Likelihood Training of Schrödinger Bridge Using Forward-Backward SDEs Theory

Tianrong Chen*, Guan-Horng Liu*, Evangelos A. Theodorou
Georgia Institute of Technology, USA
{tianrong.chen, ghliu, evangelos.theodorou}@gatech.edu

Abstract

Schrödinger Bridge (SB) is an entropy-regularized optimal transport problem that has received increasing attention in deep generative modeling for its mathematical flexibility compared to the Score-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 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. Our code is available at https://github.com/ghliu/SB-FBSDE.

1 Introduction

Score-based Generative Model (SGM; Song et al. (2020)) is an emerging generative model class that has achieved remarkable results in synthesizing high-fidelity data (Song & Ermon, 2020; Kong et al., 2020a; b). Like many deep generative models, SGM seeks to find 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 the score function (i.e. the gradient of the log probability density) at each time step (Hyvärinen & Dayan, 2005). This reversed process thereby defines the generation (see Fig. 1).

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 entropy-regularized optimal transport problem – seeks two optimal policies that transform back-and-forth between two arbitrary distributions in a finite horizon. The similarity between the

*Equal contribution. Order determined by coin flip. See Author Contributions section.
two problems (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 retreat to traditional SB algorithms, e.g. Iterative Proportional Fitting (IPF; 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 computation (Song 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 appearing in stochastic optimal control for solving nonlinear PDEs (Han et al., 2018) – paves an elegant way to connect the two objectives. The implication of our findings is nontrivial: It yields an exact log-likelihood expression of SB that precisely generalizes the one of SGM (Song et al., 2021) to fully nonlinear diffusion, thereby providing novel theoretical connections between the two model classes. Algorithmically, our framework suggests rich training procedures that resemble the joint optimization for diffusion flow-based models (Zhang & Chen, 2021) or more traditional IPF approaches (Kullback, 1968; De Bortoli et al., 2021). This allows one to marry the best of both worlds by improving the SB training with e.g. a SGM-inspired Langevin corrector (Song & Ermon, 2019). The resulting method, SB-FBSDE, generates encouraging images on MNIST, CelebA, and CIFAR10 and outperforms prior optimal transport models by a large margin.

Our method differs from the concurrent SB methods (De Bortoli et al., 2021; Vargas et al., 2021) in various aspects. First, while both prior methods rely on solving SB with mean-matching regression, our SB-FBSDE instead utilizes a divergence-based objectives (see §3.2). Secondly, neither of the prior methods focuses on log-likelihood training, which is the key finding in SB-FBSDE to bridge connections to SGM and adopt modern training improvements. Indeed, due to the difference in the underlying SDE classes, their connections to SGM can only be made after time discretization by carefully choosing each step size (De Bortoli et al., 2021). In contrast, our theoretical connection is derived readily in continuous-time; hence unaffected by the choice of numerical discretization.

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 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.
- We show that the resulting method – named SB-FBSDE – outperforms previous optimal transport-inspired baselines on synthesizing high-fidelity images and is comparable to other existing models.

**Notation.** We denote \( p_t \) as the marginal density driven by some SDE process \( X(t) \) until the time step \( t \in [0, T] \). The time direction is aligned throughout this article such that \( p_t \) and \( p_T \) respectively correspond to the data and prior distributions. The gradient, divergence, and Hessian of a function \( f(x) \), where \( x \in \mathbb{R}^n \), will be denoted as \( \nabla_x f \in \mathbb{R}^n \), \( \nabla_x \cdot f \in \mathbb{R} \), and \( \nabla_x^2 f \in \mathbb{R}^{n \times n} \).

## 2 Preliminaries

### 2.1 Score-based Generative Model (SGM)

Given a data point \( X_0 \in \mathbb{R}^n \) sampled from an unknown data distribution \( p_{\text{data}} \), SGM first progressively diffuses the data towards random noise with the following forward SDE:

\[
\frac{dX_t}{dt} = f(t, X_t)dt + g(t)dW_t, \quad X_0 \sim p_{\text{data}},
\]

---

1. We adopt the recent advance in SB theory (Caluya & Haldar, 2021) that extends classical SB models (used in prior works) to the exact SDE class appearing in SGM. See Appendices A and C for more details.
where \( f(\cdot, t) : \mathbb{R}^n \rightarrow \mathbb{R}^n, g(t) \in \mathbb{R}, \) and \( W_t \in \mathbb{R}^n \) are 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 \( T \). Reversing (1) yields another SDE\(^2\) that traverses backward in time (Anderson, 1982):

\[
\begin{align*}
\text{d}X_t &= [f - g^2 \nabla_x \log p_T^{(1)}(X_t)] \text{d}t + g \text{d}W_t, \quad X_T \sim p_T^{(1)}, \quad (2)
\end{align*}
\]

where \( p_T^{(1)} \) corresponds to the marginal density of SDE (1) at time \( t \), and \( \nabla_x \log p_T^{(1)} \) is known as the \textit{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 \( \nabla_x \log p_T^{(1)}(X_t | X_0 = x_0) \equiv \nabla_x \log p_t(x_0 | x) \) admits an analytic solution at any time \( t \). Hence, SGM proposes to train a parameterized score network \( s(t, x; \theta) \) by regressing its outputs to the ground-truth values, \textit{i.e.} \( \mathbb{E}[\lambda(t) | s(t, X_t; \theta) - \nabla_x \log p_t(x_0 | x)]^2 \), where the expectation is taken over the SDE (1). In practice, \( \lambda(t) \) is some hand-designed weighting function that largely affects the performance. Recent works (Song et al., 2021; Huang et al., 2021) have shown that the log-likelihood of SGM, despite being complex, can be lower-bounded as follows:

\[
\begin{align*}
\log p_{\text{SGM}}(x_0; \theta) &\geq \mathcal{L}_{\text{SGM}}(x_0; \theta) = \mathbb{E}[\log p_T(X_T)] - \int_0^T \mathbb{E}\left[ \frac{1}{2} g^2 \| s_t \|^2 + \nabla_x \cdot (g^2 s_t - f) \right] \text{d}t, \quad (3) \\
&= \mathbb{E}[\log p_T(X_T)] - \int_0^T \mathbb{E}\left[ \frac{1}{2} g^2 \| s_t - \nabla_x \log p_t(x_0 | x)^2 - \frac{1}{2} g \nabla_x \log p_t(x_0 | x)^2 - \nabla_x \cdot f \right] \text{d}t,
\end{align*}
\]

where \( s_t \equiv s(t, x; \theta) \) and the expectation is taken over the SDE (1) given a data point \( X_0 = x_0 \). This objective (3) suggests a principled choice of \( \lambda(t) := g(t)^2 \). After training, SGM simply substitutes the score function with the learned score network \( s(t, x; \theta) \) to generate data from \( p_{\text{prior}} \).

\[
\begin{align*}
\text{d}X_t &= [f - g^2 s(t, X_t; \theta)] \text{d}t + g \text{d}W_t, \quad X_T \sim p_{\text{prior}}. \quad (4)
\end{align*}
\]

It is important to notice that \( p_{\text{prior}} \) needs not equal \( p_T^{(1)} \) in practice, and the approximation is close only through a careful design of (1). Notably, 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 lead to unstable training (Nichol & Dhariwal, 2021). In contrast, Schrödinger Bridge considers a more flexible framework for designing the forward diffusion that requires minimal manipulation.

### 2.2 Schrödinger Bridge (SB)

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

\[
\min_{\mathcal{Q} \in \mathcal{P}(p_{\text{data}}, p_{\text{prior}})} D_{\text{KL}}(\mathcal{Q} \parallel \mathcal{P}), \quad (5)
\]

where \( \mathcal{Q} \in \mathcal{P}(p_{\text{data}}, p_{\text{prior}}) \) belongs to a set of path measure with \( p_{\text{data}} \) and \( p_{\text{prior}} \) as its marginal densities at \( t = 0 \) and \( T \). On the other hand, \( \mathcal{P} \) 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; Chen et al. (2021); Pavon & Wakolbinger (1991); Caluya & Halder (2021)). Let \( \Psi(t, x) \) and \( \hat{\Psi}(t, x) \) be the solutions to the following PDEs:

\[
\begin{align*}
\frac{\partial \Psi}{\partial t} &= -\nabla_x \Psi f - \frac{1}{2} \text{Tr}(g^2 \nabla_x^2 \Psi) \\
\frac{\partial \hat{\Psi}}{\partial t} &= -\nabla_x \cdot (\hat{\Psi} f) + \frac{1}{2} \text{Tr}(g^2 \nabla_x^2 \hat{\Psi}) \quad \text{s.t. } \Psi(0, \cdot) \Psi(0, \cdot) = p_{\text{data}}, \quad \Psi(T, \cdot) \hat{\Psi}(T, \cdot) = p_{\text{prior}} \quad (6)
\end{align*}
\]

Then, the solution to the optimization (5) can be expressed by the path measure of the following forward (7a), or equivalently backward (7b), SDE:

\[
\begin{align*}
\text{d}X_t &= [f + g^2 \nabla_x \log \Psi(t, X_t)] \text{d}t + g \text{d}W_t, \quad X_0 \sim p_{\text{data}}, \quad (7a) \\
\text{d}X_t &= [f - g^2 \nabla_x \log \hat{\Psi}(t, X_t)] \text{d}t + g \text{d}W_t, \quad X_T \sim p_{\text{prior}}, \quad (7b)
\end{align*}
\]

where \( \nabla_x \log \Psi(t, X_t) \) and \( \nabla_x \log \hat{\Psi}(t, X_t) \) are the optimal forward and backward drifts for SB.

---

\(^2\)Hereafter, we will sometimes drop \( f = f(t, x_t) \) and \( g = g(t) \) 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.

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

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 ($p_{data} \Rightarrow p_{prior}$) with multiple stages, $p_{data} \Rightarrow p_{middle} \Rightarrow p_{prior}$, so that each segment admits an analytic solution (Wang et al., 2021), or consider the following half-bridge ($p_{data} \leftarrow p_{prior}$ vs. $p_{data} \rightarrow p_{prior}$) optimization (De Bortoli et al., 2021; Vargas et al., 2021),

$$Q^{(1)} := \arg\min_{Q \in P(\cdot, p_{prior})} D_{KL}(Q \parallel Q^{(0)}), \quad Q^{(0)} := \arg\min_{Q \in P(p_{data})} D_{KL}(Q \parallel Q^{(1)})$$

which can be solved with IPF algorithm (Kullback, 1968) starting from $Q^{(0)} := P$. 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 similar structures. From the stochastic control perspective, these SDEs belong to the class of control-affine SDEs with additive noise:

$$dX_t = A(t, X_t)dt + B(t, X_t)u(t, X_t)dt + C(t)\, dW_t. \quad (8)$$

It is clear that the control-affine SDE (8) includes all SDEs (1,2,4,7) appearing in §2 by considering $\langle A, B, C \rangle := (f, I, g)$ and different interpretations of the control variables $u(t, X_t)$. This implies that the optimization processes of both SGM and SB can be aligned 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 $\nabla_a \log p_t(x_0)$ in SGM or $\nabla_a \log \hat{\Psi}$ in SB, that minimizes some objectives, (3) vs. (5), while subjected to some control-affine SDEs, (1,2) vs. (7). In what follows, we will show that a specific mathematical methodology in nonlinear SOC literature – called Forward-Backward SDEs theory (FBSDEs; see Ma et al. (1999)) – links the optimality condition of SB (6) to the log-likelihood objectives of SGM (3). All proofs are left to Appendix B.

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 the following connection related to our problem.

**Lemma 2** (Nonlinear Feynman-Kac; Exarchos & Theodorou (2018)). Consider the coupled SDEs

$$\begin{align*}
\begin{cases}
 dX_t = f(t, X_t)dt + G(t, X_t)dW_t, & X_0 = x_0 \\
 dY_t = -h(t, X_t, Y_t, Z_t)dt + Z(t, X_t)^T dW_t, & Y_T = \varphi(X_T)
\end{cases}
\end{align*} \quad (9a)$$

$\begin{align*}
\begin{cases}
 dY_t = g(t, Y_t, Z_t, Y_{\Sigma})dt + Z(t, Y_t)^T dW_t, & Y_0 = y_0 \\
 dX_t = -h(t, Y_t, Z_t, X_{\Sigma})dt + X(t, Y_t)^T dW_t, & X_T = \Psi(Y_T)
\end{cases}
\end{align*} \quad (9b)

\text{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.}
where the functions $f$, $G$, $h$, and $\varphi$ satisfy proper regularity conditions\footnote{Yong & Zhou (1999); Kobylanski (2000) require $f$, $G$, $h$, and $\varphi$ to be continuous, $f$ and $G$ to be uniformly Lipschitz in $x$, and $h$ to satisfy quadratic growth condition in $z$.} so that there exists a pair of unique strong solutions satisfying (9). Now, consider the following second-order parabolic PDE and suppose $v(t, x) \equiv v$ is once continuously differentiable in $t$ and twice in $x$, i.e. $v \in C^{1,2}$,

\[
\frac{\partial v}{\partial t} + \frac{1}{2} \text{Tr}(\nabla^2_x v \, G G^T) + \nabla_x v^T f + h(t, x, v, G^T \nabla_x v) = 0, \quad v(T, x) = \varphi(x),
\]

then the solution to (9) coincides with the solution to (10) along paths generated by the forward SDE (9a) almost surely, i.e., the following stochastic representation (known as the nonlinear Feynman-Kac relation) is valid:

\[
v(t, X_t) = Y_t \quad \text{and} \quad G(t, X_t)^T \nabla_x v(t, X_t) = Z_t.
\]

Lemma 2 states that solutions to a certain class of nonlinear (via the function $h$ in (10)) PDEs can be represented by solutions to a set of forward-backward SDEs (9) through the transformation (11), and this relation can be extended to the viscosity case (Pardoux & Peng (1992); see also Appendix B). Note that $Y_t$ is the solution to the backward SDE (9b) whose randomness is driven by the forward SDE (9a). Indeed, it is clear from (11) that $Y_t$ (hence also $Z_t$) is a time-varying function of $X_t$. Since the $v$ appearing in the nonlinear Feynman-Kac relation (11) takes the random vector $X_t$ as its argument, $v(t, X_t)$ shall also be understood as a random variable. Finally, it is known that the original (deterministic) PDE solution $v(t, x)$ can be recovered by taking conditional expectation, i.e.

\[
v(t, x) = \mathbb{E}_{X_t \sim (9a)}[Y_t | X_t = x] \quad \text{and} \quad G(t, x)^T \nabla_x v(t, x) = \mathbb{E}_{X_t \sim (9a)}[Z_t | X_t = x].
\]

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 (11) under proper conditions, with the optimal control given in the form of $Z_t$. 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)). Consider the following set of coupled SDEs,

\[
\begin{align*}
\text{d}X_t &= (f + g Z_t) \, \text{d}t + g \, \text{d}W_t \quad \tag{13a}
\end{align*}
\]

\[
\begin{align*}
\text{d}Y_t &= \frac{1}{2} Z_t^T Z_t \, \text{d}t + Z_t^T \, \text{d}W_t \quad \tag{13b}
\end{align*}
\]

\[
\begin{align*}
\text{d}\hat{Y}_t &= \left( \frac{1}{2} \hat{Z}_t^T \hat{Z}_t + \nabla_x \cdot (g \hat{Z}_t - f) + \hat{Z}_t^T \hat{Z}_t \right) \, \text{d}t + \hat{Z}_t^T \, \text{d}W_t \quad \tag{13c}
\end{align*}
\]

where $f$ and $g$ satisfy the same regularity conditions in Lemma 2 (see Footnote 4), and the boundary conditions are given by $X(0) = x_0$ and $Y_T + \hat{Y}_T \equiv \log \rho_{\text{prior}}(X_T)$. Suppose $\Psi, \hat{\Psi} \in C^{1,2}$, then the nonlinear Feynman-Kac relations between the FBSDEs (13) and PDEs (6) are given by

\[
\begin{align*}
Y_t &\equiv Y(t, X_t) = \log \Psi(t, X_t), \quad \hat{Z}_t \equiv \hat{Z}(t, X_t) = g \nabla_x \log \Psi(t, X_t), \\
\hat{Y}_t &\equiv \hat{Y}(t, X_t) = \log \hat{\Psi}(t, X_t), \quad \hat{Z}_t = \hat{Z}(t, X_t) = g \nabla_x \log \hat{\Psi}(t, X_t).
\end{align*}
\]

Furthermore, $(Y_t, \hat{Y}_t)$ obey the following relation:

\[
Y_t + \hat{Y}_t = \log \rho_{\text{SB}}(X_t).
\]

The FBSDEs for SB (13) share a similar forward-backward structure as in (9), where (13a) and (13b,13c) respectively represent the forward and backward SDEs. One can verify that the forward SDE (13a) coincides with the optimal forward SDE (7a) with the substitution $Z_t = g \nabla_x \log \Psi$. In other words, these FBSDEs provide a local representation of $\log \Psi$ and $\log \hat{\Psi}$ evaluated on the optimal path governed by (7a). Since $Z_t$ and $\hat{Z}_t$ can be understood as the forward/backward policies, in a similar spirit of policy-based methods (Pereira et al., 2020; Schulman et al., 2015), that guide the SDE processes of SB, they sufficiently characterize the SB model. Hence, our next step is to derive a proper training objective to optimize these policies.
3.2 Log-likelihood Computation of SB

Theorem 3 has an important implication: It suggests that given a path sampled from the forward SDE (13a), the solutions to the backward SDEs (13b,13c) at $t = 0$ provide an unbiased estimation of the log-likelihood of the data point $x_0$, i.e.

$$
\mathbb{E} \left[ Y_0 + \hat{Y}_0 | X_0 = x_0 \right] = \log p^{SB}_0(x_0) = \log p_{dat}(x_0),
$$

where $X_t$ is sampled from (13a). We now state our main result, which makes this observation formal:

**Theorem 4** (Log-likelihood of SB model). Given the solution satisfying the FBSDE system in (13), the log-likelihood of the SB model $(Z_t, \hat{Z}_t)$, at a data point $x_0$, can be expressed as

$$
\log p^{SB}_0(x_0) = \mathbb{E} \left[ \log p_T(X_T) \right] - \int_0^T \mathbb{E} \left[ \frac{1}{2} \| Z_t \|^2 + \frac{1}{2} \| \hat{Z}_t - g \nabla_x \log p^{SB}_t + Z_t \|^2 
\right.
\left. - \frac{1}{2} \| g \nabla_x \log p^{SB}_t - Z_t \|^2 - \nabla_x \cdot f \right] dt, 
\tag{15}
$$

$$
= \mathbb{E} \left[ \log p_T(X_T) \right] - \int_0^T \mathbb{E} \left[ \frac{1}{2} \| Z_t \|^2 + \frac{1}{2} \| \hat{Z}_t \|^2 + \nabla_x \cdot (g \hat{Z}_t - f) + \hat{Z}_t \cdot Z_t \right] dt, \tag{16}
$$

where the expectation is taken over the forward SDE (13a) with the initial condition $X_0 = x_0$.

Similar to (3), Theorem 4 suggests a parameterized lower bound to the log-likelihoods, i.e.

$$
\log p^{SB}_0(x_0) \geq L_{SB}(x_0; \theta, \phi) \text{ where } L_{SB}(x_0; \theta, \phi) \text{ shares the same expression in (16) except that } Z_t \approx Z(t; x; \theta) \text{ and } \hat{Z}_t \approx \hat{Z}(t; x; \phi) \text{ are approximated with some parameterized models (e.g. DNNs). Note that } \nabla_x \log p^{SB}_t \text{ is intractable in practice for any nontrivial } (Z_t, \hat{Z}_t). \text{ Hence, we use the divergence-based objective in (16) as our training objective of both policies.}
$$

**Connection to score-based models.** Recall Fig. 2 and compare the parameterized log-likelihoods of SB (16) to SGM (3); one can verify that $L_{SB}$ collapses to $L_{SGM}$ when $(Z_t, \hat{Z}_t) := (0, g s_t)$. From the SB perspective, this occurs only when $p_T^{(1)} = p_{prior}$. Since no effort is required in the forward process to reach $p_{prior}$, the optimal forward control $Z_t$, by definition, degenerates; thereby making the backward control $\hat{Z}_t$ collapses to the score function. However, in any case when $p_T^{(1)} \neq p_{prior}$, for instance when the diffusion SDEs are improperly designed, the forward policy $Z_t$ steers the diffusion process back to $p_{prior}$, while its backward counterpart $\hat{Z}_t$ 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, Theorem 4 generalizes the log-likelihood training from SGM to SB.

**Connection to flow-based models.** Interestingly, the log-likelihood computation in Theorem 4, where we use a path $\{X_t\}_{t \in [0,T]}$ sampled from a data point $X_0$ 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., 2020; 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 $\forall t$, $p_t^{(1)} \equiv p_t^{(7)} \equiv p_t^{SB}$.

$$
\text{d}X_t = \left[ f + gZ(t, X_t) - \frac{1}{2} g (Z(t, X_t) + \hat{Z}(t, X_t)) \right] dt \tag{17}
$$

One can verify (see Remark 10 in §B) that computing the log-likelihood of this ODE model (17) using flow-based training techniques indeed recovers the training objective of SB derived in (16).

3.3 Practical Implementation

In this section, we detail the implementation of our FBSDE-inspired SB model, named **SB-FBSDE**.

**Training process.** We treat the log-likelihood in (16) as our training objective, where the divergence can be estimated efficiently following Hutchinson (1989). This immediately distinguishes SB-FBSDE from prior SB models (De Bortoli et al., 2021; Vargas et al., 2021), which instead rely on
Our training process is summarized in Alg. 1. While the joint training scheme in Alg. 2 resembles recent diffusion flow-based models (Zhang & Chen, 2021), the alternate training in Alg. 3 relates to the classical IPF (De Bortoli et al., 2021), despite differing in the underlying objectives. Empirically, the joint training scheme can converge faster yet at the cost of introducing memory complexity. We highlight these flexible training procedures arising from the unified viewpoint provided in Theorem 4. Hereafter, we refer to each cycle, i.e. 2K training steps, in Alg. 3 as a training stage of SB-FBSDE.

**Generative process.** While the generative processes for SB can be performed as simply as propagating (7b) given the trained policy \( \hat{Z}(\cdot, \cdot; \phi) \), it has been constantly observed that adopting Langevin sampling to the generative process greatly improves performance (Song et al., 2020). This procedure, often referred to as the Langevin corrector, requires knowing the score function \( \nabla_x \log p_t \). For SB, we can estimate its value by recalling (see §2.2) that \( \frac{\nabla}{\nabla_t} = g \nabla_x \log p_t^{SB} \). This results in the following predictor-corrector sampling procedure (see Alg. 4 in Appendix D for more details).

\[
\text{Predict step: } \quad X_t \leftarrow X_t + g Z_t \Delta t \sqrt{2} \Delta t \epsilon \\
\text{Correct step: } \quad X_t \leftarrow X_t + g Z_t \Delta t + \sigma_t \epsilon \\
\] (20)  (21)

where \( \epsilon \sim \mathcal{N}(0, I) \) and \( \sigma_t \) is the pre-specified noise scales (see (59) in Appendix D).

**Limitations & efficiency.** The main computational burden of our method comes from the computation of the divergence and maintaining two distinct networks. Despite it typically increases the memory by 2–2.5 times compared to SGM, we empirically observe that the divergence-based training converges much faster per iteration than standard regression. As a result, SB-FBSDE admits comparable training time (+6.8% in our CIFAR10 experiment) compared to SGM, yet with a substantially fewer sampling time (-80%) due to adopting nonlinear SDEs.

---

5In fact, their regression targets may be recovered from (15) under proper transformation; see Appendix C.
4 EXPERIMENTS

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 $\tilde{Z}(\cdot, \cdot; \phi)$. **Bottom**: Diffusion ($p_{\text{data}} \rightarrow p_{\text{prior}}$) process with the forward vector field $Z(\cdot, \cdot; \theta)$.

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

**Setups.** We testify SB-FBSDE on two toy datasets and three image datasets, i.e. MNIST, CelebA, and CIFAR10. $p_{\text{prior}}$ is set to a zero-mean Gaussian whose variance varies for each task and can be computed according to Song & Ermon (2020). We parameterize $Z(\cdot, \cdot; \theta)$ and $\tilde{Z}(\cdot, :: \phi)$ with residual-based networks for toy datasets and consider Unet (Ronneberger et al., 2015) and NCSN++ (Song et al., 2020) 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. $f := 0$; see Song et al. (2020)) 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 D.

**Toy datasets.** We first validate our joint optimization (i.e. Alg 2) on generating a mixture of Gaussian and checkerboard (adopted from Grathwohl et al. (2018)) as the representatives of continuous and discontinuous distributions. Figure 3 shows how the learned policies, i.e. $Z(\cdot, :: \theta)$ and $\tilde{Z}(\cdot, :: \phi)$, construct the vector fields that progressively transport samples back-and-forth between $p_{\text{prior}}$ and $p_{\text{data}}$. The 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. We highlight this as a distinct feature arising from SB models.

**Image datasets.** Next, we validate our alternate training (i.e. Alg 3) 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 Appendix E. Regarding the quantitative evaluation, Table 1 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 steps.

---

We follow a similar setup of prior SB models (De Bortoli et al., 2021) and resize the image size to 32.
Table 1: 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.

<table>
<thead>
<tr>
<th>Model Class</th>
<th>Method</th>
<th>NLL ↓</th>
<th>FID ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimal Transport</td>
<td>SB-FBSDE (ours)</td>
<td>2.96</td>
<td>3.01</td>
</tr>
<tr>
<td></td>
<td>DOT (Tanaka, 2019)</td>
<td>-</td>
<td>15.78</td>
</tr>
<tr>
<td></td>
<td>Multi-stage SB (Wang et al., 2021)</td>
<td>-</td>
<td>12.32</td>
</tr>
<tr>
<td></td>
<td>DGflow (Ansari et al., 2020)</td>
<td>-</td>
<td>9.63</td>
</tr>
<tr>
<td>SGMs</td>
<td>SDE (deep, sub-VP; Song et al. (2020))</td>
<td>2.99</td>
<td>2.92</td>
</tr>
<tr>
<td></td>
<td>ScoreFlow (Song et al., 2021)</td>
<td>2.74</td>
<td>5.7</td>
</tr>
<tr>
<td></td>
<td>VDM (Kingma et al., 2021)</td>
<td>2.49</td>
<td>4.00</td>
</tr>
<tr>
<td></td>
<td>LSGM(Vahdat et al., 2021)</td>
<td>3.43</td>
<td>2.10</td>
</tr>
</tbody>
</table>

Figure 5: Validation of our SB-FBSDE on MNIST and CelebA (resized) datasets. Our method generates terminal distributions that are much closer to the desired density \( p_{\text{prior}} \) compared to SGM.

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

Validity of SB forward diffusion. Our theoretical analysis in §3.2 suggests that the forward policy \( Z(\cdot; \theta) \equiv Z_\theta \) plays an essential role in governing samples towards \( p_{\text{prior}} \). Here, we validate this conjecture by computing the KL divergence between the terminal distribution induced by \( Z_\theta \), i.e. \( \mathbb{KL}(p_{\text{prior}}; Z_\theta) \), and the designated prior \( p_{\text{prior}} \). We refer readers to Appendix D for the actual computation. Figure 5 reports these values over MNIST training. For both degenerate (\( f := 0 \)) and linear (\( f := \alpha_t X_t \)) base drifts, our SB-FBSDE generates terminal distributions that are much closer to \( p_{\text{prior}} \). 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 \( Z_\theta \) gradually shortens the KL gap to \( p_{\text{prior}} \), thereby providing a better forward diffusion for training the backward policy.

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, their propagation can be subjected to discretization errors accumulated over time. These Langevin steps thereby help re-distributing the samples at each time step towards the desired density \( p_{\text{prior}} \). We emphasize this improvement as the benefit gained from applying modern generative training techniques based on the solid connection between SB and SGM.

5 Conclusion

In this work, we present a novel computational framework, grounded on Forward-Backward SDEs theory, for computing the log-likelihood of Schrödinger Bridge (SB) – a recently emerging model that adopts entropy-regularized optimal transport for generative modeling. Our findings provide new theoretical insights by generalizing previous theoretical results for Score-based Generative Model, and facilitate applications of modern generative training for SB. We validate our method on various image generative tasks, e.g. MNIST, CelebA, and CIFAR10, showing encouraging results in synthesizing high-fidelity samples while retaining the rigorous mathematical framework.
ACKNOWLEDGMENTS

The authors would like to thank Ioannis Exarchos and Oswin So for their generous involvement and helpful supports during the rebuttal. The authors would also like to thank Marcus A Pereira and Ethan N Evans for their participation and kind discussion in the early stage of project exploration.

AUTHOR CONTRIBUTIONS

The original idea of solving the PDE optimality of SB with FBSDEs theory was initiated by Tianrong. Later, Guan derived the main theories (i.e. Theorem 3, 4, 11, and Corollary 5) presented in Section 3.1, 3.2 and Appendix B with few helps from Tianrong. Tianrong designed the practical algorithms (e.g. stage-wise optimization and Langevin-corrector) in Section 3.3 and conducted most experiments with few helps from Guan. Guan wrote the main paper except for Section 4, which were written by both Tianrong and Guan. Both Guan and Tianrong contributed to code development.

REPRODUCIBILITY STATEMENT

Our training algorithms are detailed in Alg. 1, 2, and 3, with the training objectives given in the same section (see (16, 18, 19)). Other implementation details (e.g. data pre-processing) are left in Appendix D. This shall provide sufficient information for readers of interests to reproduce our results. As we strongly believe in the merit of open sourcing, we intend to release our implementation upon publication. On the theoretical side, all proofs are left to Appendix B due to space constraint. We provide the assumptions in the same section.

REFERENCES


A INTRODUCTION OF SCHRÖDINGER BRIDGE

In this subsection, we provide a brief review for Schrödinger Bridge (SB) and some reasonings for Theorem 1. The SB problem, at its classical form, considers the following optimization (Dai Pra, 1991; Pavon & Wakolbinger, 1991),

\[
\min_{u \in U} \mathbb{E} \left[ \int_0^T \frac{1}{2} \|u(t, X_t)\|^2 \right] \\
\text{s.t.} \begin{align*}
&\frac{dX_t}{dt} = u(t, X_t) dt + \sqrt{2\epsilon} dW_t \\
&X_0 \sim p_0(X), \quad X_T \sim p_T(X),
\end{align*}
\]

where \( U := \{ u : [0, T] \times \mathbb{R}^n \to \mathbb{R}^n | (u, u) < \infty \} \). The optimization (22) characterizes a standard stochastic optimal control (SOC) programming with energy (i.e., \( \frac{1}{2} \| u \|^2 \)) minimization except with an additional terminal boundary condition. The optimality conditions to (22) are given by

\[
\begin{align*}
\frac{\partial \psi}{\partial t} &= -\frac{1}{2} \| \nabla_x \psi \|^2 - \epsilon \Delta \psi, \\
\frac{\partial p^*}{\partial t} &= \nabla_x \cdot (p^* \nabla_x \psi) + \epsilon \Delta p^*,
\end{align*}
\]

where \( \psi(t, x) \in C^{1,2} \) is known as the value function in SOC literature and \( p^*(t, x) \in C^{1,2} \) is the associated optimal marginal density. \( \Delta \) denotes the Laplace operator. Equations (23a) and (23b) are respectively the Kolmogorov’s backward and forward PDEs, also known as Hamilton-Jacobi-Bellman and Fokker-Planck equations. The SB system can be obtained by applying the Hopf-Cole (Hopf, 1950; Cole, 1951) transformation \( (\psi, p^*) \mapsto (\hat{\psi}, \hat{p}) \),

\[
\begin{align*}
\frac{\partial \hat{\psi}}{\partial t} &= -\epsilon \Delta \hat{\psi} \quad \text{s.t.} \quad \hat{\psi}(0, \cdot) = \hat{\psi}(T, \cdot) = p_0, \quad \hat{\psi}(T, \cdot) = p_T.
\end{align*}
\]

In this work, we consider a recent generalization of (22) and (24) to an SDE class with nonlinear drift, affine control, and time-varying diffusion. We synthesize their results below.

**Theorem 6** (SB optimality; Caluya & Halder (2021)). Consider the following optimization

\[
\min_u \mathbb{E} \left[ \int_0^T \frac{1}{2} \|u(t, X_t)\|^2 \right] \\
\text{s.t.} \begin{align*}
&\frac{dX_t}{dt} = [f(t, X_t) + g(t) u(t, X_t)] dt + \sqrt{2\epsilon} g(t) dW_t \\
&X_0 \sim p_0(X), \quad X_T \sim p_T(X)
\end{align*}
\]

where \( g(t) \) is uniformly lower-bounded and \( f(t, X_t) \) satisfies Lipschitz conditions with at most linear growth in \( x \). Then, the Hopf-Cole transformation to (25) becomes

\[
\begin{align*}
\frac{\partial \hat{\psi}}{\partial t} &= -\nabla_x \hat{\psi} T f - \epsilon \text{Tr}(g^2 \nabla_x^2 \hat{\psi}) \\
\frac{\partial \hat{p}}{\partial t} &= -\nabla_x \cdot (\hat{p} \nabla_x \hat{\psi}) + \epsilon \text{Tr}(g^2 \nabla_x^2 \hat{\psi})
\end{align*}
\]

with the same boundary conditions \( \hat{\psi}(0, \cdot) = \hat{\psi}(T, \cdot) = p_0, \hat{\psi}(T, \cdot) = p_T. \) The optimal control to (25) is thereby given by

\[
u^*(t, X_t) = 2\epsilon g(t) \nabla_x \log \hat{\psi}(t, X_t).
\]

**Proof.** See Section III and Theorem 2 in Caluya & Halder (2021). \( \square \)

Theorem 6 is particularly attractive to us since its SDE corresponds exactly to the one appearing in score-based generative models. One can recover Theorem 1 by

(i) Following Pavon & Wakolbinger (1991), we know that the objective in (25) is equivalent to \( D_{KL}(Q \| P) \) by an application of Girsanov’s Theorem.

14
We first restate the celebrated Itô lemma, which is known as the extension of the chain rule of ordinary calculus to the stochastic setting. It relies on the fact that \( \nabla \cdot \nabla \) keeps the expansion up to \( O(d) \).}

Finally, reversing the SDE (7a) from forward to backward following Anderson (1982),

\[
dX_t = \left[ f + g^2 \nabla_x \log \Psi(t, X_t) - g^2 \nabla_x \log p^{\text{SB}}_t \right] dt + g \, dW_t, \tag{28}
\]

and recalling the factorization principle, \( \log p^{\text{SB}}_t(\cdot) = \log \Psi(t, \cdot) + \log \hat{\Psi}(t, \cdot) \), from Equation (4.15) in Chen et al. (2021) yield the optimal backward SDE in (7b).

### B Proofs and Remarks in Section 3

In this section, we provide proofs for all of our theorems. We following the same notation by denoting \( p^\text{SDE}_t(X_t) \) as the marginal density driven by some SDE process \( X(t) \equiv X_t \) until the time step \( t \in [0, T] \). Gradient and Hessian of a function \( f(x) \), where \( x \in \mathbb{R}^n \), will respectively be denoted as \( \nabla_x f \in \mathbb{R}^n \) and \( \nabla^2_x f \in \mathbb{R}^{n \times n} \). Divergence and Laplace operators will respectively be denoted as \( \nabla \cdot \) and \( \Delta \). Note that \( \Delta = \nabla \cdot \nabla \). For notational brevity, we will only keep the subscript \( x \) for multivariate functions. Finally, \( \text{Tr}(A) \) denotes the trace of a square matrix \( A \).

We first restate the celebrated Itô lemma, which is known as the extension of the chain rule of ordinary calculus to the stochastic setting. It relies on the fact that \( dW_t^2 \) and \( dt \) are of the same scale and keeps the expansion up to \( O(dt) \).

**Lemma 7 (Itô formula; Itô (1951)).** Let \( v \in C^{1,2} \) and let \( X_t \) be the stochastic process satisfying

\[
dX_t = f(t, X_t)dt + G(t, X_t)dW_t.
\]

Then, the stochastic process \( v(t, X_t) \) is also an Itô process satisfying

\[
\frac{dv(t, X_t)}{dt} = \left[ \nabla_x v(t, X_t) \right]^T f + \frac{1}{2} \text{Tr} \left[ G^T \nabla^2_x v(t, X_t) G \right] dt + \left[ \nabla_x v(t, X_t) \right]^T G(t, X_t) dW_t.
\]

Next, the following lemma will be useful in proving Theorem 3.

**Lemma 8.** The following equality holds at any point \( x \in \mathbb{R}^n \) such that \( p(x) \neq 0 \).

\[
\frac{1}{p(x)} \text{Tr} \left( \nabla^2 p(x) \right) = \left\| \nabla \log p(x) \right\|^2 + \Delta \log p(x)
\]

**Proof.**

\[
\text{Tr} \left( \nabla^2 p(x) \right) = \Delta p(x) = \nabla \cdot \nabla p(x) = \nabla \cdot (p(x) \nabla \log p(x)) = \nabla p(x)^T \nabla \log p(x) + p(x) \Delta \log p(x) = p(x) \nabla \log p(x)^T \nabla \log p(x) + p(x) \Delta \log p(x) = p(x) \left( \left\| \nabla \log p(x) \right\|^2 + \Delta \log p(x) \right)
\]

\( \square \)

**Assumptions** Before stating our proofs, we provide the assumptions used throughout the paper. These assumptions are adopted from stochastic analysis for SGM (Song et al., 2021; Yong & Zhou, 1999; Anderson, 1982), SB (Caluya & Halder, 2021), and FBSDE (Exarchos & Theodorou, 2018; Gorodetsky et al., 2015).

1. \( p_{\text{prior}}, p_{\text{data}} \in C^2 \) with finite second-order moment.
2. \( f \) and \( g \) are continuous functions, and \( |g(t)|^2 > 0 \) is uniformly lower-bounded w.r.t. \( t \).
3. \( \forall t \in [0, T] \), we have \( f(t, x), \nabla_x \log p_t(x), \nabla_x \log \Psi(t, x), \nabla_x \log \hat{\Psi}(t, x), Z(t, x; \theta), \) and \( \hat{Z}(t, x; \phi) \) Lipschitz and at most linear growth w.r.t. \( x \).
(iv) \( \Psi, \hat{\Psi} \in C^{1,2} \), \( h \), and \( \varphi \) are continuous functions. \( h \) satisfies quadratic growth w.r.t. \( x \) uniformly in \( t \).

(v) \( \exists k > 0 : p_t^{SB}(x) = \mathcal{O}(\exp-\|x\|^2) \) as \( x \to \infty \).

Assumptions (i) (ii) (iii) are standard conditions in stochastic analysis to ensure the existence-uniqueness of the SDEs; hence also appear in SGM analysis (Song et al., 2021). Assumption (iv) allows applications of Itô formula and properly defines the backward SDE in FBSDE theory. Finally, assumption (v) assures the exponential limiting behavior when performing integration by parts.

Now, let us begin the proofs of Theorem 3, 4, and Corollary 5.

**Theorem 3** (FBSDEs to SB optimality (6)). Consider the following set of coupled SDEs,

\[
\begin{align*}
\text{d}X_t &= (f + gZ_t)\,\text{d}t + g\text{d}W_t \\
\text{d}Y_t &= \frac{1}{2}Z_t^T Z_t \text{d}t + Z_t^T \text{d}W_t \\
\text{d}\hat{Y}_t &= \left(\frac{1}{2}Z_t^T Z_t + \nabla_x (g\hat{Z}_t - f) + \hat{Z}_t^T Z_t\right) \,\text{d}t + \hat{Z}_t^T \text{d}W_t
\end{align*}
\]

where \( f \) and \( g \) satisfy the same regularity conditions in Lemma 2 (see Footnote 4), and the boundary conditions are given by \( X(0) = x_0 \) and \( Y_T + \hat{Y}_T = \log p_{\text{prior}}(X_T) \). Suppose \( \Psi, \hat{\Psi} \in C^{1,2} \), then nonlinear Feynman-Kac relations between the FBSDEs (13) and PDEs (6) are given by

\[
\begin{align*}
Y_t &\equiv Y(t, X_t) = \log \Psi(t, X_t), \quad Z_t \equiv Z(t, X_t) = g\nabla_x \log \Psi(t, X_t), \\
\hat{Y}_t &\equiv \hat{Y}(t, X_t) = \log \hat{\Psi}(t, X_t), \quad \hat{Z}_t \equiv \hat{Z}(t, X_t) = g\nabla_x \log \hat{\Psi}(t, X_t).
\end{align*}
\]

Furthermore, \((Y_t, \hat{Y}_t)\) obey the following relation:

\( Y_t + \hat{Y}_t = \log p_t^{SB}(X_t) \).

**Proof.** Similar to how the original nonlinear Feynman-Kac (i.e. Lemma 2) can be carried out by an application of Itô lemma (Ma et al., 1999). We can apply Itô lemma 7 to the stochastic process \( \log \Psi(t, X_t) \) w.r.t. the optimal forward SDE (7a).

\[
\text{d} \Psi = \frac{\partial \log \Psi}{\partial t} \,\text{d}t + \left[ \nabla_x \log \Psi^T (f + g^2 \nabla_x \log \Psi) + \frac{1}{2} g^2 \text{Tr} \left[ \nabla_x^2 \log \Psi \right] \right] \,\text{d}t + \left[ g \nabla_x \log \Psi^T \right] \text{d}W_t.
\]

From the PDE dynamics (6), we know that

\[
\frac{\partial \log \Psi}{\partial t} = \frac{1}{\Psi} \left( -\nabla_x \Psi^T f - \frac{1}{2} \text{Tr} \left( g^2 \nabla_x^2 \Psi \right) \right) = -\nabla_x \log \Psi^T f - \frac{1}{2} g^2 \text{Tr} \left( \frac{1}{\Psi^2} \nabla_x^2 \Psi \right).
\]

The first term in the RHS can be readily canceled out with the related \( f \)-term in (32). The second term can also be canceled out using the fact that \( \nabla_x \log \Psi = \frac{1}{\Psi} \nabla_x \Psi - \frac{1}{\Psi^2} \nabla_x \Psi \nabla_x \Psi^T \). Hence, we are left with

\[
\text{d} \log \Psi = \left[ g \nabla_x \log \Psi \right] \,\text{d}t + g \nabla_x \log \Psi^T \,\text{d}W_t.
\]

Likewise, applying Itô lemma to \( \log \hat{\Psi}(t, X_t) \), where \( X_t \) follows the SDE in (7a),

\[
\text{d} \log \hat{\Psi} = \frac{\partial \log \hat{\Psi}}{\partial t} \,\text{d}t + \left[ \nabla_x \log \hat{\Psi}^T (f + g^2 \nabla_x \log \Psi) + \frac{1}{2} g^2 \text{Tr} \left[ \nabla_x^2 \log \hat{\Psi} \right] \right] \,\text{d}t + \left[ g \nabla_x \log \hat{\Psi}^T \right] \,\text{d}W_t.
\]
but now noticing that the dynamics of \( \frac{\partial \log \hat{\Psi}}{\partial t} \) become

\[
\frac{\partial \log \hat{\Psi}}{\partial t} = \frac{1}{\hat{\Psi}} \left( -\nabla_x \cdot (\hat{\Psi} f) + \frac{1}{2} \text{Tr}(g^2 \nabla^2 \hat{\Psi}) \right)
= -\nabla_x \log \hat{\Psi}^T f - \nabla_x \cdot f + \frac{1}{2} g^2 \text{Tr}(\hat{\Psi} \nabla_x^2 \hat{\Psi}).
\]

Only the first term in the RHS will be canceled out in (34). Hence, we are left with

\[
d \log \hat{\Psi} = \left[-\nabla_x \cdot f + \frac{1}{2} g^2 \text{Tr}\left(\frac{1}{\hat{\Psi}} \nabla^2 \hat{\Psi}\right)\right] dt
+ \left[g^2 \nabla_x \log \hat{\Psi}^T \nabla_x \log \Psi + \frac{1}{2} g^2 \text{Tr}\left(\nabla^2 \log \hat{\Psi}\right)\right] dt + g \nabla_x \log \hat{\Psi}^T dW_t.
\]

Notice that the trace terms above can be simplified to

\[
\frac{1}{2} \text{Tr}\left(\frac{1}{\hat{\Psi}} \nabla^2 \hat{\Psi} + \nabla_x^2 \log \hat{\Psi}\right) = \text{Tr}\left(\frac{1}{\hat{\Psi}} \nabla^2 \hat{\Psi}\right) - \frac{1}{2} \| \nabla_x \log \hat{\Psi} \|^2
= \frac{1}{2} \| \nabla_x \log \Psi \|^2 + \Delta_x \log \Psi,
\]

where the last equality follows by Lemma 8. Substituting this result back to (35), we get

\[
d \log \hat{\Psi} = \left[-\nabla_x \cdot f + \frac{1}{2} g \nabla_x \log \hat{\Psi} \right] dt + g \nabla_x \log \hat{\Psi}^T dW_t
= \left[\nabla_x \cdot \left(g^2 \nabla_x \log \hat{\Psi} - f\right) + \frac{1}{2} g \nabla_x \log \hat{\Psi} \right] dt + g \nabla_x \log \hat{\Psi}^T dW_t
\]

Finally, by rewriting (33) and (36) with the nonlinear Feynman-Kac in (31) yields

\[
\begin{align*}
dX_t &= (f + gZ_t) dt + g dW_t \\
dY_t &= \frac{1}{2} Z_t^T Z_t dt + Z_t^T dW_t \\
d\hat{Y}_t &= \left(\frac{1}{2} \hat{Z}_t^T \hat{Z}_t + \nabla_x \cdot (g \hat{Z}_t - f) + \hat{Z}_t^T Z_t\right) dt + \hat{Z}_t^T dW_t
\end{align*}
\]

This concludes the proof.

**Remark 9** (Viscosity solutions). These FBSDE results can be extended to viscosity solutions in the case when the classical solution does not exist (Pardoux & Peng, 1992). For the completeness, one shall understand them in the sense of \( \psi(t, x) = \lim_{\epsilon \to 0^+} \psi_\epsilon(t, x) \) uniformly in \((t, x)\) over a compact set. Here \( \psi_\epsilon(t, x) \) is the classical solution to (10) with \((f_\epsilon, G_\epsilon, h_\epsilon, \varphi_\epsilon)\) converge uniformly toward \((f, G, h, \varphi)\) over the compact set. We refer readers of interests to Exarchos & Theodorou (2018); Negyessy et al. (2021), and their references therein.

**Theorem 4** (Log-likelihood for SB models). Given the solution satisfying the FBSDE system in (13), the log-likelihood of the SB model \((Z_t, \hat{Z}_t)\), at a data point \(x_0\), can be expressed as

\[
\mathcal{L}_{SB}(x_0) = \mathbb{E} \left[ \log p_T(X_T) \right] - \int_0^T \mathbb{E} \left[ \frac{1}{2} \| Z_t \|^2 + \frac{1}{2} \| \hat{Z}_t - g \nabla_x \log \hat{P}_t^{SB} + Z_t \|^2 \\
- \frac{1}{2} g \nabla_x \log \hat{P}_t^{SB} - Z_t \|^2 - \nabla_x \cdot f \right] dt
\]

\[
= \mathbb{E} \left[ \log p_T(X_T) \right] - \int_0^T \mathbb{E} \left[ \frac{1}{2} \| Z_t \|^2 + \frac{1}{2} \| \hat{Z}_t \|^2 + \nabla_x \cdot (g \hat{Z}_t - f) + \hat{Z}_t^T Z_t \right] dt,
\]

where the expectation is taken over the forward SDE (13a) with the initial condition \(X_0 = x_0\).
Proof.

\[ \mathcal{L}_{SB}(x_0) = \mathbb{E}[Y_0 + \tilde{Y}_0|X_0 = x_0] \]

\[ = \mathbb{E}[Y_T - \int_0^T \left( \frac{1}{2}\|Z_t\|^2 + \nabla \cdot (g \tilde{Z}_t - f) + \tilde{Z}_t \right) dt|X_0 = x_0] \]

\[ = \mathbb{E}[Y_T + \tilde{Y}_T|X_0 = x_0] - \int_0^T \mathbb{E}\left[ \frac{1}{2}\|Z_t\|^2 + \frac{1}{2}\|\tilde{Z}_t\|^2 + \nabla \cdot (g \tilde{Z}_t - f) + \tilde{Z}_t \right] dt \]

which recovers (39). Finally, notice that with integration by part, we have

\[ \mathbb{E}_X \left[ g \nabla \cdot \tilde{Z}_t \right] = \int (g \nabla \cdot \tilde{Z}_t) \rho_t^{SB} dX_t \]

\[ = - \int g \tilde{Z}_t \nabla_x \log \rho_t^{SB} dX_t \]

(41)

where we adopt common practice and assume the limiting behavior of \( \rho_t^{SB} \); in other words, \( \exists k > 0 : \rho_t^{SB}(x) = O(\exp^{-\|x\|^2}) \) as \( x \to \infty \). With (41), we can rewrite the related parts in (40) as

\[ \mathbb{E}\left[ \frac{1}{2}\|\tilde{Z}_t\|^2 + g \nabla \cdot \tilde{Z}_t + \tilde{Z}_t \right] \]

\[ = \mathbb{E}\left[ \frac{1}{2}\|\tilde{Z}_t\|^2 - \tilde{Z}_t \left( g \nabla \log \rho_t^{SB} \right) + \tilde{Z}_t \right] \]

\[ = \mathbb{E}\left[ \frac{1}{2}\|\tilde{Z}_t - g \nabla \log \rho_t^{SB} + Z_t\|^2 - \frac{1}{2}\|g \nabla \log \rho_t^{SB} - Z_t\|^2 \right] \].

(42)

Hence, we also recover (38).

\[ \square \]

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

\[ dX_t = \left[ f + gZ(t, X_t) - \frac{1}{2} g \left( Z(t, X_t) + \tilde{Z}(t, X_t) \right) \right] dt \]

(43)

**Proof.** The probability ODE flow (Song et al., 2020; Maoutsa et al., 2020) suggests that the equivalent ODE model for the SDE (1) is given by

\[ dX_t = \left[ f - \frac{1}{2} g^2 \nabla_x \log \rho_t^{(1)} \right] dt. \]

We can adopt this result to the SDEs of SB (7a) by considering \( f \leftarrow f + gZ_t \) and \( \rho_t^{(1)} \leftarrow \rho_t^{SB} \). This yields

\[ dX_t = \left[ f + gZ_t - \frac{1}{2} g^2 \nabla_x \log \rho_t^{SB} \right] dt. \]

(44)

Applying the factorization principle (Chen et al., 2021) with \( g \log \rho_t^{SB} = Z_t + \tilde{Z}_t \) concludes the proof.
Remark 10 (Connection between SB-FBSDE and flow-based models). To demonstrate how applying flow-based training techniques to the probability ODE flow of SB (43) recovers the same log-likelihood objective in (39), recall that given an ODE $dX_t = F(t, X_t)dt$ with $X_0 = x_0 \sim p_{data}$, flow-based models compute the change in log-density using the instantaneous change of variables formula (Chen et al., 2018):

$$\frac{\partial \log p(X_t)}{\partial t} = -\nabla_x \cdot F,$$

which implies that the log-likelihood of $x_0$ can be computed as

$$\log p(X_T) = \log p(x_0) - \int_0^T \nabla_x \cdot F \, dt.$$

(45)

Now, consider the probability ODE flow of SB in (44),

$$F_{SB} := f + gZ_t - \frac{1}{2} g(Z_t + \hat{Z}_t) = f + \frac{1}{2} g(Z_t - \hat{Z}_t).$$

Substituting this vector field $F_{SB}$ to (45) yields

$$\log p_{SB}(X_T) = \log p_0(x_0) - \int_0^T \nabla_x \cdot \left( f + \frac{1}{2} g(Z_t - \hat{Z}_t) \right) \, dt$$

$\Rightarrow \mathbb{E} [\log p_0(x_0)] = \mathbb{E} [\log p_{SB}(X_T)] + \int_0^T \mathbb{E} \left[ \nabla_x \cdot \left( f + \frac{1}{2} g(Z_t - \hat{Z}_t) \right) \right] \, dt$

$$= \mathbb{E} [\log p_{SB}(X_T)] - \int_0^T \mathbb{E} \left[ \nabla_x \cdot (g\hat{Z}_t - f) - \frac{1}{2} g\nabla_x \cdot (Z_t + \hat{Z}_t) \right] \, dt$$

$$= \mathbb{E} [\log p_{SB}(X_T)] - \int_0^T \mathbb{E} \left[ \nabla_x \cdot (g\hat{Z}_t - f) + \frac{1}{2} g(Z_t + \hat{Z}_t) \right] \, dt$$

$$= \mathbb{E} [\log p_{SB}(X_T)] - \int_0^T \mathbb{E} \left[ \nabla_x \cdot (g\hat{Z}_t - f) + \frac{1}{2} g(Z_t + \hat{Z}_t)^2 \right] \, dt,$$

(46)

where (*) is due to integration by parts (recall (41)) and (**) again uses the factorization principle $Z_t + \hat{Z}_t = g\nabla_x \log p_{SB}^t$. One can verify that (46) indeed recovers (39).

Theorem 11 (FBSDE computation for $L_{SB}(x_T)$ in SB models). With the same regularity conditions in Theorem 3, the following FBSDEs also satisfy the nonlinear Feynman-Kac relations in (31).

\[
\begin{align*}
\frac{dX_t}{dt} &= (f - g\hat{Z}_t) \, dt + g \, dW_t \tag{47a} \\
\frac{dY_t}{dt} &= -\left(\frac{1}{2} \nabla_x \cdot (g\hat{Z}_t + f) + Z_t^T \hat{Z}_t \right) \, dt + Z_t^T \, dW_t \tag{47b} \\
\frac{d\hat{Y}_t}{dt} &= -\frac{1}{2} \nabla_x \cdot (g\hat{Z}_t + f) + Z_t^T \hat{Z}_t \, dt + \hat{Z}_t^T \, dW_t \tag{47c}
\end{align*}
\]

Given a backward trajectory sampled from (47a), where $X_T = x_T$ and $x_T \sim p_{prior}$, the log-likelihood of $x_T$ is given by $\log p_{prior}(x_T) = \mathbb{E} \left[ Y_T + \hat{Y}_T | X_T = x_T \right] := L_{SB}(x_T)$. In particular,

$$L_{SB}(x_T) = \mathbb{E} [\log p_{SB}(X_0)] - \int_0^T \mathbb{E} \left[ \frac{1}{2} ||\hat{Z}_t||^2 + \frac{1}{2} ||Z_t||^2 + \nabla_x \cdot (g\hat{Z}_t + f) + Z_t^T \hat{Z}_t \right] \, dt,$$

(48)

Proof. Due to the symmetric structure of SB, we can consider a new time coordinate

$$s \triangleq T - t.$$

Under this transformation, the base reference $\mathbb{P}$ appearing in (5) is equivalent to

$$dX_s = -f(s, X_s) \, ds + g \, dW_s.$$

The corresponding PDE optimality becomes

\[
\begin{cases}
\frac{\partial \hat{\Phi}}{\partial s} = \nabla_x \Phi^T f - \frac{1}{2} \text{Tr}(g^2 \nabla_x^2 \Phi) \\
\frac{\partial \Phi}{\partial s} = \nabla_x \cdot (\Phi f) + \frac{1}{2} \text{Tr}(g^2 \nabla_x^2 \Phi) \\
\text{s.t. } \Phi(0, \cdot) = p_{prior}, \quad \Phi(T, \cdot) = p_{data},
\end{cases}
\]

(49)
We conclude the proof. Both concurrent methods rely on solving SB. To see that, recall the forward and backward transition models considered in De Bortoli et al. (2021), utilising a current drift and (estimated) optimal drift. This is in contrast to our SB-FBSDE, which instead relates to the original FBSDE system (30) by

\[
\Phi(s, X_s) = \tilde{\Psi}(T-t, X_{T-t}) \quad \text{and} \quad \hat{\Phi}(s, X_s) = \Psi(T-t, X_{T-t}).
\]  

Equation (51) shall be understood as the forward policy in t-coordinate system corresponds to the backward policy in s-coordinate system, and vice versa. Following similar derivations in the proof of Theorem 3, we can apply Itô lemma to expend the stochastic processes \(d \log \Phi \) and \(d \log \hat{\Phi} \) w.r.t. (50a). This yields the following FBSDE system.

\[
dx_s = (gZ'_s - f) \, ds + g \, dW_s \quad (52a)
\]

\[
dY'_s = \frac{1}{2} \|Z'_s\|^2 \, ds + Z'_s \, dW_s \quad (52b)
\]

\[
d\hat{Y}'_s = \left( \frac{1}{2} \|\hat{Z}'_s\|^2 + \nabla_x \cdot (g\hat{Z}'_s + f) + \hat{Z}'_s \hat{Z}'_s^T \right) \, ds + \hat{Z}'_s \, dW_s \quad (52c)
\]

Similar to (51), \((Y'_s, \hat{Y}'_s, Z'_s, \hat{Z}'_s)\) relate to the original FBSDE system (30) by

\[
(Y'_s, \hat{Y}'_s, Z'_s, \hat{Z}'_s) = (\tilde{\Psi}_{T-t}, Y'_{T-t}, \tilde{Z}_{T-t}, Z_{T-t}).
\]  

Changing the coordinate from \(s\) to \(t\) and applying (53) readily yield (47). Finally, the expression in (48) can be carried out similar to (40):

\[
\mathcal{L}_{SB}(x_T) = E \left[ Y_T + \tilde{\Psi}_T | X_T = x_T \right] = E \left[ Y_0 - \int_0^T \frac{1}{2} \|\bar{Z}_t\|^2 \right] dt + \tilde{\Psi}_0 - \int_0^T \frac{1}{2} \|Z_t\|^2 + \nabla \cdot (gZ_t + f) + Z_t \hat{Z}_t \, dt \bigg| X_T = x_T
\]

\[
= E \left[ Y_0 + \tilde{\Psi}_0 - \int_0^T E \left[ Y_0 + \tilde{\Psi}_0 - \int_0^T \frac{1}{2} \|Z_t\|^2 + \nabla \cdot (gZ_t + f) + Z_t \hat{Z}_t \right] \right] dt
\]

We conclude the proof.

\[\blacksquare\]

C \hspace{1ex} COMPARISON WITH PRIOR SB WORKS

Our method is closely related to two concurrent SB models (De Bortoli et al., 2021; Vargas et al., 2021), yet differs in various aspects. Below we enumerate some of the differences.

**Training loss.** Both concurrent methods rely on solving SB mean-matching regression between the current drift and (estimated) optimal drift. This is in contrast to our SB-FBSDE, which instead utilizes a divergence-based objective (16). However, the regression objectives are in fact captured by Theorem 4. To see that, recall the forward and backward transition models considered in De Bortoli et al. (2021),

\[
X_{k+1} \sim \mathcal{N}(F_k(x_k), 2\gamma_{k+1} I), \quad \text{and} \quad X_k \sim \mathcal{N}(B_{k+1}(X_{k+1}), 2\gamma_{k+1} I),
\]

where \(F_k(x) := x + \gamma_{k+1} f_k(x)\) and \(B_{k+1}(x) := x + \gamma_{k+1} b_{k+1}(x)\) are solved alternately via

\[
B_{k+1} \leftarrow \arg \min_{B_{k+1}} E \left[ \|B_{k+1}(X_{k+1}) - (X_{k+1} + F_k(X_k) - F_k(X_{k+1}))\|^2 \right] \quad (55a)
\]

\[
F_k \leftarrow \arg \min_{F_k} E \left[ \|F_k(X_{k+1}) - (X_k + B_{k+1}(X_k) - B_{k+1}(X_k))\|^2 \right]. \quad (55b)
\]
In what follows, we focus mainly on the connection between (55a) and Theorem 4, yet similar analysis can be applied to (55b). Now, expanding (55a) with the definition of \((B_{k+1}, F_k)\)

\[
\|B_{k+1}(X_{k+1}) - (X_{k+1} + F_k(X_k) - F_k(X_{k+1}))\|^2
\]

\[
= \|\gamma_{k+1} b_{k+1}(X_{k+1}) - (X_{k+1} + \gamma_{k+1} f_k(X_k) - X_{k+1} - \gamma_{k+1} f_k(X_{k+1}))\|^2
\]

\[
= \|\gamma_{k+1} f_k(X_{k+1}) + \gamma_{k+1} b_{k+1}(X_{k+1}) - (X_{k+1} + \gamma_{k+1} f_k(X_k) - X_{k+1})\|^2,
\]

(56)

which resembles the term \(\|Z_t + \tilde{Z}_t - g\nabla_x \log p^{SB}_{t}\|^2\) in (15). While (2) indeed corresponds to our backward policy \(\tilde{Z}(k+1, X_{k+1})\) after time discretization, (1) slightly differs from \(Z(k+1, X_{k+1})\) in how time is integrated, \(\gamma_{k+1} f_k(k, X_{k+1})\) vs. \(Z(k+1, X_{k+1})\). On the other hand, (3) may be seen as an approximation of \(g\nabla_x \log p^{SB}_{t}\), which, crucially, is not utilized in SB-FBSDE training. Since \(\nabla_x \log p^{SB}_{t}\) is often intractable (nor does SB-FBSDE try to approximate it), SB-FBSDE instead uses the divergence-based objective (16), which does not appear in their practical training.

**SDE model class.** It is important to recognize that both concurrent methods are rooted in the classical SB model formulation with the following SDE model,

\[
dX_t = f(t, X_t) dt + \sqrt{2\gamma} dW_t,
\]

which, crucially, differs from the SDE concerned by both our SB-FBSDE and SGM,

\[
dX_t = f(t, X_t) dt + g(t) dW_t,
\]

(57)

in that the diffusion \(g(t)\) is a time-varying function. This implies that the connection between classical SB models and SGM can only be made in discrete-time after choosing proper step sizes. For instance, De Bortoli et al. (2021) considers the Euler-Maruyama discretization (see their §C.3),

\[
X_{k+1} = X_k + \gamma_{k+1} f(k, X_k) + \sqrt{2\gamma_{k+1}} \epsilon,
\]

(58)

where \(\epsilon \sim N(0, I)\). In order for (58) to match the discretization of (57), where \(g(t)\) is often a monotonically increasing function, the step sizes \(\{\gamma_k\}_{k=1}^N\) must also increase monotonically. However, since \(\nabla_x \log p^{SB}_{t}\) is approximated in De Bortoli et al. (2021) using the states from two consecutive steps (see (55)), this may also affect the accuracy of the regression targets.

In contrast, our SB-FBSDE is grounded on the recent SB theory (Caluya & Halder, 2021), which considers the same SDE model as in (57). As such, connection between SB-FBSDE and SGM is made directly in continuous-time (and can be extended to discrete-time flawlessly); hence unaffected by the choice of numerical discretization or step sizes.

**Model parametrization.** While Vargas et al. (2021) utilizes non-parametric models, e.g. Gaussian processes (hence are not directly comparable), both De Bortoli et al. (2021) and SB-FBSDE use DNNs to approximate the SB policies.

**Training algorithm and convergence.** Both concurrent methods rely on solving SB with IPF, which performs alternate training between the forward/backward policies. While SB-FBSDE can also be trained with IPF (see Alg. 3), we stress that it is also possible to train both policies jointly whenever the computational budget permits. Interestingly, this joint optimization – which is not presented in concurrent methods – resembles the training scheme of the recently-proposed diffusion flow-based model (Zhang & Chen, 2021). We highlight these flexible training procedures arising from the unified viewpoint provided in Theorem 4. Finally, with the close relation between Alg. 3 and IPF (despite different objectives and SDE model classes), convergence analysis from classical IPF can be applied with few efforts. We leave it as a promising future work.

**Corrector Sampling.** While both De Bortoli et al. (2021) and SB-FBSDE implement corrector sampling, they corresponds to different quantities. Specifically, our SB-FBSDE relies on the same predictor-corrector scheme proposed in SGM (see Sec 4.2 in Song et al. (2020)), where the “corrector” part is made with a Langevin sampling using the desired optimal density \(\nabla \log p^u\). In SB, this term corresponds exactly to adding the outputs of our networks \(Z + \tilde{Z}\). This computation differs from the corrector sampling appearing in De Bortoli et al. (2021), which relies on single network (i.e. either \(Z\) or \(\tilde{Z}\). Crucially, this implies that the two methods is approaching different target distributions; hence leading to different training results. Notably, it has been reported in De Bortoli et al. (2021) that corrector sampling only gives negligible improvement (see §J.2 in De Bortoli et al. (2021)), yet in our case we observe major quantitative improvement (up to 4 FID).
### D Experiment Details

**Table 1 with other models.**

<table>
<thead>
<tr>
<th>Model Class</th>
<th>Method</th>
<th>NLL ↓</th>
<th>FID ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Optimal Transport</strong></td>
<td>SB-FBSDE (ours)</td>
<td>2.96</td>
<td>3.01</td>
</tr>
<tr>
<td></td>
<td>DOT (Tanaka, 2019)</td>
<td>-</td>
<td>15.78</td>
</tr>
<tr>
<td></td>
<td>Multi-stage SB (Wang et al., 2021)</td>
<td>-</td>
<td>12.32</td>
</tr>
<tr>
<td></td>
<td>DGflow (Ansari et al., 2020)</td>
<td>-</td>
<td>9.63</td>
</tr>
<tr>
<td><strong>SGMs</strong></td>
<td>SDE (deep, sub-VP; Song et al. (2020))</td>
<td>2.99</td>
<td>2.92</td>
</tr>
<tr>
<td></td>
<td>ScoreFlow (Song et al., 2021)</td>
<td>2.74</td>
<td>5.7</td>
</tr>
<tr>
<td></td>
<td>VDM (Kingma et al., 2021)</td>
<td><strong>2.49</strong></td>
<td>4.00</td>
</tr>
<tr>
<td></td>
<td>LSGM(Vahdat et al., 2021)</td>
<td>3.43</td>
<td><strong>2.10</strong></td>
</tr>
<tr>
<td><strong>VAEs</strong></td>
<td>VDVAE (Child, 2020)</td>
<td>2.87</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>NVAE (Vahdat &amp; Kautz, 2020)</td>
<td>2.91</td>
<td>23.49</td>
</tr>
<tr>
<td></td>
<td>BIVA (Maaløe et al., 2019)</td>
<td>3.08</td>
<td>-</td>
</tr>
<tr>
<td><strong>Flows</strong></td>
<td>FFJORD (Grathwohl et al., 2018)</td>
<td>3.40</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>VFlow (Chen et al., 2020)</td>
<td>2.98</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>ANF (Huang et al., 2020)</td>
<td>3.05</td>
<td>-</td>
</tr>
<tr>
<td><strong>GANs</strong></td>
<td>AutoGAN (Gong et al., 2019)</td>
<td>-</td>
<td>12.42</td>
</tr>
<tr>
<td></td>
<td>StyleGAN2-ADA (Karras et al., 2020a)</td>
<td>-</td>
<td>2.92</td>
</tr>
<tr>
<td></td>
<td>LeCAM (Park &amp; Kim, 2021)</td>
<td>-</td>
<td>2.47</td>
</tr>
</tbody>
</table>

**Figure 7: Training Hyper-parameters**

<table>
<thead>
<tr>
<th>Dataset</th>
<th>learning rate</th>
<th>time steps</th>
<th>batch size</th>
<th>variance of $p_{\text{prior}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Toy</td>
<td>2e-4</td>
<td>100</td>
<td>400</td>
<td>1.0</td>
</tr>
<tr>
<td>Mnist</td>
<td>2e-4</td>
<td>100</td>
<td>200</td>
<td>1.0</td>
</tr>
<tr>
<td>CelebA</td>
<td>2e-4</td>
<td>100</td>
<td>200</td>
<td>900.0</td>
</tr>
<tr>
<td>CIFAR10</td>
<td>1e-5</td>
<td>200</td>
<td>64</td>
<td>2500.0</td>
</tr>
</tbody>
</table>

**Figure 8: Network Architectures**

<table>
<thead>
<tr>
<th>Dataset</th>
<th>$Z_t(\cdot, \cdot; \theta)$ and # of parameters</th>
<th>$Z_t(\cdot, \cdot; \phi)$ and # of parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Toy</td>
<td>FC-ResNet (0.76M)</td>
<td>FC-ResNet (0.76M)</td>
</tr>
<tr>
<td>Mnist</td>
<td>reduced Unet (1.95M)</td>
<td>reduced Unet (1.95M)</td>
</tr>
<tr>
<td>CelebA</td>
<td>Unet (39.63M)</td>
<td>Unet (39.63M)</td>
</tr>
<tr>
<td>CIFAR10</td>
<td>NCSN++ (62.69M)</td>
<td>Unet (39.63M)</td>
</tr>
</tbody>
</table>

**Training.** We use Exponential Moving Average (EMA) with the decay rate of 0.99. Table 7 details the hyper-parameters used for each dataset. As mentioned in De Bortoli et al. (2021), the alternate training scheme may substantially accelerate the convergence under proper initialization. Specifically, when $Z_t$ is initialized with degenerate outputs (e.g. by zeroing out its last layer), training $\hat{Z}_t$ at the first $K$ steps can be made in a similar SGM fashion since $p_{SB}^t$ now admits analytical expression. As for the proceeding stages, we resume to use (18, 19) since $(Z_t, \hat{Z}_t)$ no longer have trivial outputs.

**Data pre-processing.** MNIST is padded from $28 \times 28$ to $32 \times 32$ to prevent degenerate feature maps through Unet. CelebA is resized to $3 \times 32 \times 32$ to accelerate training. Both CelebA and CIFAR10 are augmented with random horizontal flips to enhance the diversity.
Sampling. The sampling procedure is summarized in Alg. 4. Given some pre-defined signal-to-noise ratio \( r \) (we set \( r = 0.05 \) for all experiments), the Langevin noise scale \( \sigma_{t,i} \) at each time step \( t \) and each corrector step \( i \) is computed by

\[
\sigma_{t,i} = \frac{2r^2g^2\|\epsilon_i\|^2}{\|\begin{bmatrix} Z(t, X_{t,i}) + Z(t, X_{t,i}) \end{bmatrix}\|^2},
\]

(59)

Network architectures. Table 8 summarizes the network architecture used for each dataset. For toy datasets, we parameterize \( Z(\cdot, \cdot; \theta) \) and \( Z(\cdot, \cdot; \phi) \) with the architectures shown in Fig. 9. Specifically, FCBlock represents a fully connected layer followed by a swish nonlinear activation (Ramachandran et al., 2017). 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. (2020).

Remarks on Table 1. We note that the values of our SB-FBSDE reported in Table 1 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 \( KL(p_T, p_{\text{prior}}) \), we first compute the pixel-wise first and second moments given the generated samples \( X_T \) at the end of the forward diffusion. After fitting a diagonal Gaussian to \( \{X_T\} \), 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.

E Additional Experiments

Comparison to De Bortoli et al. (2021) under same setup. To demonstrate the superior performance of our model, we conduct experiments with the exact same setup implemented in De Bortoli et al. (2021). Specifically, we adopt the same network architecture (reduced U-net), image pre-processing (center-cropping 140 pixel and resizing to 32 \( \times \) 32), step sizes (\( N=50 \)), and horizon (0.5 second) for fair comparison. Comparing our Fig. 10b to De Bortoli et al. (2021) (see their Fig. 6), it is clear that images generated by our model have higher diversity (e.g., color skin, facing angle, background, etc) and better visual quality. We conjecture that our performance difference may come from (i) the (in)sensitivity to numerical discretization between our divergence objectives and their mean-matching regression, and (ii) the foundational differences in how diffusion coefficients are designed.
Figure 10: Comparison between images generated by ground truth and SB-FBSDE on reduced CelebA. Our SB-FBSDE is trained under the same data pre-processing, network architecture and stepsizes implemented in De Bortoli et al. (2021).

Figure 11: Qualitative results at the different stages of training. (a) Results after 50k training iterations using SGM’s regression loss. (b) Refine the results of Fig. 11a by training the backward policy using (18) with 5k iterations. (c) Refine the results of Fig. 11a with a full SB-FBSDE stage using (18,19).

SGM regression training + SB divergence-based training. Table 3 reports the FID (using 10k samples, without corrector steps) at different stages of CIFAR10 training. We first train the backward policy with SGM’s regression loss for a sufficient long iterations (50k) until the FID roughly converges. Then, we switch to our alternate training (Alg. 3) using the divergence-based objectives. Crucially,
Table 3: SGM regression training + SB divergence-based training. We denote “SGM/50k” as “training 50k steps using SGM loss”, and “SB/(f,b)/5k” as “training forward/backward policy with 5k steps using our divergence loss”, and etc.

<table>
<thead>
<tr>
<th>Initialization</th>
<th>SGM/10k</th>
<th>SGM/20k</th>
<th>SGM/50k</th>
<th>SGM/50k + SB/b/5k</th>
<th>SGM/50k + SB/f/5k + SB/b/5k</th>
</tr>
</thead>
<tbody>
<tr>
<td>FID</td>
<td>448</td>
<td>41.37</td>
<td>35.47</td>
<td>33.68</td>
<td>13.35</td>
</tr>
</tbody>
</table>

With only 5k iterations of our divergence-based training, we drop the FID dramatically down to 13.35 from 33.68. With a full stage of training (last column), the FID decreases even lower to 11.85. The qualitative results are provided in Fig. 11. Comparing Fig. 11a (corresponds to “SGM/50k” in Table 3) and Fig. 11b (corresponds to “SGM/50k + SB/b/5k” in Table 3), it can be seen that the visible flaw and noise have been substantially improved.

Additional Figures

![Figure 12: Uncurated samples generated by our SB-FBSDE on MNIST.](image)
Figure 13: Uncurated samples generated by our SB-FBSDE on resized CelebA.
Figure 14: Uncurated samples generated by our SB-FBSDE on CIFAR10.