e.g.* likelihood computation (Song et al., 2021) and Langevin sampling (Song & Ermon, 2019). We show that the resulting method generates comparable images on MNIST, CelebA, and CIFAR10 and outperforms prior optimal transport models by a large margin.

In summary, we present the following contributions.

- We present a novel computational framework, grounded on *Forward-Backward SDEs* theory, for computing the log-likelihood objectives of Schrödinger Bridge (SB) and solidifying their theoretical connections to Score-based Generative Model (SGM).
- Our framework suggests a new training principle that retains the mathematical flexibility from SB while enjoying advanced techniques from the modern generative training of SGM (see Table 1).
- We show that the resulting method outperforms previous optimal transport-inspired baselines on synthesizing high-fidelity images that are comparable to those generated by other existing models.

2 Preliminaries

Notation. We denote $p_t^{\mathrm{SDE}}(\mathbf{x}_t)$ as the marginal density driven by some SDE process until the time step $t \in [0, T]$. The time direction is aligned throughout this article such that p_0 and p_T respectively correspond to the data and prior distributions.

2.1 Score-based Generative Model (SGM)

Given a data point $\mathbf{x}_0 \in \mathbb{R}^n$ sampled from an unknown data distribution p_{data}, SGM first progressively diffuses the data towards random noise with the following forward SDE:

$$\mathrm{d}\mathbf{x}_t = f(t, \mathbf{x}_t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}_t, \quad \mathbf{x}_0 \sim p_{\text{data}}, \tag{1}$$

where $f(\cdot, t) : \mathbb{R}^n \to \mathbb{R}^n$, $g(t) \in \mathbb{R}$, and $\mathbf{w}_t \in \mathbb{R}^n$ are respectively the drift, diffusion, and standard Wiener process. Typically, $g(\cdot)$ is some monotonically increasing function such that for sufficiently large time steps, we have $p_T^{(1)} \approx p_{\text{prior}}$ resemble some prior distribution (*e.g.* Gaussian) at the terminal horizon T. It is known that reversing (1) yields another backward SDE[1] that traverses backward in time (Anderson, 1982):

$$\mathrm{d}\mathbf{x}_t = [f - g^2 \, \nabla_{\mathbf{x}} \log p_t^{(1)}(\mathbf{x}_t)]\mathrm{d}t + g \, \mathrm{d}\mathbf{w}_t, \quad \mathbf{x}_T \sim p_T^{(1)}, \tag{2}$$

where $\nabla_{\mathbf{x}} \log p_t^{(1)}$ is the gradient of the log density induced by the SDE (1), often known as the *score* function. These two stochastic processes are equivalent in the sense that their marginal densities are equal to each other throughout $t \in [0, T]$; in other words, $p_t^{(1)} \equiv p_t^{(2)}$.

When the drift f is of simple structure, for instance linear (Ho et al., 2020) or simply degenerate (Song & Ermon, 2019), the conditional score function $\nabla_{\mathbf{x}} \log p_t^{(1)}(\mathbf{x}_t|\mathbf{x}_0) \equiv \nabla_{\mathbf{x}} \log p_{\mathbf{x}_t|\mathbf{x}_0}$ admits an analytic solution at any time t. Hence, SGM proposes to train a parameterized score network $\boldsymbol{s}(t, \mathbf{x}_t; \theta) \equiv \boldsymbol{s}_\theta$ by regressing its outputs to the ground-truth values, *i.e.* $\mathbb{E}[\lambda(t) \left\| \boldsymbol{s}_\theta - \nabla_{\mathbf{x}} \log p_{\mathbf{x}_t|\mathbf{x}_0} \right\|^2]$, where $\lambda(t)$ is some hand-designed weighting function. In practice, the choice of $\lambda(t)$ can largely affect the performance of SGM. Fortunately, recent works (Song et al., 2021; Huang et al., 2021) have shown that the log-likelihood of SGM, despite being complex, can be parameterized as follows:

$$\begin{aligned}\mathcal{L}_{\text{SGM}}(\mathbf{x}_0; \theta) &= \mathbb{E}\left[\log p_T(\mathbf{x}_T)\right] - \int_0^T \mathbb{E}\left[\frac{1}{2} g^2 \left\|\boldsymbol{s}_\theta\right\|^2 + \nabla_{\mathbf{x}} \cdot \left(g^2 \boldsymbol{s}_\theta - f\right)\right] \mathrm{d}t, \\ &= \mathbb{E}\left[\log p_T(\mathbf{x}_T)\right] - \int_0^T \mathbb{E}\left[\frac{1}{2} g^2 \left\|\boldsymbol{s}_\theta - \nabla_{\mathbf{x}} \log p_{\mathbf{x}_t|\mathbf{x}_0}\right\|^2 - \frac{1}{2} \left\|g \nabla_{\mathbf{x}} \log p_{\mathbf{x}_t|\mathbf{x}_0}\right\|^2 - \nabla_{\mathbf{x}} \cdot f\right] \mathrm{d}t\end{aligned} \tag{3}$$

where $\nabla_{\mathbf{x}} \cdot$ denotes the divergence w.r.t. $\mathbf{x}$. The objective in (3) justifies the regression (*i.e.* denoising score-matching; Vincent (2011)) training and suggests a principled choice of $\lambda(t) := g(t)^2$. After training, SGM simply substitutes the score function with $\boldsymbol{s}_\theta$ to generate data from p_{prior}, *i.e.*

$$\mathrm{d}\mathbf{x}_t = [f - g^2 \, \boldsymbol{s}_\theta]\mathrm{d}t + g \, \mathrm{d}\mathbf{w}_t, \quad \mathbf{x}_T \sim p_{\text{prior}}. \tag{4}$$

It is important to notice that p_{prior} needs *not* equal $p_T^{(1)}$ in practice, and the approximation is close only through a careful design of (1). In practice, designing the diffusion $g(t)$ can be particularly problematic, as it affects both the approximation $p_T^{(1)} \approx p_{\text{prior}}$ and the training via the weighting $\lambda(t)$; hence can easily lead to unstable training (Song et al., 2021; Nichol & Dhariwal, 2021). This is in contrast to Schrödinger Bridge, which considers a more flexible framework that requires minimal manipulation.

2.2 Schrödinger Bridge (SB)

Following the dynamic expression of SB (Pavon & Wakolbinger, 1991; Dai Pra, 1991), consider

$$\min_{\mathbb{Q} \in \mathcal{P}(p_{\text{data}}, p_{\text{prior}})} D_{\text{KL}}(\mathbb{Q} \,||\, \mathbb{P}), \tag{5}$$

where $\mathbb{Q} \in \mathcal{P}(p_{\text{data}}, p_{\text{prior}})$ belongs to a set of path measure with p_{data} and p_{prior} as its marginal densities at $t = 0$ and T. On the other hand, $\mathbb{P}$ denotes a reference measure, which we will set to the path measure of (1) for later convenience. The optimality condition to (5) is characterized by two PDEs that are coupled through their boundary conditions. We summarize the related result below.

Theorem 1 (SB optimality; Léonard (2013); Caluya & Halder (2021))**.** *Let $\Psi(t, \mathbf{x}_t)$ and $\hat{\Psi}(t, \mathbf{x}_t)$ be the solutions to the following PDEs, which can be obtained through the Hopf-Cole transform (Léger & Li, 2021):*

$$\begin{cases} \frac{\partial \Psi}{\partial t} = -\Psi_{\mathbf{x}}^{\mathsf{T}} f - \frac{1}{2} \operatorname{Tr}(g^2 \Psi_{\mathbf{xx}}) \\ \frac{\partial \hat{\Psi}}{\partial t} = -\nabla_{\mathbf{x}} \cdot (\hat{\Psi} f) + \frac{1}{2} \operatorname{Tr}(g^2 \hat{\Psi}_{\mathbf{xx}}) \end{cases} \quad s.t. \ \Psi(0, \cdot)\hat{\Psi}(0, \cdot) = p_{\text{data}}, \ \Psi(T, \cdot)\hat{\Psi}(T, \cdot) = p_{\text{prior}}, \tag{6}$$

where $\Psi_{\mathbf{x}} \equiv \frac{\partial \Psi}{\partial \mathbf{x}}$, $\Psi_{\mathbf{xx}} \equiv \frac{\partial^2 \Psi}{\partial \mathbf{x} \partial \mathbf{x}}$, and etc. Then, the solution to the optimization (5) can be expressed

[1] Hereafter, we will sometimes drop $f \equiv f(t, \mathbf{x}_t)$ and $g \equiv g(t)$ for brevity.

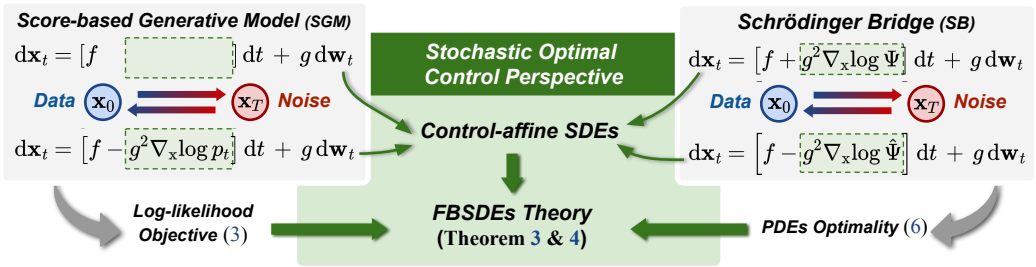

Figure 2: Schematic diagram of the our stochastic optimal control interpretation, and how it connects the objective of SGM (3) and optimality of SB (6) through Forward-Backward SDEs theory.

by the path measure of the following forward (7a), or equivalently backward (7b), SDE:

$$\mathrm{d}\mathbf{x}_t = [f + g^2\,\nabla_{\mathbf{x}}\log\Psi(t,\mathbf{x}_t)]\mathrm{d}t + g\,\mathrm{d}\mathbf{w}_t, \quad \mathbf{x}_0 \sim p_{\text{data}}, \tag{7a}$$

$$\mathrm{d}\mathbf{x}_t = [f - g^2\,\nabla_{\mathbf{x}}\log\hat{\Psi}(t,\mathbf{x}_t)]\mathrm{d}t + g\,\mathrm{d}\mathbf{w}_t, \quad \mathbf{x}_T \sim p_{\text{prior}}, \tag{7b}$$

where $\nabla_{\mathbf{x}}\log\Psi(t,\mathbf{x}_t)$ and $\nabla_{\mathbf{x}}\log\hat{\Psi}(t,\mathbf{x}_t)$ are the optimal forward and backward drifts for SB.

Similar to the forward/backward processes in SGM, the stochastic processes of SB in (7a) and (7b) are also equivalent in the sense that $\forall t \in [0,T]$, $p_t^{(7a)} \equiv p_t^{(7b)} \equiv p_t^{\text{SB}}$. In fact, its marginal density obeys a factorization principle: $p_t^{\text{SB}}(\mathbf{x}_t) = \Psi(t,\mathbf{x}_t)\hat{\Psi}(t,\mathbf{x}_t)$.

To construct the generative pipeline from (7b), one requires solving the PDEs in (6) to obtain $\hat{\Psi}$. Unfortunately, these PDEs are hard to solve even for low-dimensional systems (Renardy & Rogers, 2006); let alone for generative applications. Indeed, previous works either have to replace the original Schrödinger Bridge ($p_{\text{data}} \leftrightarrows p_{\text{prior}}$) with multiple stages, $p_{\text{data}} \leftrightarrows p_{\text{middle}} \leftrightarrows p_{\text{prior}}$, so that each segment admits an analytic solution (Wang et al., 2021), or consider the following half-bridge ($p_{\text{data}} \leftarrow p_{\text{prior}}$ *vs.* $p_{\text{data}} \rightarrow p_{\text{prior}}$) optimization (De Bortoli et al., 2021; Vargas et al., 2021),

$$\mathbb{Q}^{(1)} := \underset{\mathbb{Q}\in\mathcal{P}(\cdot,p_{\text{prior}})}{\arg\min}\; D_{\text{KL}}(\mathbb{Q}\,||\,\mathbb{Q}^{(0)}), \quad \mathbb{Q}^{(0)} := \underset{\mathbb{Q}\in\mathcal{P}(p_{\text{data}},\cdot)}{\arg\min}\; D_{\text{KL}}(\mathbb{Q}\,||\,\mathbb{Q}^{(1)})$$

which can be solved with traditional SB methods such as Iterative Proportional Fitting (Ruschendorf (1995); Kullback (1968)) starting from $\mathbb{Q}^{(0)} := \mathbb{P}$. In the following section, we will present a scalable computational framework for solving the optimality PDEs in (6) and show that it paves an elegant way connecting the optimality principle of SB (6) to the parameterized log-likelihood of SGM (3).

3 Approach

We motivate our approach starting from some control-theoretic observation (see Fig. 2). Notice that both SGM and SB consist of forward and backward SDEs with a surprisingly similar structure. From the stochastic control perspective, these SDEs belong to the class of *control-affine* SDEs with time-dependent additive noise:

$$\mathrm{d}\mathbf{x}_t = \boldsymbol{A}(t,\mathbf{x}_t)\mathrm{d}t + \boldsymbol{B}(t,\mathbf{x}_t)\mathbf{u}(t,\mathbf{x}_t)\mathrm{d}t + \boldsymbol{C}(t)\,\mathrm{d}\mathbf{w}_t. \tag{8}$$

It is clear that the control-affine SDE (8) includes all SDEs (1,2,4,7) appeared in Section 2 by considering $(\boldsymbol{A},\boldsymbol{B},\boldsymbol{C}) := (f,\boldsymbol{I},g)$ and different interpretations of the *control* variables $\mathbf{u}(t,\mathbf{x}_t)$. This implies that we should be able to align the optimization processes of both SGM and SB through the lens of *stochastic optimal control* (SOC). Indeed, both problems can be interpreted as seeking some time-varying control policy, either the score function $\nabla_{\mathbf{x}}\log p_{\mathbf{x}_t|\mathbf{x}_0}$ in SGM or $\nabla_{\mathbf{x}}\log\hat{\Psi}$ in SB, that minimizes some objectives, (3) *vs.* (5), while subjected to some control-affine SDEs, (1,2) *vs.* (7). Our main contribution is then to identify a specific mathematical methodology appeared in nonlinear SOC literature – called *Forward-Backward SDEs* theory (FBSDEs; see Ma et al. (1999)) – that links the optimality condition of SB (6) to the log-likelihood objectives of SGM (3). We summarize these findings in Theorem 3 and 4, and provide connections to other generative models, *i.e.* flow-based models, in Corollary 5. All proofs are left to Appendix A.

3.1 FORWARD-BACKWARD SDES (FBSDES) REPRESENTATION FOR SB

The theory of FBSDEs establishes an innate connection between different classes of PDEs and forward-backward SDEs. Below we introduce one of which that is particularly related to our problem.

Lemma 2 (Nonlinear Feynman-Kac;[2] Exarchos & Theodorou (2018))**.** *Consider the Cauchy problem*

$$v_t + \frac{1}{2}\operatorname{Tr}(v_{xx}G(t,x)G^{\mathsf{T}}(t,x)) + v_x^{\mathsf{T}} f(t,x) + h(t,x,v,G^{\mathsf{T}}(t,x)v_x) = 0, \quad v(T,x) = \varphi(x), \tag{9}$$

wherein the functions G, f, h, and φ satisfy mild regularity conditions (c.f. Appendix A.5). Then the PDE in (9) admits a unique (viscosity) solution with the stochastic representation:

$$v(t,x) = y(t,x) \qquad \textit{and} \qquad G(t,x)^{\mathsf{T}} v_x(t,x) = z(t,x), \tag{10}$$

where $x(t)$, $y(t,x)$, and $z(t,x)$ are the unique solutions to the following "forward-backward SDEs":

$$\begin{cases} \mathrm{d}x = f(t,x)\mathrm{d}t + G(t,x)\mathrm{d}w_t \\ \mathrm{d}y = -h(x,y,z,t)\mathrm{d}t + z^{\mathsf{T}}\mathrm{d}w_t \end{cases} \qquad \textit{s.t. } x(0) = x_0, \quad y(T,x) = \phi(x), \tag{11}$$

Lemma 2 states that solutions to a certain class of nonlinear (via the function h in (9)) PDEs can be represented by solutions to a set of forward-backward SDEs (11) through the transformation (10). From a statistical physics viewpoint, it shares a similar spirit to how Fokker-Plank PDE characterizes the density evolution of an SDE, by providing conversion between the *macroscopic* dynamics (in terms of PDEs) and their underlying *microscopic* fluctuations (in terms of forward-backward SDEs).

Since it is often computationally favorable to solve SDEs rather than PDEs, Lemma 2 has been widely used as a scalable method for solving high-dimensional PDEs (Han et al., 2018; Pereira et al., 2019). Take SOC applications for instance, their PDE optimality condition can be characterized by (10) under proper conditions, and the associated optimal control is given by an affine mapping from $z(t,x)$. Hence, one can adopt Lemma 2 to solve the underlying FBSDEs, rather than the original PDE optimality, for the optimal control. Despite seemingly attractive, whether these principles can be extended to SB, whose optimality conditions are given by *two coupled PDEs* in (6), remains unclear. Below we derive a similar FBSDEs representation for SB.

Theorem 3 (FBSDEs to SB Optimality (6))**.** *With the same regularity conditions in Lemma 2, the solutions to the following forward-backward SDEs provide a stochastic representation to (6).*

$$\begin{cases} \mathrm{d}\mathbf{x}_t = (f + g\mathbf{z}_t)\,\mathrm{d}t + g\mathrm{d}\mathbf{w}_t & \text{(12a)} \\ \mathrm{d}\mathbf{y}_t = \frac{1}{2}\mathbf{z}_t^{\mathsf{T}}\mathbf{z}_t\mathrm{d}t + \mathbf{z}_t^{\mathsf{T}}\mathrm{d}\mathbf{w}_t & \text{(12b)} \\ \mathrm{d}\hat{\mathbf{y}}_t = \left(\frac{1}{2}\hat{\mathbf{z}}_t^{\mathsf{T}}\hat{\mathbf{z}}_t + \nabla\cdot(g\hat{\mathbf{z}}_t - f) + \hat{\mathbf{z}}_t^{\mathsf{T}}\mathbf{z}_t\right)\mathrm{d}t + \hat{\mathbf{z}}_t^{\mathsf{T}}\mathrm{d}\mathbf{w}_t & \text{(12c)} \end{cases}$$

where the boundary conditions are given by $\mathbf{x}(0) = \mathbf{x}_0$ and $\mathbf{y}_T + \hat{\mathbf{y}}_T = \log p_{\text{prior}}(\mathbf{x}_T)$. Specifically, the solutions to (12) relate to the SB optimality PDE (6) by

$$\begin{aligned} \mathbf{y}_t &\equiv \mathbf{y}(t,\mathbf{x}_t) = \log\Psi(t,\mathbf{x}_t), \qquad & \mathbf{z}_t &\equiv \mathbf{z}(t,\mathbf{x}_t) = g\nabla_{\mathbf{x}}\log\Psi(t,\mathbf{x}_t), \\ \hat{\mathbf{y}}_t &\equiv \hat{\mathbf{y}}(t,\mathbf{x}_t) = \log\hat{\Psi}(t,\mathbf{x}_t), \qquad & \hat{\mathbf{z}}_t &\equiv \hat{\mathbf{z}}(t,\mathbf{x}_t) = g\nabla_{\mathbf{x}}\log\hat{\Psi}(t,\mathbf{x}_t). \end{aligned} \tag{13}$$

Furthermore, $(\mathbf{y}_t, \hat{\mathbf{y}}_t)$ obey the following relation:

$$\mathbf{y}_t + \hat{\mathbf{y}}_t = \log p_t^{\text{SB}}(\mathbf{x}_t).$$

The FBSDEs for SB (12) share a similar forward-backward structure as in (11), where (12a) and (12b,12c) respectively represent the forward and backward SDEs. One can verify that the forward SDE (12a) coincides with the *optimal* forward SDE (7a) with the substitution $\mathbf{z}_t = g\nabla_{\mathbf{x}}\log\Psi$. In other words, these FBSDEs provide a *local* representation of $\log\Psi$ and $\log\hat{\Psi}$ evaluated on the optimal path governed by (7a). Finally, comparing (12a) to (8) implies that we can interpret $\mathbf{z}_t$ as the *control* for the forward diffusion process; similarly, $\hat{\mathbf{z}}_t$ can be viewed as the one for the backward reversed process. With these interpretations in mind, solving the original SB problem is mathematically equivalent to finding the optimal control policies $\mathbf{z}_t$ and $\hat{\mathbf{z}}_t$ that solve the FBSDEs in (12).

[2] Lemma 2 can be viewed as the nonlinear extension of the celebrated Feynman-Kac formula (Karatzas & Shreve, 2012), which characterizes the connection between linear PDEs and forward SDEs.

3.2 LOG-LIKELIHOOD COMPUTATION OF SB

Theorem 3 has an important implication: It suggests that given a path sampled from the forward SDE (12a), the solutions to the backward SDEs (12b,12c) at $t = 0$ can be used to construct an unbiased estimation of the log-likelihood of the data point $\mathbf{x}_0$, *i.e.* $\mathbb{E}\left[\mathbf{y}_0 + \hat{\mathbf{y}}_0\right] = \log p_0^{\text{SB}}(\mathbf{x}_0) = \log p_{\text{data}}(\mathbf{x}_0)$. We now state our main result, which makes this observation formal:

Theorem 4 (Parameterized log-likelihood of SB)**.** *Suppose $\mathbf{z}_t \approx \mathbf{z}_\theta(t, \mathbf{x}_t)$ and $\hat{\mathbf{z}}_t \approx \hat{\mathbf{z}}_\phi(t, \mathbf{x}_t)$ are respectively parameterized by θ and ϕ, then the log-likelihood of SB can be parameterized as:*

$$\mathcal{L}_{\text{SB}}(\mathbf{x}_0; \theta, \phi) = \mathbb{E}\left[\log p_T(\mathbf{x}_T)\right] - \int_0^T \mathbb{E}\left[\frac{1}{2}\left\|\mathbf{z}_t\right\|^2 + \frac{1}{2}\left\|\hat{\mathbf{z}}_t\right\|^2 + \nabla \cdot (g\hat{\mathbf{z}}_t - f) + \hat{\mathbf{z}}_t^{\mathsf{T}}\mathbf{z}_t\right] \mathrm{d}t \tag{14}$$

$$= \mathbb{E}\left[\log p_T(\mathbf{x}_T)\right] - \int_0^T \mathbb{E}\left[\frac{1}{2}\left\|\mathbf{z}_t\right\|^2 + \frac{1}{2}\left\|\hat{\mathbf{z}}_t - g\nabla \log p_t^{\text{SB}} + \mathbf{z}_t\right\|^2 - \frac{1}{2}\left\|g\nabla \log p_t^{\text{SB}} - \mathbf{z}_t\right\|^2 - \nabla \cdot f\right] \mathrm{d}t,$$

where the expectation is taken over the path measure induced by the forward SDE (12a).

With Theorem 4, we now have a more compact picture: Let us recall Fig. 2 again and compare the parameterized log-likelihoods of SB (14) and SGM (3); one can verify that $\mathcal{L}_{\text{SB}}$ collapses to $\mathcal{L}_{\text{SGM}}$ when $(\mathbf{z}_t, \hat{\mathbf{z}}_t) := (\mathbf{0}, g\, s_\theta)$. From the SB perspective, this occurs only when $p_T^{(1)} = p_{\text{prior}}$. Since no effort is required in the forward process to reach p_{prior}, the optimal forward control $\mathbf{z}_t$, by definition, degenerates; thereby making the backward control $\hat{\mathbf{z}}_t$ collapses to the score function (recall that (7a) and (7b) share the same marginal density). However, in any case when $p_T^{(1)} \neq p_{\text{prior}}$, for instance when the diffusion SDEs are improperly designed, the forward policy $\mathbf{z}_t$ steers the diffusion process back to p_{prior}, while its backward counterpart $\hat{\mathbf{z}}_t$ compensates the reversed process accordingly. From this view, SB alleviates the problematic design in SGM by enlarging the class of diffusion processes to accept *nonlinear* drifts and providing an optimization principle on learning these processes. Moreover, our finding in Theorem 4 generalizes the log-likelihood training from SGM to SB.

In practice, we parameterize the forward and backward controls, $\mathbf{z}_\theta$ and $\hat{\mathbf{z}}_\phi$, by two distinct DNNs. As our SB training approach is based heavily on the FBSDEs theory, we name our model **SB-FBSDE**.

Connection to flow-based models. Interestingly, the log-likelihood computation in Theorem 4, where we use a path $\{\mathbf{x}_t\}_{t\in[0,T]}$ sampled from a data point $\mathbf{x}_0$ to parameterize its log-likelihood, resembles modern training of (deterministic) flow-based models (Grathwohl et al., 2018), which have recently been shown to admit a close relation to SGM (Song et al., 2020b; Gong & Li, 2021). The connection is built on the concept of *probability flow* – which suggests that the marginal density of an SDE can be evaluated through an ordinary differential equation (ODE). Below, we provide a similar flow representation for SB, further strengthening their connection to modern generative models.

Corollary 5 (Probability flow for SB)**.** *The following ODE characterizes the probability flow of the optimal processes of SB (7) in the sense that $\forall t,\ p_t^{(15)} \equiv p_t^{(7)} \equiv p_t^{\text{SB}}$.*

$$\mathrm{d}\mathbf{x}_t = \left[f + g\mathbf{z}_t - \frac{1}{2}g(\mathbf{z}_t + \hat{\mathbf{z}}_t)\right] \mathrm{d}t + g\, \mathrm{d}\mathbf{w}_t, \tag{15}$$

One can verify (see Appendix A.5) that computing the log-likelihood of this ODE model (15) using flow-based training techniques indeed recovers the training objective of SB derived in (14).

Practical training. So far we have established a solid connection showing that optimizing $\hat{\mathbf{z}}_\phi$ with $\mathcal{L}_{\text{SB}}$ (14) resembles the training process of the score network $\boldsymbol{s}$ in SGM. While (14) may also be used to jointly optimize $\mathbf{z}_\theta$, it requires keeping the entire computational graph, which can be probability expensive on high-dimensional datasets.[3] Rather, an alternative training scheme that better leverages the symmetric structure of SB is to re-derive the entire computational framework (*i.e.* Theorem 3 and 4) for optimizing $\mathbf{z}_\theta$ based on the *backward* state-trajectories sampled from (7b). Crucially, re-basing the sampled path $\{\mathbf{x}_t\}_{t\in[0,T]}$ from (7a) to (7b) yields a new parameterized log-likelihood for the noise $\mathcal{L}_{\text{SB}}(\mathbf{x}_T; \theta, \phi)$. Hence, in practice we alternate the training between $\min_\phi \mathcal{L}_{\text{SB}}(\mathbf{x}_0)$ and $\min_\theta \mathcal{L}_{\text{SB}}(\mathbf{x}_T)$. Our log-likelihood training of SB is summarized in Alg. 1, and hereafter we refer to

[3] Due to space constraint, we leave the computational discussion of SB training, as well as the related result for computing $\mathcal{L}_{\text{SB}}(\mathbf{x}_T; \theta, \phi)$, to Appendices A.4 and B.

Algorithm 1 Likelihood Training of SB-FBSDE

Input: boundary distributions p_{data} and p_{prior}, learning rate η, parameterized $\mathbf{z}_\theta(\cdot,\cdot)$ and $\hat{\mathbf{z}}_\phi(\cdot,\cdot)$
repeat
  for $k = 1$ **to** K **do**
    Sample $\mathbf{x}_{t\in[0,T]}$ from (12a) where $\mathbf{x}_0 \sim p_{\text{data}}$.
    Compute $\mathcal{L}_{\text{SB}}(\mathbf{x}_0;\theta,\phi)$ with (14).
    Update $\phi \leftarrow \phi + \eta\nabla_\phi\mathcal{L}_{\text{SB}}$.
  end for
  for $k = 1$ **to** K **do**
    Sample $\mathbf{x}_{t\in[0,T]}$ from (23a) where $\mathbf{x}_T \sim p_{\text{prior}}$.
    Compute $\mathcal{L}_{\text{SB}}(\mathbf{x}_T;\theta,\phi)$ with (24).
    Update $\theta \leftarrow \theta + \eta\nabla_\theta\mathcal{L}_{\text{SB}}$.
  end for
until converges

Algorithm 2 Generative Process of SB-FBSDE

Input: p_{prior}, parameterized $\mathbf{z}_\theta(\cdot,\cdot)$ and $\hat{\mathbf{z}}_\phi(\cdot,\cdot)$, noise scales for Langevin corrector $\{\sigma_i\}$
Sample $\mathbf{x}_T \sim p_{\text{prior}}$.
for $t = T$ **to** Δt **do**
  Sample $\epsilon \sim \mathcal{N}(\mathbf{0},\boldsymbol{I})$.
  Predict $\mathbf{x}_t \leftarrow \mathbf{x}_t + g\,\hat{\mathbf{z}}_\phi(t,\mathbf{x}_t)\Delta t + \sqrt{g\Delta t}\,\epsilon$.
  for $i = 1$ **to** N **do**
    Sample $\epsilon \sim \mathcal{N}(\mathbf{0},\boldsymbol{I})$.
    Compute $g\nabla_{\mathbf{x}}\log p_t^{\text{SB}} = \mathbf{z}_\theta(t,\mathbf{x}_t)+\hat{\mathbf{z}}_\phi(t,\mathbf{x}_t)$.
    Correct $\mathbf{x}_t \leftarrow \mathbf{x}_t + \sigma_i\nabla_{\mathbf{x}}\log p_t^{\text{SB}} + \sqrt{2\sigma_i}\,\epsilon$.
  end for
  Propagate $\mathbf{x}_{t-\Delta t} \leftarrow \mathbf{x}_t$.
end for
return $\mathbf{x}_0$

each cycle, *i.e.* $2K$ training steps, as a *training stage* of SB-FBSDE. This distinct training scheme substantially accelerates the convergence under proper initialization. Specifically, when $\mathbf{z}_\theta$ is initialized with degenerate outputs (*e.g.* by zeroing out its last layer), training $\mathbf{z}_\phi$ at the first stage can be made in a similar SGM fashion since p_t^{SB} admits analytical expression. As for the proceeding stages, we resume the standard log-likelihood training of SB as $(\mathbf{z}_\theta, \mathbf{z}_\phi)$ no longer have trivial outputs.

Langevin-corrected generative process. Finally, let us discuss an application of modern training techniques to our SB-FBSDE. While the generative processes for both SGM and SB can be performed as simply as propagating (4) and (7b), it has been constantly observed that adopting Langevin annealing to the generative process greatly improves performance (Song et al., 2020b). This procedure, often referred to as the *Langevin corrector*, requires knowing the score function $\nabla_{\mathbf{x}}\log p_t(t,\mathbf{x}_t)$. For SB, we can estimate its value using the fact that $\mathbf{z}_\theta + \hat{\mathbf{z}}_\phi \approx g\nabla_{\mathbf{x}}\log p_t^{\text{SB}}$, which can be readily verified from (13). The procedure is summarized in Alg. 2, and we highlight this as the benefit gained from bridging SB with SGM through a rigorous analysis inspired by stochastic optimal control.

4 Experiments

4.1 Main results

Setups. We testify our likelihood training of SB on two synthetic toy datasets and three image datasets, *i.e.* MNIST, CelebA,[4] and CIFAR10. p_{prior} is set to a zero-mean Gaussian whose variance varies for each task and can be computed according to Song & Ermon (2020). We parameterize $\mathbf{z}_\theta$ and $\hat{\mathbf{z}}_\phi$ with residual-based networks for toy datasets and consider Unet (Ronneberger et al., 2015) and NCSN++ (Song et al., 2020b) respectively for MNIST/CelebA and CIFAR10. All networks adopt position encoding and are trained with AdamW (Loshchilov & Hutter, 2017) on a TITAN RTX. We adopt VE-SDE (*i.e.* $f := \mathbf{0}$; see Song et al. (2020b)) as our SDE backbone, which implies that in order to achieve reasonable performance, SB must *learn* a proper data-to-noise diffusion process. On all datasets, we set the horizon T =1.0 and solve the SDEs via the Euler-Maruyama method. The interval $[0,T]$ is discretized into 200 steps for CIFAR10 and 100 steps for all other datasets, which are much fewer than the ones in SGM ($\geq$1000 steps). Other details are left in Appendix B.

Toy datasets. We first validate our method on generating a mixture of Gaussian and checkerboard (adopted from Grathwohl et al. (2018)) as the representatives of continuous and discontinuous distributions. Figure 3 demonstrates how the learned policies, *i.e.* $\mathbf{z}_\theta$ and $\hat{\mathbf{z}}_\phi$, construct the vector fields that progressively transport samples back-and-forth between p_{prior} and p_{data}. Importantly, these vector fields can be highly nonlinear and dissimilar to each other. This resembles neither SGMs, whose forward vector field must obey linear structure, nor flow-based models, whose vector fields are simply with opposite directions, and we highlight this as a distinct feature arising from SB models.

Image datasets. Next, we validate our method on high-dimensional image generation. The generated images for MNIST, CelebA, and CIFAR10 are presented in Fig. 4, which clearly suggest that our SB-FBSDE is able to synthesize high-fidelity images. More uncurated images can be founded in

[4]We follow a similar setup of prior SB models (De Bortoli et al., 2021) and resize the image size to 32.

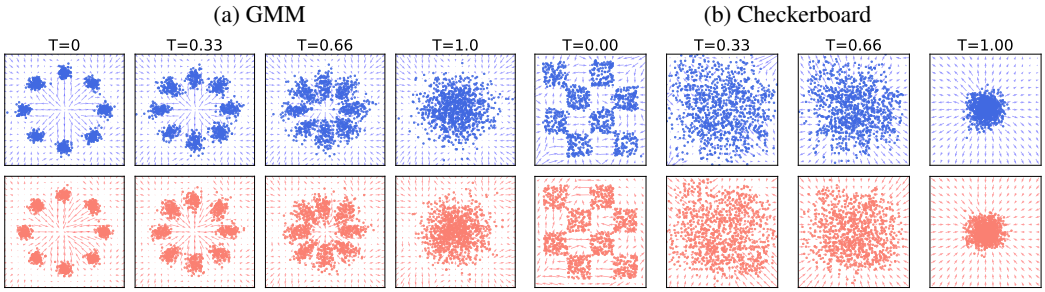

Figure 3: Validation of our SB-FBSDE model on two synthetic toy datasets that represent continuous and discontinuous distributions. *Upper*: Generation ($p_{\text{data}} \leftarrow p_{\text{prior}}$) process with the backward vector field $\hat{\mathbf{z}}_\phi$. *Bottom*: Diffusion ($p_{\text{data}} \rightarrow p_{\text{prior}}$) process with the forward vector field $\mathbf{z}_\theta$.

Figure 4: Uncurated samples from our SB-FBSDE models trained on MNIST (left), CelebA (middle) and CIFAR10 (right). More images can be found in Appendix C.

Table 2: CIFAR10 evaluation using negative log-likelihood (NLL; bits/dim) on the test set and sample quality (FID score) w.r.t. the training set. Our **SB-FBSDE** outperforms other optimal transport baselines by a large margin and is comparable to existing generative models.

Model Class	Method	NLL ↓	FID ↓
Optimal Transport	**SB-FBSDE (ours)**	2.98	3.18
	DOT (Tanaka, 2019)	-	15.78
	Multi-stage SB (Wang et al., 2021)	-	12.32
SGMs	SDE (deep, sub-VP; Song et al. (2020b))	2.99	2.92
	ScoreFlow (Song et al., 2021)	2.74	5.7
	VDM (Kingma et al., 2021)	**2.49**	4.00
	LSGM(Vahdat et al., 2021)	3.43	**2.10**
VAEs	VDVAE (Child, 2020)	2.87	-
	NVAE (Vahdat & Kautz, 2020)	2.91	23.49
	BIVA (Maaløe et al., 2019)	3.08	-
Flows	FFJORD (Grathwohl et al., 2018)	3.40	-
	VFlow (Chen et al., 2020)	2.98	-
	ANF (Huang et al., 2020)	3.05	-
GANs	AutoGAN (Gong et al., 2019)	-	12.42
	StyleGAN2-ADA (Karras et al., 2020a)	-	2.92
	LeCAM (Park & Kim, 2021)	-	2.47

Appendix C. Regarding the quantitative evaluation, Table 2 summarizes the negative log-likelihood (NLL; measured in bits/dim) and the Fréchet Inception Distance score (FID; Heusel et al. (2017)) on CIFAR10. For our SB-FBSDE, we compute the NLL on the test set using Corollary 5, in a similar vein to SGMs and flow-based models, and report the FID over 50k samples w.r.t the training

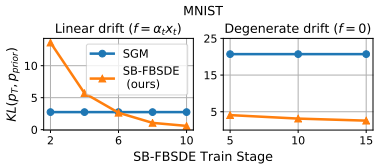

Figure 5: Validation of our SB-FBSDE on *learning* forward diffusions that are closer (in KL sense) to p_{prior} compared to SGM.

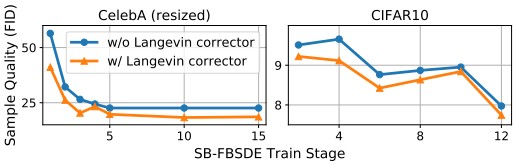

Figure 6: Ablation analysis where we show that adding Langevin corrector to SB-FBSDE uniformly improves the FID scores on both CelebA and CIFAR10 training.

set. Notably, our SB-FBSDE achieves 2.98 bits/dim and 3.18 FID score on CIFAR10, which is comparable to the top existing methods from other model classes (*e.g.* SGMs) and outperforms prior Optimal Transport (OT) methods (Wang et al., 2021; Tanaka, 2019) by a large margin in terms of the sample quality. More importantly, it enables log-likelihood computations that are otherwise infeasible in prior OT methods. We note that the quantitative comparisons on MNIST and CelebA are omitted as the scores on these two datasets are not widely reported and different pre-processing (*e.g.* resizing of CelebA) can lead to values that are not directly comparable.

4.2 Discussion

Validity of SB forward diffusion. Our theoretical analysis in Section 3.2 suggests that the forward control policy $\mathbf{z}_\theta$ plays an essential role in governing samples towards p_{prior}. Here, we validate this conjecture by computing the KL divergence between the terminal distribution induced by $\mathbf{z}_\theta$, *i.e.* $p_T^{(12a)}$, and the designated prior p_{prior}, then comparing the value with the one corresponding to SGM We refer readers to Appendix B for the actual computation. Figure 5 reports these comparisons over MNIST training. For both degenerate ($f := \mathbf{0}$) and linear ($f := \alpha_t \mathbf{x}_t$) base drifts, our SB-FBSDE generates terminal distributions that are much closer to p_{prior}. Note that the values of SGM remain unchanged throughout training since SGM relies on *pre-specified* diffusion. This is in contrast to our SB-FBSDE whose forward policy $\mathbf{z}_\theta$ gradually shortens the KL gap to p_{prior} as the training progresses, thereby providing a better forward diffusion for training the backward reversed policy $\hat{\mathbf{z}}_\phi$.

Effect of Langevin corrector. In practice, we observe that the Langevin corrector greatly affects the generative performance. As shown in Fig. 6, including these corrector steps uniformly improves the sample quality (FID) on both CelebA and CIFAR10 throughout training. Since the SDEs are often solved via the Euler-Maruyama method for computational efficiency, their propagation can be subjected to discretization errors accumulated over time. The Langevin steps thereby help re-distributing the samples at each time step t towards the desired density p_t^{SB}. We emphasize this improvement as the benefit gained from applying modern generative training techniques based on the solid connection between SB and SGM.

Limitations. Finally, let us discuss few limitations of our approach. Adopting SB formulation, by construction, requires one to maintain two distinct networks for $\mathbf{z}_\theta$ and $\hat{\mathbf{z}}_\phi$. Obvious as how this will lead to additional memory, it also increases the sampling runtime when using the Langevin corrector. Indeed, while SGM can compute the corrector values, $\nabla_{\mathbf{x}} \log p_t$, through a single call of its score network, our SB-FBSDE requires calling both networks, $\mathbf{z}_\theta + \hat{\mathbf{z}}_\phi \approx g \nabla_{\mathbf{x}} \log p_t^{\text{SB}}$. Fortunately, thanks to the mathematical flexibility of SB, SB-FBSDE requires much fewer time steps (100-200 steps) to achieve similar performance as SGM ($\geq$1000 steps). Hence, our SB-FBSDE still achieves an overall improvement on generation runtime compared to SGM without sacrificing the sample quality.

5 Conclusion

In this work, we present a novel computational framework for the likelihood training of Schrödinger Bridge (SB), a recently emerging method that adopts optimal transport for generative modeling. Grounded on Forward-Backward SDEs – a mathematical methodology that widely appears in stochastic optimal control, we show that there is an innate connection between the optimality condition of SB and the log-likelihood objective of Score-based Generative Model (SGM). From which, we derive equivalent log-likelihood objectives for SB that generalize previous theoretical results for SGM and facilitate applications of modern generative training for SB. We validate our training method on various image generative tasks, *e.g.* MNIST, CelebA, and CIFAR10, showing encouraging results in synthesizing high-fidelity samples while retaining the rigorous optimal transport framework.

REFERENCES

Brian DO Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

Kenneth Caluya and Abhishek Halder. Wasserstein proximal algorithms for the schrödinger bridge problem: Density control with nonlinear drift. *IEEE Transactions on Automatic Control*, 2021.

Jianfei Chen, Cheng Lu, Biqi Chenli, Jun Zhu, and Tian Tian. Vflow: More expressive generative flows with variational data augmentation. In *International Conference on Machine Learning*, pp. 1660–1669. PMLR, 2020.

Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, pp. 6572–6583, 2018.

Rewon Child. Very deep vaes generalize autoregressive models and can outperform them on images. *arXiv preprint arXiv:2011.10650*, 2020.

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

Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. *arXiv preprint arXiv:2106.01357*, 2021.

Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. *arXiv preprint arXiv:2105.05233*, 2021.

Ioannis Exarchos and Evangelos A Theodorou. Stochastic optimal control via forward and backward stochastic differential equations and importance sampling. *Automatica*, 87:159–165, 2018.

Wenbo Gong and Yingzhen Li. Interpreting diffusion score matching using normalizing flow. *arXiv preprint arXiv:2107.10072*, 2021.

Xinyu Gong, Shiyu Chang, Yifan Jiang, and Zhangyang Wang. Autogan: Neural architecture search for generative adversarial networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3224–3234, 2019.

Will Grathwohl, Ricky TQ Chen, Jesse Betterncourt, Ilya Sutskever, and David Duvenaud. Ffjord: Free-form continuous dynamics for scalable reversible generative models. *arXiv preprint arXiv:1810.01367*, 2018.

Jiequn Han, Arnulf Jentzen, and E Weinan. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences*, 115(34):8505–8510, 2018.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arXiv:2006.11239*, 2020.

Chin-Wei Huang, Laurent Dinh, and Aaron Courville. Augmented normalizing flows: Bridging the gap between generative flows and latent variable models. *arXiv preprint arXiv:2002.07101*, 2020.

Chin-Wei Huang, Jae Hyun Lim, and Aaron Courville. A variational perspective on diffusion-based generative models and score matching. *arXiv preprint arXiv:2106.02808*, 2021.

Michael F Hutchinson. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics-Simulation and Computation*, 18(3):1059–1076, 1989.

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

Alexia Jolicoeur-Martineau, Ke Li, Rémi Piché-Taillefer, Tal Kachman, and Ioannis Mitliagkas. Gotta go fast when generating data with score-based models. *arXiv preprint arXiv:2105.14080*, 2021.

Ioannis Karatzas and Steven Shreve. *Brownian motion and stochastic calculus*, volume 113. Springer Science & Business Media, 2012.

Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *arXiv preprint arXiv:2006.06676*, 2020a.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8110–8119, 2020b.

Diederik P Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *arXiv preprint arXiv:2107.00630*, 2021.

Magdalena Kobylanski. Backward stochastic differential equations and partial differential equations with quadratic growth. *Annals of probability*, pp. 558–602, 2000.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *arXiv preprint arXiv:2010.05646*, 2020a.

Zhifeng Kong and Wei Ping. On fast sampling of diffusion probabilistic models. *arXiv preprint arXiv:2106.00132*, 2021.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*, 2020b.

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

Flavien Léger and Wuchen Li. Hopf–cole transformation via generalized schrödinger bridge problem. *Journal of Differential Equations*, 274:788–827, 2021.

Christian Léonard. A survey of the schrödinger problem and some of its connections with optimal transport. *arXiv preprint arXiv:1308.0215*, 2013.

J-P Lepeltier and J San Martín. Existence for bsde with superlinear–quadratic coefficient. *Stochastics: An International Journal of Probability and Stochastic Processes*, 63(3-4):227–240, 1998.

Qianxiao Li and Shuji Hao. An optimal control approach to deep learning and applications to discrete-weight neural networks. *arXiv preprint arXiv:1803.01299*, 2018.

Guan-Horng Liu, Tianrong Chen, and Evangelos A Theodorou. Ddpnopt: Differential dynamic programming neural optimizer. In *International Conference on Learning Representations*, 2021a.

Guan-Horng Liu, Tianrong Chen, and Evangelos A Theodorou. Dynamic game theoretic neural optimizer. In *International Conference on Machine Learning*, 2021b.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2837–2845, 2021.

Jin Ma, J-M Morel, and Jiongmin Yong. *Forward-backward stochastic differential equations and their applications*. Number 1702. Springer Science & Business Media, 1999.

Lars Maaløe, Marco Fraccaro, Valentin Liévin, and Ole Winther. Biva: A very deep hierarchy of latent variables for generative modeling. *arXiv preprint arXiv:1902.02102*, 2019.

Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. *arXiv preprint arXiv:2102.09672*, 2021.

Bernt Øksendal. Stochastic differential equations. In *Stochastic differential equations*, pp. 65–84. Springer, 2003.

Jeeseung Park and Younggeun Kim. Styleformer: Transformer based generative adversarial networks with style vector. *arXiv preprint arXiv:2106.07023*, 2021.

Michele Pavon and Anton Wakolbinger. On free energy, stochastic control, and schrödinger processes. In *Modeling, Estimation and Control of Systems with Uncertainty*, pp. 334–348. Springer, 1991.

Marcus Pereira, Ziyi Wang, Ioannis Exarchos, and Evangelos A Theodorou. Neural network architectures for stochastic control using the nonlinear feynman-kac lemma. *arXiv preprint arXiv:1902.03986*, 2019.

Wei Ping, Kainan Peng, Kexin Zhao, and Zhao Song. Waveflow: A compact flow-based model for raw audio. In *International Conference on Machine Learning*, pp. 7706–7716. PMLR, 2020.

Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017.

Michael Renardy and Robert C Rogers. *An introduction to partial differential equations*, volume 13. Springer Science & Business Media, 2006.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Ludger Ruschendorf. Convergence of the iterative proportional fitting procedure. *The Annals of Statistics*, pp. 1160–1174, 1995.

Erwin Schrödinger. 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.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *arXiv preprint arXiv:1907.05600*, 2019.

Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *arXiv preprint arXiv:2006.09011*, 2020.

Yang Song, Sahaj Garg, Jiaxin Shi, and Stefano Ermon. Sliced score matching: A scalable approach to density and score estimation. In *Uncertainty in Artificial Intelligence*, pp. 574–584. PMLR, 2020a.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020b.

Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. *arXiv e-prints*, pp. arXiv–2101, 2021.

Akinori Tanaka. Discriminator optimal transport. *arXiv preprint arXiv:1910.06832*, 2019.

Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. *arXiv preprint arXiv:2007.03898*, 2020.

Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. *arXiv preprint arXiv:2106.05931*, 2021.

Francisco Vargas, Pierre Thodoroff, Neil D Lawrence, and Austen Lamacraft. Solving schrödinger bridges via maximum likelihood. *arXiv preprint arXiv:2106.02081*, 2021.

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

Gefei Wang, Yuling Jiao, Qian Xu, Yang Wang, and Can Yang. Deep generative learning via schrödinger bridge. *arXiv preprint arXiv:2106.10410*, 2021.

Jiongmin Yong and Xun Yu Zhou. *Stochastic controls: Hamiltonian systems and HJB equations*, volume 43. Springer Science & Business Media, 1999.

A MISSING PROOFS AND REMARKS IN SECTION 3

A.1 POOF OF THEOREM 3

The following lemma will be useful for proving Theorem 3.

Lemma 6.

$$\frac{1}{p}\operatorname{Tr}\left(p_{xx}\right)=\|\nabla \log p\|^2+\Delta \log p$$

Proof.

$$\begin{aligned}\frac{1}{p}\operatorname{Tr}\left(p_{xx}\right)&=\frac{1}{p}\Delta p_{xx}=\frac{1}{p}\nabla\cdot\nabla p=\frac{1}{p}\nabla\cdot\left(p\nabla\log p\right)\\&=\frac{1}{p}\left(p\Delta\log p+\nabla p^{\mathsf{T}}\nabla\log p\right)=\|\nabla\log p\|^2+\Delta\log p\end{aligned}$$

□

Now, apply itô lemma to $\log\Psi(t,\mathbf{x}_t)$ w.r.t. the reference measure $\mathbb{P}$ governed by the base SDE (1):

$$\begin{aligned}&\mathrm{d}\log\Psi\\=&\frac{\partial\log\Psi}{\partial t}\mathrm{d}t+\left(\frac{\partial\log\Psi}{\partial\mathbf{x}}\right)^{\mathsf{T}}\mathrm{d}\mathbf{x}+\frac{1}{2}\mathrm{d}\mathbf{x}^{\mathsf{T}}\left(\frac{\partial^2\log\Psi}{\partial\mathbf{x}\partial\mathbf{x}}\right)\mathrm{d}\mathbf{x}\\=&\frac{1}{\Psi}\left(\cancel{-\Psi_{\mathbf{x}}^{\mathsf{T}}f}-\cancel{\frac{1}{2}\operatorname{Tr}(g^2\Psi_{\mathbf{xx}})}\right)\mathrm{d}t+\frac{1}{\Psi}\left(\Psi_{\mathbf{x}}^{\mathsf{T}}(\cancel{f\,\mathrm{d}t}+g\mathrm{d}\mathbf{w}_t)\right)+\frac{1}{2}\operatorname{Tr}\left(g^2\left(\cancel{\frac{1}{\Psi}\Psi_{\mathbf{xx}}}-\frac{1}{\Psi^2}\Psi_{\mathbf{x}}^2\right)\right)\mathrm{d}t\\=&-\frac{1}{2}\|g\nabla\log\Psi\|^2\,\mathrm{d}t+g\left(\nabla\log\Psi\right)^{\mathsf{T}}\mathrm{d}\mathbf{w}_t.\end{aligned}$$

Hence, denote $\mathbf{y}_t=\log\Psi$ and $\mathbf{z}_t=g\nabla_{\mathbf{x}}\log\Psi$, the associated FBSDEs representation to the dynamics of $\log\Psi$ is given by

$$\mathrm{d}\mathbf{x}_t=f\,\mathrm{d}t+g\mathrm{d}\mathbf{w}_t, \tag{16a}$$

$$\mathrm{d}\mathbf{y}_t=-\frac{1}{2}\mathbf{z}_t^{\mathsf{T}}\mathbf{z}_t\mathrm{d}t+\mathbf{z}_t^{\mathsf{T}}\mathrm{d}\mathbf{w}_t. \tag{16b}$$

Likewise, apply itô lemma to $\log\hat{\Psi}$ w.r.t. the same reference measure $\mathbb{P}$.

$$\begin{aligned}&\mathrm{d}\log\hat{\Psi}\\=&\frac{\partial\log\hat{\Psi}}{\partial t}\mathrm{d}t+\left(\frac{\partial\log\hat{\Psi}}{\partial\mathbf{x}}\right)^{\mathsf{T}}\mathrm{d}\mathbf{x}+\frac{1}{2}\mathrm{d}\mathbf{x}^{\mathsf{T}}\left(\frac{\partial^2\log\hat{\Psi}}{\partial\mathbf{x}\partial\mathbf{x}}\right)\mathrm{d}\mathbf{x}\\=&\frac{1}{\hat{\Psi}}\left(-\nabla\cdot(\hat{\Psi}f)+\frac{1}{2}\operatorname{Tr}(g^2\hat{\Psi}_{\mathbf{xx}})\right)\mathrm{d}t+\frac{1}{\hat{\Psi}}\left(\hat{\Psi}_{\mathbf{x}}^{\mathsf{T}}(f\,\mathrm{d}t+g\mathrm{d}\mathbf{w}_t)\right)+\frac{1}{2}\operatorname{Tr}\left(g^2\left(\frac{1}{\hat{\Psi}}\hat{\Psi}_{\mathbf{xx}}-\frac{1}{\hat{\Psi}^2}\hat{\Psi}_{\mathbf{x}}^2\right)\right)\mathrm{d}t\\=&\frac{1}{\hat{\Psi}}\left(-\hat{\Psi}(\nabla\cdot f)-\cancel{\hat{\Psi}_{\mathbf{x}}^{\mathsf{T}}b}+\frac{1}{2}\operatorname{Tr}(g^2\hat{\Psi}_{\mathbf{xx}})\right)\mathrm{d}t+\frac{1}{\hat{\Psi}}\left(\hat{\Psi}_{\mathbf{x}}^{\mathsf{T}}(\cancel{f\,\mathrm{d}t}+g\mathrm{d}\mathbf{w}_t)\right)+\frac{1}{2}\operatorname{Tr}\left(g^2\left(\frac{1}{\hat{\Psi}}\hat{\Psi}_{\mathbf{xx}}-\frac{1}{\hat{\Psi}^2}\hat{\Psi}_{\mathbf{x}}^2\right)\right)\mathrm{d}t\\=&\left(-\nabla\cdot f+\frac{1}{\hat{\Psi}}\operatorname{Tr}(g^2\hat{\Psi}_{\mathbf{xx}})-\frac{1}{2}\left\|g\nabla\log\hat{\Psi}\right\|^2\right)\mathrm{d}t+g\left(\nabla\log\hat{\Psi}\right)^{\mathsf{T}}\mathrm{d}\mathbf{w}_t\\=&\left(-\nabla\cdot f+\nabla\cdot g^2\nabla\log\hat{\Psi}+\frac{1}{2}\left\|g\nabla\log\hat{\Psi}\right\|^2\right)\mathrm{d}t+g\left(\nabla\log\hat{\Psi}\right)^{\mathsf{T}}\mathrm{d}\mathbf{w}_t,\end{aligned}$$

where the last equality follows from Lemma 6. Now, denote $\hat{\mathbf{y}}(t,\mathbf{x})=\log\hat{\Psi}$ and $\hat{\mathbf{z}}(t,\mathbf{x})=g\nabla\log\hat{\Psi}$, the associated FBSDEs representation to the dynamics of $\log\hat{\Psi}$ is given by

$$\mathrm{d}\mathbf{x}_t=f\,\mathrm{d}t+g\mathrm{d}\mathbf{w}_t, \tag{17a}$$

$$\mathrm{d}\hat{\mathbf{y}}_t=\left[\frac{1}{2}\hat{\mathbf{z}}_t^{\mathsf{T}}\hat{\mathbf{z}}_t+\nabla\cdot(g\hat{\mathbf{z}}_t-f)\right]\mathrm{d}t+\hat{\mathbf{z}}_t^{\mathsf{T}}\mathrm{d}\mathbf{w}_t. \tag{17b}$$

Notice that (17a) coincides with (16a). Hence, the collection of SDEs (16a, 16b, 17b) gives the FBSDEs representation of (6) w.r.t. the reference measure $\mathbb{P}$.

Our final step is to apply Girsanov's Theorem (Øksendal, 2003) on change of measure so that $\mathbf{x}_t$ is sampled from a *controlled* SDE with the optimal forward drift given by (7a). This can be done by applying importance sampling techniques to FBSDEs (Exarchos & Theodorou, 2018). In our case, it leads to

$$\boxed{\begin{aligned} \mathrm{d}\mathbf{x}_t &= (f + g\mathbf{z}_t)\,\mathrm{d}t + g\mathrm{d}\mathbf{w}_t && \text{(18a)}\\ \mathrm{d}\mathbf{y}_t &= \frac{1}{2}\mathbf{z}_t^\top \mathbf{z}_t \mathrm{d}t + \mathbf{z}_t^\top \mathrm{d}\mathbf{w}_t && \text{(18b)}\\ \mathrm{d}\hat{\mathbf{y}}_t &= \left[\frac{1}{2}\hat{\mathbf{z}}_t^\top \hat{\mathbf{z}}_t + \nabla\cdot(g\hat{\mathbf{z}}_t - f) + \hat{\mathbf{z}}_t^\top \mathbf{z}_t\right]\mathrm{d}t + \hat{\mathbf{z}}_t^\top \mathrm{d}\mathbf{w}_t && \text{(18c)} \end{aligned}}$$

which concludes the proof.

A.2 Poof of Theorem 4

Recall that $\mathbf{y}_t$ and $\hat{\mathbf{y}}_t$ are stochastic representation of $\log\Psi$ and $\log\hat{\Psi}$. Since

$$\mathcal{L}_{\text{SB}}(\mathbf{x}_0) \triangleq \log p_0(\mathbf{x}_0) = \mathbb{E}\left[\mathbf{y}(0,\mathbf{x}_0) + \hat{\mathbf{y}}(0,\mathbf{x}_0)\right],$$

we can expand the RHS with Theorem 3. This leads to

$$\begin{aligned} &\mathcal{L}_{\text{SB}}(\mathbf{x}_0)\\ =&\mathbb{E}\left[\mathbf{y}_0(0,\mathbf{x}_0) + \hat{\mathbf{y}}_0(0,\mathbf{x}_0)\right]\\ =&\mathbb{E}\left[\mathbf{y}_T - \int_0^T \left(\frac{1}{2}\left\|\mathbf{z}_t\right\|^2\right)\mathrm{d}t\right] + \mathbb{E}\left[\hat{\mathbf{y}}_T - \int_0^T \left(\frac{1}{2}\left\|\hat{\mathbf{z}}_t\right\|^2 + \nabla\cdot(g\hat{\mathbf{z}}_t - f) + \hat{\mathbf{z}}_t^\top \mathbf{z}_t\right)\mathrm{d}t\right]\\ =&\mathbb{E}\left[\mathbf{y}_T(T,\mathbf{x}_T) + \hat{\mathbf{y}}_T(T,\mathbf{x}_T)\right] - \int_0^T \mathbb{E}\left[\frac{1}{2}\left\|\mathbf{z}_t\right\|^2 + \frac{1}{2}\left\|\hat{\mathbf{z}}_t\right\|^2 + \nabla\cdot(g\hat{\mathbf{z}}_t - f) + \hat{\mathbf{z}}_t^\top \mathbf{z}_t\right]\mathrm{d}t\\ =&\mathbb{E}\log p_T(\mathbf{x}_T) - \int_0^T \underbrace{\mathbb{E}\left[\frac{1}{2}\left\|\mathbf{z}_t\right\|^2 + \frac{1}{2}\left\|\hat{\mathbf{z}}_t\right\|^2 + \nabla\cdot(g\hat{\mathbf{z}}_t - f) + \hat{\mathbf{z}}_t^\top \mathbf{z}_t\right]}_{(*)}\mathrm{d}t. \end{aligned} \tag{19}$$

Finally, applying integration by part to $(*)$ yields

$$\begin{aligned} (*) =&\mathbb{E}\left[\frac{1}{2}\left\|\mathbf{z}_t\right\|^2 + \frac{1}{2}\left\|\hat{\mathbf{z}}_t\right\|^2 + \nabla\cdot(g\hat{\mathbf{z}}_t - f) + \hat{\mathbf{z}}_t^\top \mathbf{z}_t\right]\\ =&\mathbb{E}\left[\frac{1}{2}\left\|\mathbf{z}_t\right\|^2 + \frac{1}{2}\left\|\hat{\mathbf{z}}_t\right\|^2 - \hat{\mathbf{z}}_t^\top\left(g\nabla\log p_t^{\text{SB}}\right) - \nabla\cdot f + \hat{\mathbf{z}}_t^\top \mathbf{z}_t\right]\\ =&\mathbb{E}\left[\frac{1}{2}\left\|\mathbf{z}_t\right\|^2 + \frac{1}{2}\left\|\hat{\mathbf{z}}_t - g\nabla\log p_t^{\text{SB}} + \mathbf{z}_t\right\|^2 - \frac{1}{2}\left\|g\nabla\log p_t^{\text{SB}} - \mathbf{z}_t\right\|^2 - \nabla\cdot f\right]. \end{aligned} \tag{20}$$

Equations (19) and (20) conclude the proof.

A.3 Proof of Corollary 5

Recall the forward SDE (12a) of our SB-FBSDE:

$$\mathrm{d}\mathbf{x}_t = (f + g\mathbf{z}_t)\,\mathrm{d}t + g\mathrm{d}\mathbf{w}_t.$$

It is known that the probability ODE flow of an SDE can be expressed as

$$\mathrm{d}\mathbf{x}_t = \left[f + g\mathbf{z}_t - \frac{1}{2}g^2\nabla_{\mathbf{x}}\log p_t^{\text{SB}}\right]\mathrm{d}t, \tag{21}$$

where p_t^{SB} is the marginal density of (12a) at time t. Hence, substituting $\mathbf{z}_t + \hat{\mathbf{z}}_t = g\nabla_{\mathbf{x}}\log p_t^{\text{SB}}$ to (21) immediately concludes the proof.

A.4 DERIVATION OF $\mathcal{L}_{\text{SB}}(\mathbf{x}_T; \theta, \phi)$

As we briefly mentioned in Section 3.2, due to the symmetric structure of SB, we can repeat the same derivation for Theorem 3 and 4 using a re-parametrization of time

$$s \triangleq T - t.$$

Under this new time coordinate, the SB optimality suggests that the optimal path measure to the same optimization (5) is given by

$$\mathrm{d}\mathbf{x}_s = [\bar{f} + \bar{g}^2 \, \nabla_{\mathbf{x}} \log \Psi'(s, \mathbf{x}_s)]\mathrm{d}t + \bar{g} \, \mathrm{d}\mathbf{w}_s, \quad \mathbf{x}_0 \sim p_{\text{prior}}, \tag{22a}$$

$$\mathrm{d}\mathbf{x}_s = [\bar{f} - \bar{g}^2 \, \nabla_{\mathbf{x}} \log \hat{\Psi}'(s, \mathbf{x}_s)]\mathrm{d}t + \bar{g} \, \mathrm{d}\mathbf{w}_s, \quad \mathbf{x}_T \sim p_{\text{data}}, \tag{22b}$$

where Ψ' and $\hat{\Psi}'$ obey the same PDE dynamics (6) except with the flipped boundary conditions,

$$\Psi'(0, \cdot)\hat{\Psi}'(0, \cdot) = p_{\text{prior}}, \;\; \Psi'(T, \cdot)\hat{\Psi}'(T, \cdot) = p_{\text{data}},$$

and $(\bar{f}, \bar{g})$ relate to (f, g) in (1) by

$$\begin{aligned} \bar{f}(s, \mathbf{x}_s) &= g(s)^2 \, \nabla_{\mathbf{x}} \log p_s - f(s, \mathbf{x}_s), \\ \bar{g}(s) &= g(s). \end{aligned}$$

Equating (22) to (7) and after some arrangement, one would arrive at the following formula:

$$\mathbf{z}'_s = \hat{\mathbf{z}}_s - g(s) \, \nabla_{\mathbf{x}} \log p_s \qquad \text{and} \qquad \hat{\mathbf{z}}'_s = \mathbf{z}_s + g(s) \, \nabla_{\mathbf{x}} \log p_s,$$

where $(\mathbf{z}'_s, \hat{\mathbf{z}}'_s)$ relate to $(\Psi', \hat{\Psi}')$ similar to (13), *i.e.*

$$\mathbf{z}'_s \equiv \mathbf{z}'(s, \mathbf{x}_s) = \bar{g}\nabla_{\mathbf{x}} \log \Psi'(s, \mathbf{x}_s) \qquad \text{and} \qquad \hat{\mathbf{z}}'_s \equiv \hat{\mathbf{z}}'(s, \mathbf{x}_s) = \bar{g}\nabla_{\mathbf{x}} \log \hat{\Psi}'(s, \mathbf{x}_s).$$

Finally, substituting these transformations into the FBSDEs corresponding to the time coordinate s:

$$\boxed{\begin{aligned} \mathrm{d}\mathbf{x}_s &= \left(\bar{f} + \bar{g}\mathbf{z}'_s\right) \mathrm{d}t + \bar{g}\mathrm{d}\mathbf{w}_s && \text{(23a)} \\ \mathrm{d}\mathbf{y}_s &= \frac{1}{2} \left\|\mathbf{z}'_s\right\|^2 \mathrm{d}t + {\mathbf{z}'_s}^{\mathsf{T}} \mathrm{d}\mathbf{w}_s && \text{(23b)} \\ \mathrm{d}\hat{\mathbf{y}}_s &= \left[\frac{1}{2} \left\|\hat{\mathbf{z}}'_s\right\|^2 + \nabla \cdot (\bar{g}\hat{\mathbf{z}}'_s - \bar{f}) + {\hat{\mathbf{z}}'_s}^{\mathsf{T}} \mathbf{z}'_s\right] \mathrm{d}t + {\hat{\mathbf{z}}'_s}^{\mathsf{T}} \mathrm{d}\mathbf{w}_s && \text{(23c)} \end{aligned}}$$

and following the same derivation in Appendix A.2 yield

$$\begin{aligned} &\mathcal{L}_{\text{SB}}(\mathbf{x}_T; \theta, \phi) \\ =&\mathbb{E}\left[\log p_0(\mathbf{x}_0)\right] - \int_0^T \mathbb{E}\left[\frac{1}{2} \left\|\hat{\mathbf{z}}_s\right\|^2 + \frac{1}{2} \left\|\mathbf{z}_s\right\|^2 + \nabla \cdot (g\mathbf{z}_s + f) + \mathbf{z}_s^{\mathsf{T}} \hat{\mathbf{z}}_s\right] \mathrm{d}s, && (24) \\ =&\mathbb{E}\left[\log p_0(\mathbf{x}_0)\right] - \int_0^T \mathbb{E}\left[\frac{1}{2} \left\|\hat{\mathbf{z}}_s\right\|^2 + \frac{1}{2} \left\|\mathbf{z}_s - g\nabla \log p_t^{\text{SB}} + \hat{\mathbf{z}}_s\right\|^2 - \frac{1}{2} \left\|g\nabla \log p_t^{\text{SB}} - \hat{\mathbf{z}}_s\right\|^2 + \nabla \cdot f\right] \mathrm{d}s. \end{aligned}$$

Interestingly, the new log-likelihood objective in (24) simply switches the position between $\mathbf{z}$ and $\hat{\mathbf{z}}$. We highlight this as the result of the symmetric structure of SB, where changing the sampling direction from forward (7a) to backward (7b) flips the role of $\mathbf{z}$ and $\hat{\mathbf{z}}$ correspondingly.

A.5 REMAKRS ON LEMMA 2 AND COROLLARY 5

Following Lepeltier & Martín (1998); Kobylanski (2000); Yong & Zhou (1999), the regularity conditions require *(i)* G, f, h and ϕ to be continuous, *(ii)* G and f to be uniformly Lipschitz in $\mathbf{x}$, and *(iii)* h to be continuous and satisfy quadratic growth in $\mathbf{z}$.

Finally, we demonstrate how applying flow-based training techniques to the probability ODE flow of SB (15) recovers the same log-likelihood objective in (14). Recall that given an ODE $\mathrm{d}\mathbf{x}_t = F(t, \mathbf{x}_t)\mathrm{d}t$ with $\mathbf{x}_0 \sim p_{\text{data}}$, flow-based models compute the change in log-density using the instantaneous change of variables formula (Chen et al., 2018):

$$\frac{\partial \log p(\mathbf{x}_t)}{\partial t} = -\nabla_{\mathbf{x}} \cdot F,$$

which implies that the log-likelihood of $\mathbf{x}_0$ can be computed as

$$\log p(\mathbf{x}_T) = \log p(\mathbf{x}_0) - \int_0^T \nabla_{\mathbf{x}} \cdot F \, \mathrm{d}t. \tag{25}$$

Now, consider the probability ODE flow of SB

$$F_{\text{SB}} := f + g\mathbf{z}_t - \frac{1}{2}g(\mathbf{z}_t + \hat{\mathbf{z}}_t) = f + \frac{1}{2}g(\mathbf{z}_t - \hat{\mathbf{z}}_t).$$

Substituting this vector field F_{SB} to (25) yields

$$\begin{aligned}
\log p_T(\mathbf{x}_T) &= \log p_0(\mathbf{x}_0) - \int_0^T \nabla_{\mathbf{x}} \cdot \left(f + \frac{1}{2}g(\mathbf{z}_t - \hat{\mathbf{z}}_t) \right) \mathrm{d}t \\
\Rightarrow \mathbb{E}\left[\log p_0(\mathbf{x}_0)\right] &= \mathbb{E}\left[\log p_T(\mathbf{x}_T)\right] + \int_0^T \mathbb{E}\left[\nabla_{\mathbf{x}} \cdot \left(f + \frac{1}{2}g(\mathbf{z}_t - \hat{\mathbf{z}}_t) \right) \right] \mathrm{d}t \\
\Rightarrow \mathbb{E}\left[\log p_0(\mathbf{x}_0)\right] &= \mathbb{E}\left[\log p_T(\mathbf{x}_T)\right] - \int_0^T \mathbb{E}\left[\nabla_{\mathbf{x}} \cdot (g\hat{\mathbf{z}}_t - f) - \frac{1}{2}g\nabla_{\mathbf{x}} \cdot (\mathbf{z}_t + \hat{\mathbf{z}}_t) \right] \mathrm{d}t \\
\overset{(*)}{\Longrightarrow} \mathbb{E}\left[\log p_0(\mathbf{x}_0)\right] &= \mathbb{E}\left[\log p_T(\mathbf{x}_T)\right] - \int_0^T \mathbb{E}\left[\nabla_{\mathbf{x}} \cdot (g\hat{\mathbf{z}}_t - f) + \frac{1}{2}g(\mathbf{z}_t + \hat{\mathbf{z}}_t)^{\mathsf{T}} \left(\nabla_{\mathbf{x}} \log p_t^{\text{SB}} \right) \right] \mathrm{d}t \\
\overset{(**)}{\Longrightarrow} \mathbb{E}\left[\log p_0(\mathbf{x}_0)\right] &= \mathbb{E}\left[\log p_T(\mathbf{x}_T)\right] - \int_0^T \mathbb{E}\left[\nabla_{\mathbf{x}} \cdot (g\hat{\mathbf{z}}_t - f) + \frac{1}{2}(\mathbf{z}_t + \hat{\mathbf{z}}_t)^2 \right] \mathrm{d}t,
\end{aligned} \tag{26}$$

where (*) is due to integration by parts and (**) uses the fact that $\mathbf{z}_t + \hat{\mathbf{z}}_t = g\nabla_{\mathbf{x}} \log p_t^{\text{SB}}$. One can verify that (26) indeed recovers (14).

B Experiment Details

Figure 7: Training Hyper-parameters

Dataset	learning rate	time steps	batch size	variance of p_{prior}
Toy	2e-4	100	400	1.0
Mnist	2e-4	100	200	1.0
CelebA	2e-4	100	200	900.0
CIFAR10	1e-5	200	64	2500.0

Figure 8: Network Architectures

Dataset	$\mathbf{z}_\theta$	$\hat{\mathbf{z}}_\phi$	# of parameters of $\mathbf{z}_\theta$	# of parameters of $\hat{\mathbf{z}}_\phi$
Toy	FC-ResNet	FC-ResNet	0.76M	0.76M
Mnist	reduced Unet	reduced Unet	1.95M	1.95M
CelebA	Unet	Unet	39.63	39.63
CIFAR10	NCSN++	Unet	62.69M	39.63M

Training. We use Exponential Moving Average (EMA) with the decay rate of 0.99. Table 7 details the hyper-parameters used for each dataset. In practice, alternating the training between $\min_\phi \mathcal{L}_{\text{SB}}(\mathbf{x}_0)$ and $\min_\theta \mathcal{L}_{\text{SB}}(\mathbf{x}_T)$ implies that we may discard the computational graph of state propagation. This is due to the fact that the gradient of *e.g.* $\mathcal{L}_{\text{SB}}(\mathbf{x}_0)$ w.r.t. $\hat{\mathbf{z}}_\phi$ does not flow through its associated sampling distribution (12a). This greatly reduces the memory consumption. Additionally, the divergence appeared in (14) can be estimated efficiently using Hutchinson's trace estimator (Hutchinson, 1989).

Sampling. For the completeness, we illustrate howthe noise scale in Alg. 2 is computed. Given some pre-defined signal-to-noise ratio r (we set r =0.05 for all experiments), the Langevin noise scale σ_i at each time step t is given by

$$\sigma_i = 2\left(r \frac{\|\epsilon\|}{\left\|\log p_t^{\text{SB}}\right\|} \right)^2,$$

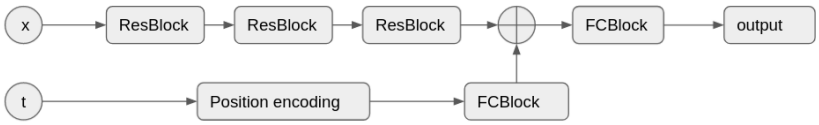

Figure 9: Network architecture for toy datasets.

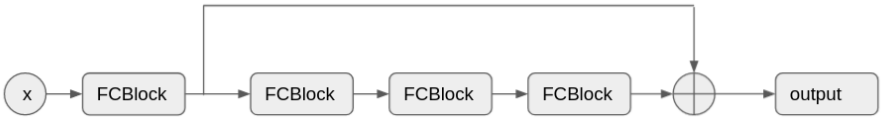

Figure 10: Residual block appeared in Fig. 9.

where ϵ is the Gaussian noise and $\log p_t^{\text{SB}}$ can be estimated through $\mathbf{z}_\theta$ and $\hat{\mathbf{z}}_\phi$.

Data pre-processing. MNIST is padded from 28×28 to 32×32 to prevent degenerate feature maps through Unet. CelebA is resized to 3×32×32 to accelerate training. Both CelebA and CIFAR10 are augmented with random horizontal flips to enhance the diversity.

Network architectures. Table 8 summarizes the network architecture used for each dataset. For toy datasets, we parameterize $\mathbf{z}_\theta$ and $\mathbf{z}_\phi$ with the architectures shown in Fig. 9. Specifically, *FCBlock* represents a fully connected layer followed by a swish nonlinear activation (Ramachandran et al., 2017), whereas the architecture of *ResBlock* is shown in Fig. 10. As for MNIST, we consider a smaller version of Unet (Ho et al., 2020) by reducing the numbers of residual block, attention heads, and channels respectively to 1, 2, and 32. Unet and NCSN++ respectively correspond to the architectures appeared in Ho et al. (2020) and Song et al. (2020b).

Remarks on Table 2. We note that the values of our SB-FBSDE reported in Table 2 are computed *without* the Langevin corrector due to the computational constraint. For all other experiments, we adopt the Langevin corrector as it generally improves the performance (see Fig. 6). This implies that our results on CIFAR10, despite already being encouraging, may be further improved with the Langevin corrector.

Remarks on Fig. 5. To estimating $\mathrm{KL}(p_T, p_{\text{prior}})$, we first compute the pixel-wise first and second moments given the generated samples $\mathbf{x}_T$ at the end of the forward diffusion. After fitting a diagonal Gaussian to $\{\mathbf{x}_T\}$, we can apply the analytic formula for computing the KL divergence between two multivariate Gaussians.

Remarks on Fig. 6. To accelerate the sampling process with the Langevin corrector, for this experiment we consider a reduced Unet (see Table 8) for CelebA. The FID scores on both datasets are computed with 10k samples. We stress, however, that the performance improvement using the Langevin corrector remains consistent across other (larger) architectures and if one increases the FID samples.

C ADDITIONAL FIGURES

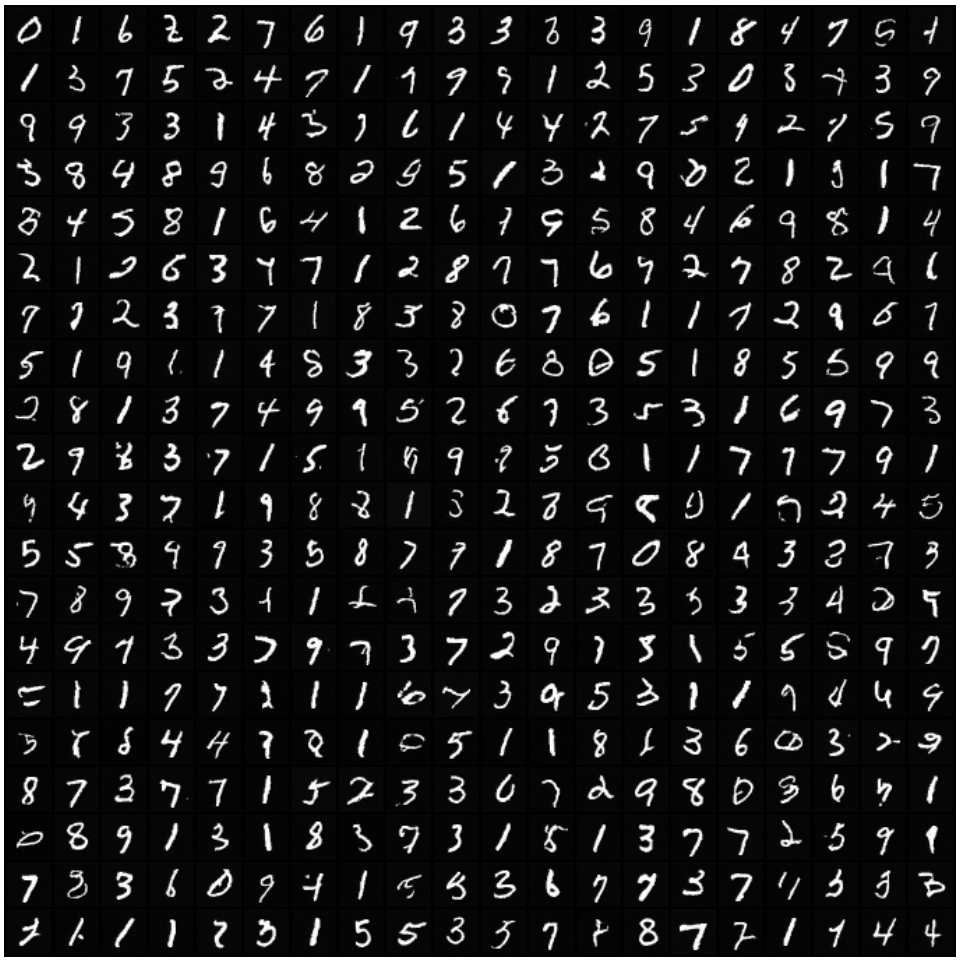

Figure 11: Uncurated samples generated by our SB-FBSDE on MNIST.

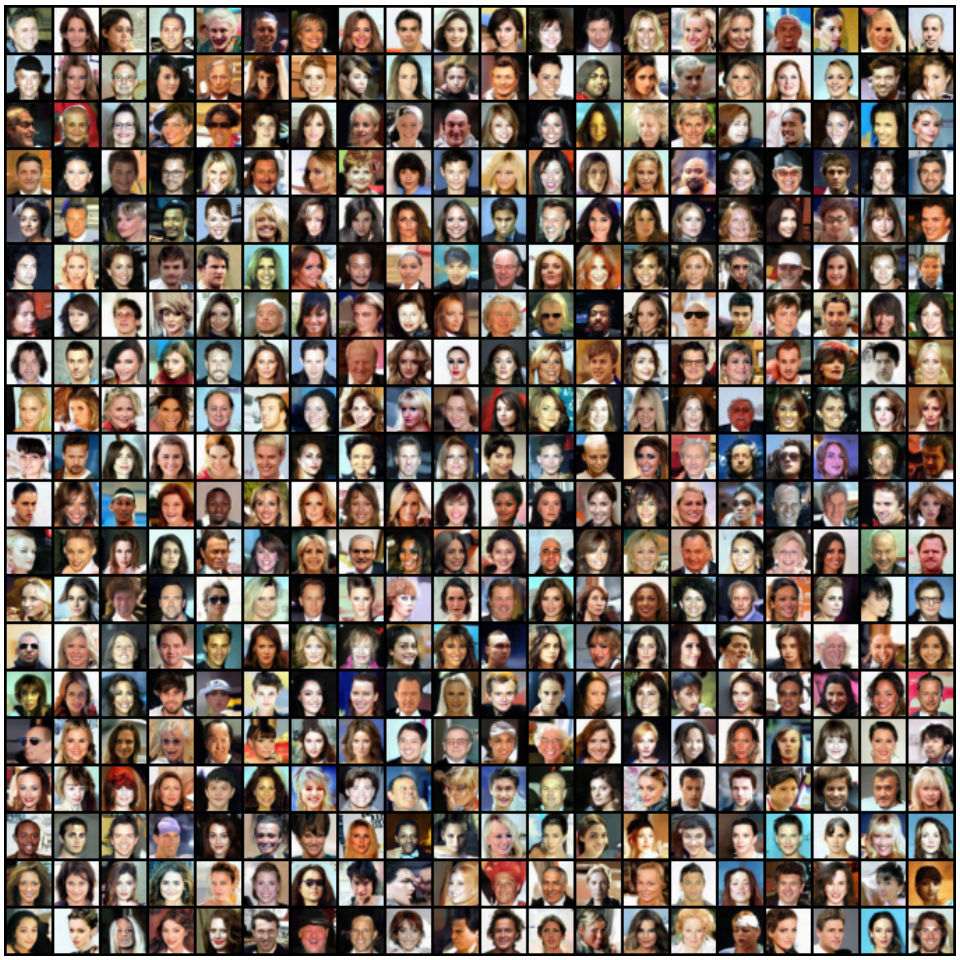

Figure 12: Uncurated samples generated by our SB-FBSDE on resized CelebA.

Figure 13: Uncurated samples generated by our SB-FBSDE on CIFAR10.