A Flexible Diffusion Model

Anonymous Authors[1]

Abstract

Denoising Diffusion (score-based) generative models have been widely used for modeling various types of complex data, including images, audio, point clouds, and biomolecules. Recently, the deep connection between forward-backward stochastic differential equations (SDEs) and diffusion-based models has been revealed, and several new variants of SDEs are proposed (e.g., sub-VP, critically-damped Langevin) along this line. Despite the empirical success of several hand-crafted forward SDEs, a great quantity of potentially promising forward SDEs remains unexplored. In this work, we propose a general framework for parameterizing the diffusion models, especially the spatial part of the forward SDEs. A systematic formalism is introduced with theoretical guarantees, and its connection with previous diffusion models is leveraged. Finally, we demonstrate the theoretical advantage of our method from the variational optimization perspective. Numerical experiments on synthetic datasets, MNIST and CIFAR10 are presented to validate the effectiveness of our framework.

1. Introduction

Denoising Diffusion (score-based) models, which originated from non-equilibrium statistical physics, have recently shown impressive success on sample generations of a wide range of types, including images (Ho et al., 2020; Nichol and Dhariwal, 2021; Song et al., 2020a; Dhariwal and Nichol, 2021; Rombach et al., 2022), 3D point clouds (Luo and Hu, 2021; Du et al., 2021), audio (Kong et al., 2020; Liu et al., 2021), and biomolecules generation (Xu et al., 2022; Hoogeboom et al., 2022; Schneuing et al., 2022). In addition to practical applications of various diffusion generative models, it is also desirable to analyze them in an appropriate and flexible framework, by which novel improvements can be further developed.

Currently, one of the promising formal frameworks for unifying different types of diffusion models is to utilize the stochastic differential equations (SDEs), as proposed in (Song et al., 2020a). Under this formalism, a diffusion model consists of a forward (noising) process and a backward (denoising) process. The forward process keeps adding noise to the real data, and the backward (generative) process can be viewed as reversing the forward process in terms of probability. Furthermore, with the help of the Feynman-Kac formula and Girsanov transform (Da Prato, 2014), the score-matching training scheme has been proved to be equivalent to certain log-likelihood (ELBO) training in the infinite-dimensional path space (Huang et al., 2021).

From the variational optimization point of view, although the ELBO optimization function of diffusion models explicitly contains both the forward and backward ingredients, the forward (noising) process is usually hand-crafted and set to be fixed throughout the training process (Huang et al., 2021). If we treat the forward-backward processes as an encoder-decoder pair, then there exists an obvious mismatch between the current training framework of diffusion models and other log-likelihood based models (e.g., Hierarchical VAE (Vahdat and Kautz, 2020)) which also optimize the encoder. Moreover, since the reverse (generative) process is uniquely determined by the forward process, the total flexibility of the model actually lies in parameterizing the forward process. Given the fact that different noising schedules have proven to affect the empirical performances (e.g., the different forward processes including VE, VP, sub-VP (Song et al., 2020a) and damped Langevin diffusion (Dockhorn et al., 2022) displayed distinct generation performances), freezing the forward process is both theoretical and practical incomplete. Therefore, the main research question of this paper is: *Can we introduce a theoretically grounded parameterization for the forward process, so that the diffusion model can automatically optimize it from data?*

To address this problem, it is crucial to incorporate flexible parameterized forward processes into the general SDE framework in (Song et al., 2020a). Though the idea of training the forward process is intuitively reasonable, the implementation is far from straightforward. The first challenge is to find the appropriate sub-class within the grand function

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

space consisting of the whole stochastic processes. A hard constraint is that the stationary distribution of the candidate stochastic processes must be simple (usually the centered Gaussian), which will be set as the generative SDE's prior distribution. The second challenge is how to make sure that our parameterization is flexible enough to include all proper SDEs. In fact, even parameterizing the noise schedule of the forward process (the one-dimensional time component) would improve the diffusion model's performance, as it was shown in (Kingma et al., 2021). However, how to efficiently parameterize the space components of the forward process remains to be explored, especially taking into account the complex structure of the data distribution (Narayanan and Mitter, 2010).

This paper concentrates on both theoretical and practical aspects of solving the flexibility challenge of the diffusion model in a unified way, emphasizing the spatial components of the forward process. First of all, inspired by concepts from Riemannian geometry and Hamilton Monte-Carlo methods, we define a flexible class of diffusion processes (FP-Diffusion) that rigorously satisfies the fixed Gaussian stationary distribution condition with theoretical guarantees. To highlight the advantages of flexible diffusion models, we also discuss the theoretical motivations and properties of parameterized forward processes from the variational optimization perspective. Furthermore, by introducing the flexible diffusion model, all sorts of regularizers for smoothing the diffusion paths (e.g., methods from continuous normalizing flows: (Finlay et al., 2020; Onken et al., 2021)) can be implemented for designing better diffusion models. We empirically test some of them in the experiment section.

Our major contributions are as follows:

- We introduce a theoretically complete framework for parameterizing the forward process with the help of symplectic structures and the anisotropic Riemannian structure. Convergence properties as $t \to \infty$ are proved along the same route.
- To motivate the parameterization of the forward process, we analyze the implications of parameterizing the forward (noising) process from the variational optimization point of view and demonstrate how our method unifies previous diffusion models. Since this extension allows merging regularization terms into the training loss, we also provide experimental results in simulated scenarios and demonstrate how the diffusion path behaves under regularization.
- Except considering the general diffusion parameterization framework, we also develop a corresponding simplified version of our method with explicit formulas for efficient Monte-Carlo training. It enables us to perform comparative studies on relatively large-scale datasets, e.g., CIFAR10.

2. Preliminaries and Related Works

Given a data distribution $p(x)$, we associate it with a Gaussian diffusion process (forward) that increasingly adds noise to the data, then the high-level idea of diffusion generative models is to approximate the real data distribution by fitting a multi-step denoising (backward) process. In a discrete setting, the forward process is formulated as an N-steps Markov chain from real data x to each noised x_t:

$$p(x_t|x_{t-1}) = \mathcal{N}(\alpha_t x, \beta_t I), \;\; t \in \{1, \dots, N\}.$$

For DDPM model (Ho et al., 2020), α_t is set to be $\alpha_t := \sqrt{1-\beta_t}$. Taking the continuous limit of β_t (when $\sqrt{1-\beta_t} \approx 1 - \frac{1}{2}\beta_t$), we find that X_t satisfies the time-changed Ornstein–Uhlenbeck stochastic differential equation (SDE):

$$dX_t = -\frac{1}{2}\beta(t)X_t dt + \sqrt{\beta(t)}dW_t, \tag{1}$$

which is exactly the so-called *variance-preserving* diffusion process (VP) in (Song et al., 2020a). Therefore, DDPM can be treated as a discretization of the Ornstein–Uhlenbeck process. Following this line, (Song et al., 2020a) proposed to characterize different types of diffusion models by formulating the underlying SDE of each model:

$$dX_t = f(X_t, t)dt + g(t)dW_t, \;\; 0 \le t \le T \tag{2}$$

where $\{W_t\}_{t=0}^{\infty}$ denotes the standard Brownian motion, and the dimension is set to be the same as the data. Usually we choose a different time parameterization (time-change) for t. Let $\beta(t)$ be a continuous function of time t such that $\beta(t) > \beta(s) > 0$ for $0 < s < t$, then $\beta(t)$ is called a specific time schedule (time-change) of t. It can be further shown that when $t \to \infty$, the stationary distribution of Eq. 1 is the standard multivariate Gaussian: $\mathcal{N}(0, I)$ (Hsu, 2002). On the other hand, SMLD diffusion models (Song and Ermon, 2019) can be seen as a discretization of the *variance-exploring* (VE) process ((9) of (Song et al., 2020a)) $\{X_t\}_{t=0}^{t=T}$, which satisfies a different SDE:

$$dX_t = \sqrt{2\sigma(t)\sigma'(t)}dW_t. \tag{3}$$

A remarkable property of all the above SDE solution classes is the existence of a reverse process Y_t with respect to each forward SDE X_t, in the sense that the marginal distributions at each time and its corresponding 'reverse' time match:

$$p_t(X_t) \equiv q_{T-t}(Y_{T-t}), \;\; 0 \le t \le T.$$

We name Y_t as the backward (denoising) stochastic process of the diffusion model. In other words, real data is generated

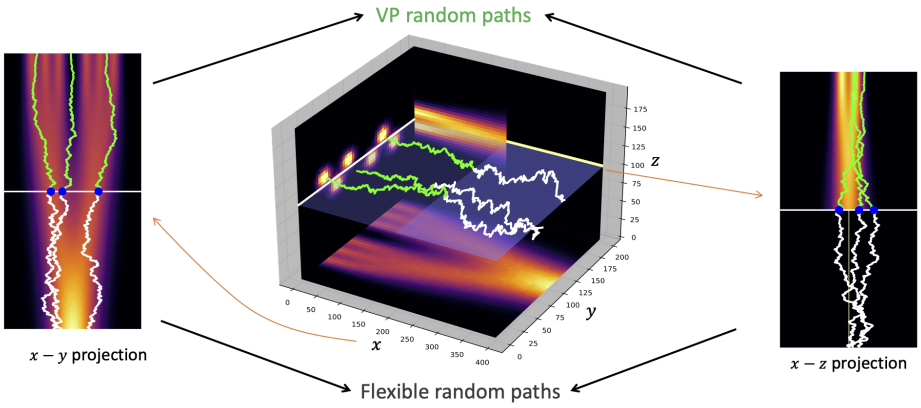


Figure 1. Evolution trajectories of fixed and flexible forward SDEs

by sampling from the Gaussian distribution and tracking the denoising process from time T to 0. Surprisingly, the underlying equation of the reverse-time process Y_t is derived analytically in (Anderson, 1982; Song et al., 2020a):

$$dY_t = [f(Y_t, t) - g^2(Y_t, t)\nabla \log p_t(Y_t)]dt + g(t)dW_t, \quad (4)$$

where W_t is a Brownian motion running backwards in time from T to 0. Then, it's obvious that the unknown score function $s_t(x) := \nabla \log p_t(x)$ depends on both the data distribution p_0 and the forward process X_t. To estimate the score function and Y_t, continuous diffusion models utilize various types of (weighted) score-matching procedures, we will briefly review some typical examples in section 3.2.

Now we summarize more related works:

Diffusion Probabilistic Models (DPM) as a generative model (Kingma and Welling, 2013; Goodfellow et al., 2014; Yang et al., 2021; Ren et al., 2021) was first introduced in (Sohl-Dickstein et al., 2015), as a probabilistic model inspired by non-equilibrium thermodynamics. The high-level idea is to treat the data distribution as the Gibbs (Boltzmann) equilibrium distribution (Friedli and Velenik, 2017), then the generating process corresponds to transitioning from non-equilibrium to equilibrium states (De Groot and Mazur, 2013). DDPM (Ho et al., 2020) and (Nichol and Dhariwal, 2021; Song et al., 2020b; Watson et al., 2022; Jolicoeur-Martineau et al., 2021; Bao et al., 2022) further improve DPMs by introducing Gaussian Markov chains and various inference and sampling methods, through which the generative model is equivalent to a denoising diffusion model. (Vahdat et al., 2021) then introduces a latent space diffusion and the number of denoising steps are also increased to improve empirical performances. On the other hand, as we will show in this article, there are infinite processes (thermodynamical systems) that can connect non-equilibrium states to a equilibrium.

Score Matching: Score-based energy models (Hyvärinen, 2005; Vincent, 2011) are based on minimizing the difference between the derivatives of the data and the model's log-density functions, which avoids calculating the normalization constant of an intractable distribution. (Song and Ermon, 2019; Song et al., 2020c) then introduced sliced score matching that enabled scalable generative training by leveraging different levels of Gaussian noise and several empirical tricks. (Song et al., 2020a; 2021) further studied how to perturb the data by a continuous stochastic process.Under this framework, (Kingma et al., 2021) proposed to reparameterize and optimize the time variable of the forward process (the spatial components remain fixed) by the signal-to-noise ratio (SNR). From this point of view, our model can be seen as a novel spatial parameterization of the forward process, which takes into account the spatial inhomogeneity of the data distribution.

3. Methods

3.1. A general framework for parameterizing diffusion models

From the preliminary section, we realize that the stationary distribution of the forward process will also be the initial distribution of the denoising (generative) process. Therefore, it must be a simple distribution we know how to sample from, mainly set to be standard Gaussian. In this article, we parameterize the spatial components of the forward process by considering the following SDE:

$$dX_t = f(X_t)dt + \sqrt{2R(X_t)}dW_t, \quad (5)$$

under the **hard constraint** that the stationary distribution of X_t is standard Gaussian (the scaled Gaussian case is included in Appendix). Introducing the time change $\beta(t)$, then by Ito's formula, $X_{\beta(t)}$ satisfies a variant of Eq. 5:

$$dX_{\beta(t)} = f(X_t)\beta'(t)dt + \sqrt{2\beta'(t)R(X_t)}dW_t. \quad (6)$$

Comparing with black-box parameterizations (e.g. (Zhang and Chen, 2021)), it's obvious that the function class of $f(x)$ and $R(x)$ should be properly restricted to satisfy the diffusion model's theoretical assumptions. To solve this issue, we propose a flexible framework for parameterizing the forward processes, and the **completeness** of our parameterization will be proved in Appendix A.1. It turns out that the whole construction can be decomposed into two parts: the Riemannian metric and the symplectic form in $\mathbf{R}^n$, inspired by ideas from the Riemannian Manifold Hamiltonian Monte-Carlo algorithm (Girolami and Calderhead, 2011; Betancourt, 2013; 2017; Seiler et al., 2014) and anisotropic diffusion technique of image processing, graph deep learning (Weickert, 1998; Perona and Malik, 1990; Alvarez et al., 1992).

Intuitively, an anisotropic Riemannian metric implies that the space was curved, and the corresponding 'inhomogeneous' Brownian motion will inject non-uniform noise along different directions. On the other hand, the symplectic form is crucial for defining the dynamics of a given Hamiltonian. Both of them set the stage for performing diffusion on the data manifold, from real data distribution to the standard multivariate normal distribution, whose density under the canonical volume form $dx_1 \dots dx_n$ is

$$\frac{1}{\sqrt{(2\pi)^n}} \exp(-\frac{1}{2} \|x\|^2) dx_1 \dots dx_n. \tag{7}$$

Now we introduce these two geometric concepts in detail. In a coordinate system, a Riemannian metric can be identified as a symmetric positive-definite matrix : $R(x) := \{R_{ij}(x)\}_{1\leq i,j\leq n}$ (the Euclidean metric corresponds to the identity matrix). Given a smooth function $H(x)$, recall that the Riemannian Langevin process satisfies the following SDE:

$$dX_t = -\tilde{\nabla} H(X_t)dt + \sqrt{2}dB_t, \tag{8}$$

where $\tilde{\nabla} H(x) := R^{-1}(x)\nabla H(x)$ is the gradient vector field of H, and B_t denotes the **Riemannian** Brownian motion (Hsu, 2002). In local coordinates, B_t equals (see (13) of (Girolami and Calderhead, 2011)):

$$dB_t^i = |R(X_t)|^{-1/2} \sum_{j=1}^{n} \frac{\partial}{\partial x_j}(R_{ij}^{-1}(X_t)|R(X_t)|^{1/2})dt + \sqrt{R^{-1}(X_t)}dW_t^i, \tag{9}$$

for $i \in \{1, 2, \dots, n\}$. One crucial property of the Riemannian Langevin process (Wang, 2014) is that its stationary distribution $p(x)$ has the following form:

$$p(x) \propto e^{-H(x)} dV(x),$$

where $dV(x) := \sqrt{|R(x)|}dx_1 \dots dx_n$ is the Riemannian volume form. Transforming back to the canonical volume form and take $H(x) = \frac{1}{4}\|x\|^2 \cdot \log(|R(x)|)$, we have proved the following lemma:

Lemma 3.1. *The stationary distribution of the SDE (Eq. 10) below is the standard Gaussian of* $\mathbf{R}^n$*:*

$$dX_t = \frac{1}{2}[-\sum_j R_{ij}^{-1}(X_t) \cdot (X_t)_j + \sum_j \frac{\partial}{\partial x_j} R_{ij}^{-1}(X_t)]dt + \sqrt{R^{-1}(X_t)}dW_t, \tag{10}$$

Remark 3.2. It's worth mentioning that the infinitesimal generator of (10) is the Riemannian Laplacian: Δ_R. When acting on a smooth function f,

$$\Delta_R f := \frac{1}{\sqrt{|R(x)|}} \partial_i (\sqrt{|R(x)|}(R^{-1})_{ij}\partial_j f).$$

Indeed, it has the same form as the anisotropic diffusion defined by (1.27) of (Weickert, 1998). The effectiveness of anisotropic noise is explored in section 4.1.

On the other hand, introducing a symplectic form ω allows us to do Hamiltonian dynamics in an even-dimensional space $\mathbf{R}^{2d}$. Since a symplectic form is a non-degenerate closed 2-form, it automatically becomes zero in odd-dimensional spaces. In this article, we will restrict ourselves to a special type of symplectic form, which consists of constant anti-symmetric matrices $\{\omega_{ij}\}_{1\leq i,j\leq 2d}$. Then the corresponding Hamiltonian dynamics of $H(x)$ is:

$$dX_t = \omega \nabla H(X_t)dt. \tag{11}$$

We mainly focus on two remarkable properties of Hamiltonian dynamics: 1. It preserves the canonical volume form (the determinant of the corresponding Jacobi matrix always equals one); 2. The Hamiltonian function $H(x)$ takes a constant value along the integral curves (see the remark in Appendix A). Using the change of variables formula, we conclude that the probabilistic density of X_t preserves the equilibrium Gibbs distribution:

$$p(x) \propto e^{-H(x)} dx_1 \dots dx_n,$$

where X_0 is sampled from the Gibbs distribution.

Let $H(x) = \frac{1}{2}x^2$, the potential energy of the Harmonic oscillator. Then by merging the Riemannian part (Eq. 10) and the symplectic part (Eq. 11) we obtain the following theorem:

Theorem 3.3. *Suppose* ω *is an anti-symmetric matrix, and* $R^{-1}(x)$ *is a symmetric positive-definite matrix-valued function of* $x \in \mathbf{R}^n$*. Then the (unique) stationary distribution of (Eq. 12) below is the standard Gaussian (Eq. 7) of* $\mathbf{R}^n$*:*

$$dX_t = \frac{1}{2}[-\sum_j R_{ij}^{-1}(X_t) \cdot (X_t)_j - 2\sum_j \omega_{ij} \cdot (X_t)_j + \sum_j \frac{\partial}{\partial x_j} R_{ij}^{-1}(X_t)]dt + \sqrt{R^{-1}(X_t)}dW_t, \tag{12}$$

We name Eq. 12 as our **FP-Diffusion** model, and previous diffusion models (e.g., Eq. 1) are included by setting $\omega \equiv 0$, $R^{-1}(x) \equiv I$. In Appendix A.1, theorem 3.3 is extended to scaled Gaussian distributions by direct computation. For a graphical presentation, Figure 1 plots the VP stochastic trajectories (the green curves) connected with our FP-Diffusion forward trajectories (the white curves) under random initialization. We also provide an informal argument on how the anisotropic FP-Diffusions mix with the low-dimensional data distribution in Appendix A.1.

Furthermore, to **unify** the critical damped Langevin diffusion model (Dockhorn et al., 2022), our FP-Diffusion is straightforward to generalize to the case when the inverse Riemannian matrix $R^{-1}(x)$ degenerates (contains zero eigenvalues). Intuitively, the diffusion part $\sqrt{R^{-1}(x)}dW_t$ is the source of randomness (noise). Suppose $R^{-1}(x)$ degenerates along the i-th direction (i.e., corresponding to zero eigenvalue), then no randomness is imposed on this direction, and the i-th component X_t^i will be frozen at X_0^i. In this case, X_t may not converge to this Gaussian stationary distribution from a deterministic starting point. To remedy this issue, we impose additional restrictions, which lead us to the following corollary:

Corollary 3.4. *Under two additional conditions: (1) the symplectic form $\omega \in \mathbb{R}^{2d\times 2d}$ has the block form: $\omega = \left(\begin{array}{c|c} 0 & A \\ \hline -A & 0 \end{array}\right)$ with a positive-definite matrix $A \in \mathbb{R}^{d\times d}$; (2) the inverse (semi-) Riemannian matrix $R^{-1}(x)$ has the block form: $R^{-1}(x) = \left(\begin{array}{c|c} 0 & 0 \\ \hline 0 & B \end{array}\right)$ with a constant positive-definite symmetric matrix $B \in \mathbb{R}^{d\times d}$, we induce that the forward diffusion X_s converges to the standard Gaussian distribution:*

$$p_s(X_s) \xrightarrow{s\to\infty} \mathcal{N}(0, I).$$

We will demonstrate how the corollary derives the damped diffusion model in Appendix A.

3.2. Parameterizing diffusion models from the optimization perspective

In this section, we illustrate the benefits of parameterized diffusion models from the variational optimization perspective. Recall that the ground-truth reverse-time SDE of the forward process X_t is denoted by Y_t, and we parameterize Y_t by Y_t^θ :

$$dY_t^\theta = [f(Y_t,t) - g^2(Y_t,t)\nabla \mathbf{s}_\theta(Y_t,t)]dt + g(t)dW_t, \quad (13)$$

where $\mathbf{s}_\theta$ is the score neural network parameterized by θ. Then the (explicit) score-matching loss function for optimization is

$$L_{\text{ESM}} := \int_0^T \mathbb{E}_{X_s}[\frac{1}{2}\left\|\mathbf{s}_\theta(X_s,s) - \nabla \log p_s(X_s)\right\|^2_{\Lambda(s)}]ds, \quad (14)$$

where $\Lambda(s)$ is a weighting positive definite matrix for the loss. Since Y_t and X_t shares the same marginal distributions, then under the condition that the parameterized generative process Y_t^θ matches Y_t perfectly:

$$\mathbf{s}_\theta(x,t) \equiv \nabla \log p_t(x) \quad (15)$$

for all $t \in [0,T]$, we know the marginal distribution of Y_t^θ at $t = 0$ is exactly the data distribution.

The major obstacle of optimizing Eq. 14 directly is that we don't have access to the ground truth score function $\nabla \log p_s(x,s)$. Fortunately, L_{ESM} can be transformed to a loss based on the accessible conditional score function $\nabla \log p_{X_s|X_0}(X_s)$ plus a constant (Song et al., 2020c;a) (**for a fixed forward process** X_s). More precisely, given two time slices $0 < s < t < T$,

$$\begin{aligned} &\mathbb{E}_{X_t}\left\|\mathbf{s}_\theta(X_t,t) - \nabla \log p_t(X_t)\right\|^2 \\ &\equiv \mathbb{E}_{X_s,X_t}\left\|\mathbf{s}_\theta(X_t,t) - \nabla \log p_t(X_t|X_s)\right\|^2 \\ &+ \underbrace{\mathbb{E}_{X_t}\left\|\nabla \log p_t(X_t)\right\|^2 - \mathbb{E}_{X_s,X_t}\left\|\nabla \log p_t(X_t|X_s)\right\|^2}_{\text{gap terms}}. \end{aligned} \quad (16)$$

Since the gap terms between the original and conditional score function loss only depend on the forward noising process, one **theoretical advantage** of FP-Diffusion is that the gap terms are also parameterized. This formula is adapted from (Song et al., 2020c; Huang et al., 2021) by modifying the initial time, and full derivations are given in Appendix A for completeness.

On the other hand, compared with log-likelihood generative models like normalizing flows and VAE, the connection between score matching and the log-likelihood of data distribution $\log p_0(x)$ (also the initial distribution of (12)) is also not straightforward due to the additional forward process X_t. Hence, we turn to the variational view established in (Huang et al., 2021), where the ELBO (evidence lower bound) of data's log-likelihood $\log p_0(x)$ is directly related with the score matching scheme. More precisely, we have

$$\log p_0(x) \geq \mathcal{E}^\infty(x),$$

and the ELBO $\mathcal{E}^\infty(x)$ of the infinite-dimensional path space is defined by

$$\begin{aligned} \mathcal{E}^\infty(x) := &\mathbb{E}_{X_T}[\log p_T(X_T)|X_0 = x] \\ &- \int_0^T \mathbb{E}_{X_s}[\frac{1}{2}\left\|\mathbf{s}_\theta\right\|^2_{g^2} + \nabla\cdot(g^2\mathbf{s}_\theta - f)|X_0 = x]ds. \end{aligned} \quad (17)$$

The above implies that learning a diffusion (score) model is equivalent to maximizing the ELBO in the variational path space defined by the generative process Y_t^θ. Thus, treating $f(x,t)$ and $g(x,t)$ as learnable functions results in enlarging the variational path space from pre-fixed f and g to flexible variational function classes, such that a lower value of ELBO is achieved in the extended space.

By Eq. 12, in FP-Diffusion model, we set

$$f(x,t) := \frac{\beta'(t)}{2}[-\sum_j R_{ij}^{-1}(x)x_j - 2\sum_j \omega_{ij}x_j + \sum_j \frac{\partial}{\partial x_j} R_{ij}^{-1}(x)], \quad g(x,t) := \sqrt{\beta'(t)R^{-1}(x)}. \tag{18}$$

Since our variational function class of the forward process defined in Eq. 12 is theoretically guaranteed to approach Gaussian when T is large, the first term of $\mathcal{E}^\infty(x)$ is close to a small constant under Eq. 18. Therefore, we only need to investigate the second term (equivalent to the implicit score matching (Hyvärinen and Dayan, 2005)), which depends on both the parameterized f, g and the score function. Finally, learning f and g opens the opportunity of adding additional **regularization penalties** to filter out irregular forward paths in the extended variational path space. Similar techniques have been applied in continuous normalizing flows (Finlay et al., 2020). Preliminary exploration on applying regularization to FP-Diffusion models is clarified in the experimental section.

3.3. A simplified formula of FP-Diffusion

Although we can always numerically simulate the SDE to a given time t, the empirical success of the Monte-Carlo training of (14) in (Ho et al., 2020) (see also (7) of (Song et al., 2020a)) indicates the importance of obtaining explicit solutions for direct sampling. In this section, we derive the solution formula for a simplified version of X_t defined in Eq. 12 and implement it on the image generation task.

To obtain the closed-form expression of the transition probabilistic density function for the forward process X_t, we assume that $R^{-1}(x)$ of Eq. 12 is a constant symmetric positive-definite matrix independent of the spatial variable x. Then within the linear SDE region (Särkkä and Solin, 2019), we have the following characterization of the marginal distributions (see Appendix A for a full derivation):

Theorem 3.5. *Suppose the forward diffusion process X_t starting at X_0 satisfies the following linear stochastic differential equation:*

$$dX_t = \frac{1}{2}\beta'(t)[-R^{-1}X_t - 2\omega X_t]dt + \sqrt{\beta'(t)R^{-1}}dW_t, \tag{19}$$

for symmetric positive-definite R and anti-symmetric ω. Then the marginal distribution of X_t at arbitrary time $t > 0$ follows the Gaussian distribution:

$$X_t \sim \mathcal{N}(e^{(-\frac{1}{2}R^{-1}-\omega)\beta(t)}X_0, \mathbf{I} - e^{-\beta(t)R^{-1}}).$$

In practice, we set ω in Eq. 11 to be an anti-symmetric matrix, and name it by the **FP-Drift** parameterization. On the other hand, R^{-1} in Eq. 10 is set to be a symmetric positive-definite matrix, and we name it by the **FP-Noise** parameterization. Both the anti-symmetric and symmetric matrices are realized through the matrix exponential map. We leave the implementation details in Appendix A.

4. Experiment

We first use a synthetic 3D dataset to illustrate the significance of parameterizing the forward process adapting to the data distribution, then validate the effectiveness of our FP-Diffusion model on standard image generation tasks.

4.1. Flexible SDEs learned from Synthetic 3D examples

According to the low-dimensional manifold hypothesis (Fefferman et al., 2016), the real data distribution concentrates on a low-dimensional sub-manifold. However, during the generation phase, the dimension of the ambient space we sample from is usually much higher. To fill in the gap, FP-Diffusion plays a nontrivial role. More precisely, note that only the diffusion part of Eq. 12 can blur the data sub-manifold to fill in the high-dimensional ambient space, which causes a distinction between the directions tangent to the data and the remaining normal directions during the (anisotropic) diffusion process. Since it is impossible to directly detect the complex data manifold, we design a simplified scenario to demonstrate how the parameterized diffusion process enhances generation.

Table 2. Results on CIFAR10. * denotes the results we reproduce locally.

Model	FID ↓	NLL↓
DDPM++ cont. (deep, VP) (Song et al., 2020a)	2.95*	3.13*
NCSN++ cont. (deep, VE) (Song et al., 2020a)	2.72*	-
DDPM (Zhang and Chen, 2021)	3.17	$\leq$ 3.75
Improved-DDPM (Nichol and Dhariwal, 2021)	2.90	3.37
LSGM (Vahdat et al., 2021)	2.10	$\leq$ 3.43
LSGM-100M (Dockhorn et al., 2022)	4.60	$\leq$ 2.96
CLD-SGM (Dockhorn et al., 2022)	2.25	$\leq$ 3.31
DiffFlow (Zhang and Chen, 2021)	14.14	3.04
FP-Drift (Joint)	4.17	3.30
FP-Noise (Joint)	3.30	3.25
FP-Drift (Mix)	2.99	3.28
FP-Noise (Mix)	2.87	3.20

Assume the data lies in $\mathbf{R}^3$, and its distribution follows a 2-dimensional Gaussian concentrated at a given hyper-

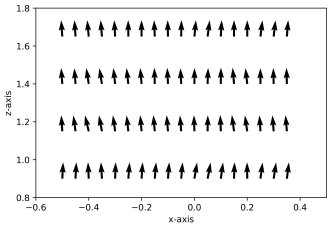


Figure 2. VP

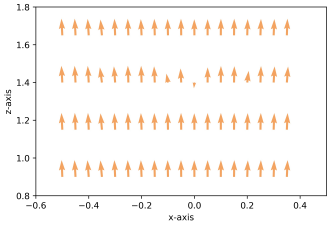


Figure 3. Non-reg. FP

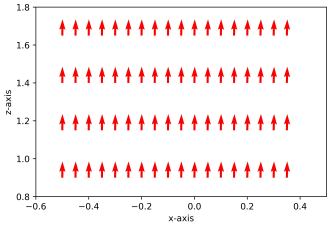


Figure 4. Reg. FP

Figure 5. Vector fields projected into 2D-section of three SDEs

plane. Obviously, the simplest way to generate the 2-dimensional Gaussian is to directly project random points sampled from 3-dimensional Gaussian to this plane. To make it rigorous, we consider the optimal transport problem from the 3-dimensional Gaussian distribution to the 2-dimensional Gaussian. Define the cost function as $c(x, y) := \|x - y\|^2$, then the Wasserstein distance between $\mathcal{N}(0, \mathbf{I})$ and $\mathcal{N}(\mu, \Sigma)$ (Mallasto and Feragen, 2017) equals $\mathcal{W}_2(\mathcal{N}(0, \mathbf{I}, \mathcal{N}(\mu, \Sigma)) = \|\mu\|^2 + \mathrm{Tr}(\mathbf{I} + \Sigma - 2\Sigma^{1/2})$. It implies that the corresponding optimal transport map $\nabla\phi$ is exactly the vertical projection.

For our FP-Diffusion model, the probabilistic flow of the generating process depends on the parameterized forward process. Then, to optimize the forward paths, we add **regularization** terms into the original score-matching loss. From (13) of (Song et al., 2020a), the vector field of the probability flow ODE is

$$v_{pc}(x,t) := f(x,t) - \frac{1}{2}\nabla \cdot [g^2(x,t)] - \frac{1}{2}g^2(x,t)\nabla \log p_t(x), x \in \mathbf{R}^3. \quad (20)$$

To control variables, We perform the experiment under three circumstances: 1. a fixed VP forward process (Eq. 1); 2. our parameterized forward process with no regularization; 3. our parameterized forward process with regularization terms. The regularization penalties are imposed on the vector field (Eq. 20), which are adapted from section 4 of (Finlay et al., 2020): $L_{reg}(f, g) = \lambda_1 \int \|v_{pc}(s)\|^2 ds + \lambda_2 \mathbb{E}_{\epsilon\sim\mathbf{N}(0,1)} \left\|\epsilon^T v_{pc}(s)\right\|^2 ds$. Notice that this term only regularizes parameters from f and g in the **forward process**.

Table 1. NLLs on MNIST

Model	NLL ↓
RealNVP (Dinh et al., 2016)	1.06
Glow (Kingma and Dhariwal, 2018)	1.05
FFJORD (Grathwohl et al., 2018)	0.99
ResFlow (Chen et al., 2019)	0.97
DiffFlow (Zhang and Chen, 2021)	0.93
FP-Drift (Mix)	1.01

Figure 6. MNIST and CIFAR10 samples

After training, we check whether the direction of the learned v_{pc} is aligned with the ground-truth projection vector field. For the projection map $\nabla\phi$ from the 3-dimensional Gaussian to 2-dimensional Gaussian supported at the plane $:z = 2$, the corresponding vector field at a spatial point $x = (x, y, z) \in \mathbb{R}^3$ equals:

$$\begin{aligned} v_{\text{proj}}(x,t) &:= (0,0,-1) \ \text{ if } \ z > 2, \\ \text{and } \ v_{\text{proj}}(x,t) &:= (0,0,1) \quad \text{if } \ z < 2. \end{aligned} \quad (21)$$

The 2D visualization results of our comparative experiments are summarized in Fig. 5. Since the 'ground-truth' vector field (Eq. 21) is strictly vertical, it's enough to plot the $x - z$ projection of the trained three vector fields for the three scenarios. From Fig. 5, although our flexible diffusion method (b) is already more vertical than the forward-fixed VP (ddpm) model (a), the flexible model trained with regularization (c) is closer to vertical lines. In Appendix B, we also sample the integration trajectories of the trained vector fields for comparison (see Fig. 5).

4.2. Image Generation

In this section, we demonstrate the generative capacity of our FP-Diffusion models on two common image datasets:

MNIST (LeCun, 1998) and CIFAR10 (Krizhevsky et al., 2009).

Training strategy. The flexible FP-Diffusion framework is designed to simultaneously learn a suitable forward diffusion process dependent on the data distribution as well as the corresponding reverse-time process. However, for some complex scenarios like image generation, it is challenging to balance the optimization of the forward and the backward processes. To compromise these two parts, we propose a two-stage training strategy. Particularly, in the first stage, we jointly optimize the parameters from both the FP-Diffusion forward process and the backward score neural network; in the second stage, we freeze all parameters from the FP-Diffusion and only tune the score neural network in the same way as prevailing score-based training approaches (Ho et al., 2020; Song and Ermon, 2019; Song et al., 2020a). Note that the two-stage strategy also makes the flexible diffusion **salable**, in the sense that after the first stage, the parameters contained in the forward process are fixed and won't be counted in the gradient computational graph. Moreover, during the sampling process, only the score neural network is implemented.

Implementation Details. For the forward diffusion process, we choose a linearly increasing time scheduler $\beta(t)$ (same as the VP-SDE setting in (Song et al., 2020a)), where $t \in [0, T]$ is a continuous time variable. To estimate the gradient vector field in the reverse-time process, we train a time-dependent score network $s_\theta(x(t), t)$ as described in Eq. 16. We adopt the same U-net style architecture used in (Ho et al., 2020) and (Song et al., 2020a) as our backbone neural network. Both the FP-Drift model and the FP-Noise model are implemented in two training paradigms: 1) **Joint Training**: the parameterized FP-Diffusion model and the score network are jointly optimized for 1.2M iterations; 2) **Mix Training**: following the proposed two-stage training strategy, we separately train the model for 600k iterations in both stages, and the batch size is set to be 96 on all datasets. Following (Song et al., 2020a), we apply the Euler-Maruyama method in our reverse-time SDEs for sampling images, where the discretization steps equal 1000. All the experiments are conducted on 4 Nvidia Tesla V100 16G GPUs. We provide further implementation details in Appendix B.

Results. We show the sampled images generated by our FP-Noise (Mix training) model in Fig. 6. According to Eq. 20, the negative log-likelihood (NLL) are explicitly caculated in bits per dimension for our models by the instantaneous change of variables formula (Grathwohl et al., 2018). Then we list the NLL metrics of our models in Tab. 1 and Tab. 2. On MNIST, our FP-Drift model achieves comparable performance in terms of NLL, compared to five standard flow-based models (including DiffFlow (Zhang and Chen, 2021)). On CIFAR10, both the FP-Drift (Mix training) and the FP-Noise (Mix training) models achieve a competitive performance compared to the state-of-the-art (SOTA) diffusion models. These results illustrate the strong capacity of FP-Diffusion in density estimation tasks.

To quantitatively evaluate the quality of the sampled images, we also report the Fenchel Inception Distance (FID) (Heusel et al., 2017) on CIFAR10. As shown in Tab. 2, the two variants of our FP-Diffusion model, FP-Drift (Mix) and FP-Noise (Mix), outperform DDPM (Ho et al., 2020) and Improved-DDPM (Nichol and Dhariwal, 2021) in FID and have a comparable performance with DDPM++ cont. (deep, VP) and NCSN++ cont. (deep, VE) (Song et al., 2020a). We notice that only LSGM and CLD-SGM have obviously better FID values than other models (including us). However, LSGM (Vahdat et al., 2021) adopts a more complicated framework and a large model with $\approx 475M$ parameters to achieve its high performance. With a comparable parameter size ($\approx 100M$), our models could achieve a significantly better FID score than LSGM ("LSGM-100M"). CLD-SGM builds its diffusion model upon a larger phase space with a special training objective (given the data point $x \in \mathbb{R}^n$, its phase space corresponding point (x, v) belongs to $\mathbb{R}^{2n}$), which leads to a more expressive optimization space but brings extra computational cost as well. We leave testing our FP-Diffusion model on phase space (defined in Corollary 3.4) in future works. It should also be noted that we use a smaller batch size (96) compared to other baseline diffusion models (128) to train our models due to limited computational resources, which may influence our empirical performance. We also report the performance of our two model variants in two training paradigms in Tab. 2. The model variants with the joint training paradigm consistently achieve a better performance, demonstrating the necessity of the two-stage training strategy. A possible reason of this phenomenon is that it may be difficult for score models to match the reverse process of a dynamical forward process, so we need to tune the score model with extra training steps after fixing a suitable forward process.

5. Conclusion

We propose FP-Diffusion model, a novel method that parameterizes the spatial components of the diffusion (score) model with theoretical guarantees. This approach combines insights from Riemannian geometry and Hamiltonian Monte-Carlo methods to obtain a complete forward diffusion parameterization that plays a nontrivial role from the variational optimization perspective. Empirical results on specially-designed datasets and standard benchmarks confirm the effectiveness of our method. However, how to efficiently optimize FP-Diffusion remains a critical challenge, which opens the door for promising future research.

References

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2020a.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

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*, pages 2837–2845, 2021.

Weitao Du, He Zhang, Yuanqi Du, Qi Meng, Wei Chen, Bin Shao, and Tie-Yan Liu. Equivariant vector field network for many-body system modeling, 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*, 2020.

Songxiang Liu, Yuewen Cao, Dan Su, and Helen Meng. Diffsvc: A diffusion probabilistic model for singing voice conversion. *arXiv preprint arXiv:2105.13871*, 2021.

Minkai Xu, Lantao Yu, Yang Song, Chence Shi, Stefano Ermon, and Jian Tang. Geodiff: A geometric diffusion model for molecular conformation generation. *arXiv preprint arXiv:2203.02923*, 2022.

Emiel Hoogeboom, Vıctor Garcia Satorras, Clément Vignac, and Max Welling. Equivariant diffusion for molecule generation in 3d. In *International Conference on Machine Learning*, pages 8867–8887. PMLR, 2022.

Arne Schneuing, Yuanqi Du, Charles Harris, Arian Jamasb, Ilia Igashov, Weitao Du, Tom Blundell, Pietro Lió, Carla Gomes, Max Welling, et al. Structure-based drug design with equivariant diffusion models. *arXiv preprint arXiv:2210.13695*, 2022.

Giuseppe Da Prato. *Introduction to stochastic analysis and Malliavin calculus*, volume 13. Springer, 2014.

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.

Arash Vahdat and Jan Kautz. NVAE: A deep hierarchical variational autoencoder. In *Neural Information Processing Systems (NeurIPS)*, 2020.

Tim Dockhorn, Arash Vahdat, and Karsten Kreis. Score-based generative modeling with critically-damped langevin diffusion. In *International Conference on Learning Representations (ICLR)*, 2022.

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

Hariharan Narayanan and Sanjoy Mitter. Sample complexity of testing the manifold hypothesis. *Advances in neural information processing systems*, 23, 2010.

Chris Finlay, Jörn-Henrik Jacobsen, Levon Nurbekyan, and Adam Oberman. How to train your neural ode: the world of jacobian and kinetic regularization. In *International Conference on Machine Learning*, pages 3154–3164. PMLR, 2020.

Derek Onken, S Wu Fung, Xingjian Li, and Lars Ruthotto. Ot-flow: Fast and accurate continuous normalizing flows via optimal transport. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 2021.

E. P. Hsu. *Stochastic Analysis on Manifolds*. Stochastic Analysis on Manifolds, 2002.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019.

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

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Tao Yang, Xuanchi Ren, Yuwang Wang, Wenjun Zeng, and Nanning Zheng. Towards building a group-based unsupervised representation disentanglement framework. In

International Conference on Learning Representations, 2021.

Xuanchi Ren, Tao Yang, Yuwang Wang, and Wenjun Zeng. Learning disentangled representation by exploiting pretrained generative models: A contrastive learning view. In *International Conference on Learning Representations*, 2021.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015.

Sacha Friedli and Yvan Velenik. *Statistical Mechanics of Lattice Systems: A Concrete Mathematical Introduction*. Cambridge University Press, 2017. ISBN 978-1-107-18482-4. doi: 10.1017/9781316882603.

Sybren Ruurds De Groot and Peter Mazur. *Non-equilibrium thermodynamics*. Courier Corporation, 2013.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv:2010.02502*, October 2020b. URL https://arxiv.org/abs/2010.02502.

Daniel Watson, William Chan, Jonathan Ho, and Mohammad Norouzi. Learning fast samplers for diffusion models by differentiating through sample quality. *ArXiv*, abs/2202.05830, 2022.

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

Fan Bao, Chongxuan Li, Jun Zhu, and Bo Zhang. Analytic-dpm: an analytic estimate of the optimal reverse variance in diffusion probabilistic models. *arXiv preprint arXiv:2201.06503*, 2022.

Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. *Advances in Neural Information Processing Systems*, 34, 2021.

Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *J. Mach. Learn. Res.*, 6:695–709, 2005. URL http://jmlr.org/papers/v6/hyvarinen05a.html.

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

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*, pages 574–584. PMLR, 2020c.

Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

Qinsheng Zhang and Yongxin Chen. Diffusion normalizing flow. *Advances in Neural Information Processing Systems*, 34, 2021.

Mark Girolami and Ben Calderhead. Riemann manifold langevin and hamiltonian monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2):123–214, 2011.

Michael Betancourt. A general metric for riemannian manifold hamiltonian monte carlo. In *International Conference on Geometric Science of Information*, pages 327–334. Springer, 2013.

Michael Betancourt. A conceptual introduction to hamiltonian monte carlo. *arXiv preprint arXiv:1701.02434*, 2017.

Christof Seiler, Simon Rubinstein-Salzedo, and Susan Holmes. Positive curvature and hamiltonian monte carlo. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper/2014/file/beed13602b9b0e6ecb5b568ff5058f07-Paper.pdf.

Joachim Weickert. *Anisotropic diffusion in image processing*, volume 1. Teubner Stuttgart, 1998.

P Perona and J Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(7):629–639, 1990. ISSN 0162-8828. doi: 10.1109/34.56205.

Luis Alvarez, Pierre-Louis Lions, and Jean-Michel Morel. Image selective smoothing and edge detection by nonlinear diffusion. ii. *SIAM Journal on numerical analysis*, 29: 845–866, 06 1992. doi: 10.1137/0729052.

F. Y. Wang. Analysis for diffusion processes on riemannian manifolds. 2014.

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

S Särkkä and A. Solin. *Applied Stochastic Differential Equations*. 2019.

Charles Fefferman, Sanjoy Mitter, and Hariharan Narayanan. Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29(4):983–1049, 2016.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018.

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

Ricky TQ Chen, Jens Behrmann, David K Duvenaud, and Jörn-Henrik Jacobsen. Residual flows for invertible generative modeling. *Advances in Neural Information Processing Systems*, 32, 2019.

Anton Mallasto and Aasa Feragen. Learning from uncertain curves: The 2-wasserstein metric for gaussian processes. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 5665–5674, 2017.

Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

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.

Luc Rey Bellet. Ergodic properties of markov processes. In *Open quantum systems II*, pages 1–39. Springer, 2006.

Lars Hörmander. Hypoelliptic second order differential equations. *Acta Mathematica*, 119:147–171, 1967.

A. Theory

A.1. Discussion of Section 3.1

A.1.1. Remark on the theoretical properties of Hamiltonian dynamics

Suppose X_t follows the Hamiltonian dynamics (11), then

$$dH(X_t) = \nabla H(X_t)\omega\nabla H(X_t)dt \equiv 0,$$

by the anti-symmetry of ω. Therefore, the Hamiltonian dynamics without random perturbations is a deterministic motion that can explore within a constant Hamiltonian (energy) surface. It means that, only by adding a diffusion term, the Hamiltonian dynamical system is able to traverse different energy levels.

A.1.2. Verification of Theorem 3.3

We will verify the theorem under the more general case, when $H(x) = \frac{m}{2}x^2$. The corresponding stationary distribution is the scaled Gaussian $\mathcal{N}(0, m\mathbf{I})$, where $m > 0$ is the scale constant. In this case, Eq. 12 is modified to:

$$\begin{aligned} dX_t =& \frac{m}{2}[-\sum_j R_{ij}^{-1}(X_t)\cdot(X_t)_j - 2\sum_j \omega_{ij}\cdot(X_t)_j \\ &+ \sum_j \frac{\partial}{\partial x_j}R_{ij}^{-1}(X_t)]dt + \sqrt{R^{-1}(X_t)}dW_t. \end{aligned} \tag{22}$$

Note that only the drift term is scaled by m.

Proof. Since the covariance matrix of the diffusion part is positive-definite, the forward process Eq. 22 satisfies the Feller property and the existence and uniqueness of the stationary distribution are guaranteed (see (Wang, 2014)). By the Fokker-Plank-Kolmogorov equation, the stationary distribution $p_s(x)$ of Eq. 22 should satisfy

$$0 = -\sum_i \frac{\partial}{\partial x_i}[f_i(x,t)p_s(x)] + \frac{1}{2}\frac{\partial^2}{\partial x_i \partial x_j}[(gg^T)_{ij}p_s(x)], \tag{23}$$

where we set $f(x,t) := \frac{m}{2}[-\sum_j R_{ij}^{-1}(x)\cdot x_j - 2\sum_j \omega_{ij}\cdot x_j + \sum_j \frac{\partial}{\partial x_j}R_{ij}^{-1}(x)]$ and $g(x,t) := \sqrt{R^{-1}(x)}$. To check whether $e^{-\frac{m}{2}x^2}$ satisfies condition (23), notice that by the anti-symmetry of ω_{ij}, we automatically have

$$\sum_i\sum_j \frac{\partial}{\partial x_i}(\omega_{ij}x_j e^{-\frac{m}{2}x^2}) = -\sum_i\sum_j \omega_{ij}x_i x_j e^{-\frac{m}{2}x^2} = 0.$$

On the other hand,

$$\begin{aligned} &\sum_i\sum_j \frac{\partial^2}{\partial x_i \partial x_j}[R_{ij}^{-1}(x)e^{-\frac{m}{2}x^2}] \\ &= -m\mathrm{Tr}(R_{ij}^{-1}(x))e^{-\frac{m}{2}x^2} + m^2\sum_i\sum_j R_{ij}^{-1}(x)x_i x_j e^{-\frac{m}{2}x^2} \\ &- \sum_i\sum_j \frac{\partial}{\partial x_j}(R_{ij}^{-1}(x))(\frac{\partial}{\partial x_i}e^{-\frac{m}{2}x^2}) \\ &+ \sum_i\sum_j \frac{\partial^2}{\partial x_i \partial x_j}(R_{ij}^{-1}(x))e^{-\frac{m}{2}x^2} \\ &- m\sum_i\sum_j \frac{\partial}{\partial x_j}(R_{ij}^{-1}(x))x_i e^{-\frac{m}{2}x^2} \\ =& -m\mathrm{Tr}(R_{ij}^{-1}(x))e^{-\frac{m}{2}x^2} + m^2\sum_i\sum_j R_{ij}^{-1}(x)x_i x_j e^{-\frac{m}{2}x^2} \\ &+ \sum_i \frac{\partial}{\partial x_i}[\sum_j \frac{\partial}{\partial x_j}R_{ij}^{-1}(x)e^{-\frac{m}{2}x^2}] \\ &- m\sum_i\sum_j \frac{\partial}{\partial x_j}R_{ij}^{-1}(x)x_i e^{-\frac{m}{2}x^2}. \end{aligned}$$

Therefore, the last thing to check is that

$$\begin{aligned} \sum_i \frac{\partial}{\partial x_i}[\sum_j R_{ij}^{-1}(x)x_j e^{-\frac{m}{2}x^2}] = \mathrm{Tr}(R_{ij}^{-1}(x))e^{-\frac{m}{2}x^2} - \\ \sum_i\sum_j [mR_{ij}^{-1}(x)x_i x_j + \frac{\partial}{\partial x_j}R_{ij}^{-1}(x)x_i]e^{-\frac{m}{2}x^2}, \end{aligned}$$

which is obviously true, since the diffusion matrix R_{ij}^{-1} is symmetric. Combining the above, we have proved that Eq. 23 holds if $p_s(x) \propto e^{-\frac{m}{2}x^2}$. □

A.1.3. Completeness of FP-Diffusion parameterization

From the last section's derivation, we can deduce the following corollary:

Corollary A.1. *Consider the following SDE:*

$$\begin{aligned} dX_t =& A(X_t)dt - \frac{1}{2}R^{-1}(X_t)\cdot X_t dt \\ &+ (\nabla\cdot R^{-1}(X_t))\cdot X_t dt + \sqrt{R^{-1}(X_t)}dW_t, \end{aligned} \tag{24}$$

and let the spatial function $A(x)$ be a linear function. Suppose we know its stationary distribution is standard Gaussian, then

$$A(x) = -\sum_j \omega_{ij}\cdot x_j,$$

for some anti-symmetric matrix ω.

Proof. In fact, every linear operator A can be decomposed into a symmetric part plus an anti-symmetric part:

$$A = \underbrace{\frac{A+A^T}{2}}_{\text{symmetric}} + \underbrace{\frac{A-A^T}{2}}_{\text{anti-symmetric}}.$$

Let $\omega = \frac{A-A^T}{2}$. Then we only need to prove that $A + A^T$ equals zero, if X_t converges to Gaussian.

From the proof of theorem 3.3, we extract the fact that if $p_s(x) \propto e^{-\frac{1}{2}x^2}$,

$$\sum_i \frac{\partial}{\partial x_i}[(A+A^T)_{ij} \cdot x_j e^{-\frac{1}{2}x^2}] = 0,$$

then

$$\sum_{i,j}[(A+A^T)_{ij} \cdot \frac{\partial}{\partial x_i}\frac{\partial}{\partial x_j}(e^{-\frac{1}{2}x^2})] = 0,$$

for all $x = (x_1, \ldots, x_n)$. Note that

$$\frac{\partial}{\partial x_i}\frac{\partial}{\partial x_j}(e^{-\frac{1}{2}x^2}) = (x_i x_j - \delta_{ij})e^{-\frac{1}{2}x^2}.$$

Since $A + A^T$ is symmetric (**doesn't hold for arbitrary linear operator**), it implies that $A + A^T \equiv 0$. □

A.1.4. Anisotropic diffusion on low dimensional data manifold

In this section, we give an informal discussion on how an anisotropic diffusion starting at a low-dimensional data manifold mixes with its own stationary distribution (supported in the high dimension ambient space).

Assume the marginal distribution of the diffusion process X_t concentrates on a low dimensional manifold $M \hookrightarrow \mathbb{R}^n$ at a given time. Moreover, suppose X_t already achieves the Gaussian stationary distribution on M (defined with respect to the Laplacian operator of M). Now we want to informally investigate the most efficient way for X_t to diffuse out of the low dimensional sub-manifold to the ambient space. By localizing in the Riemannian normal coordinates and by arranging the coordinates indexes, we can further assume that M is isometric to the hyperplane of $\mathbb{R}^n$ defined by

$$M = \{x \in \mathbb{R}^n | x = (x_1, \ldots, x_p, 0, \ldots, 0)\}.$$

Then the coordinate components of each point $x \in \mathbb{R}^n$ can be decomposed into the tangential directions and the normal directions with respect to M:

$$x \in (\underbrace{x_1, \ldots, x_p}_{\text{tangent to M}}, \underbrace{x_{p+1}, \ldots, x_n}_{\text{normal to M}}).$$

Under the above conditions, we are ready to compare the convergence rate (to the high-dimensional stationary Gaussian distribution of $\mathbb{R}^n$) of different forward diffusions defined in (10). For a fair comparison, we set the norm of the noise matrix to be one: $\left\|R^{-1}\right\|_2 \equiv \sqrt{n}$. Otherwise, the convergence can always be accelerated by increasing the noising scale ($\left\|R^{-1}\right\|_2 \to \infty$).

Under our normal coordinates, the forward diffusion can be decomposed into two parts :$X(t) = X_{tan}(t) + X_{nor}(t)$. For simplicity, suppose R^{-1} is a diagonal matrix, then the tangential part and the normal part of $X(t)$ is completely decoupled. In other word,

$$X^i_{tan}(t) = \frac{1}{2}[-R^{-1}_{ii} \cdot (X_t)^i]dt + \sqrt{R^{-1}_{ii}}dW^i_t, \;\; 1 \le i \le p$$

is a diffusion process on M. Therefore, $(X_{tan}(t), X(t))$ is indeed a Markov coupling. Suppose $X_{tan}(t)$ at $t = 0$ already converges to its stationary distribution (low dimensional Gaussian), then by Ito's formula,

$$\begin{aligned}
&d(X_{tan}(t) - X(t))^2 \\
&= dX^2_{nor}(t) \\
&= 2X_{nor}(t)(\frac{1}{2}[-R^{-1}_{nor} \cdot X_{nor}(t)]dt + \sqrt{R^{-1}_{nor}}dW_t) + \text{Tr}(R^{-1}_{nor})dt.
\end{aligned}$$

Taking the expectation of both sides, it implies that

$$\frac{d\mathbb{E}X^2_{nor}(t)}{dt} = -\mathbb{E}R^{-1}_{nor} \cdot X^2_{nor}(t) + \text{Tr}(R^{-1}_{nor}).$$

Let r_{min} denote the minimal eigenvalue of the normal part of R^{-1}, then

$$\frac{d\mathbb{E}X^2_{nor}(t)}{dt} \le -r_{min}\mathbb{E}X^2_{nor}(t) + \text{Tr}(R^{-1}_{nor}).$$

Applying Grönwall's inequality and note that $X_{nor}(0) = 0$, we have

$$\mathbb{E}X^2_{nor}(t) \le e^{-r_{min} \cdot t} \cdot \text{Tr}(R^{-1}_{nor})t.$$

The above gives an upper bound on the convergence speed of the coupling $(X_{tan}(t), X(t))$ with respect to the W_2 distance (see (Wang, 2014)). Since the stationary distribution of $X(t)$ is exactly the high dimensional Gaussian distribution (the diffusion model's prior distribution), we hope the convergence rate to be as fast as possible (given a fixed noising scale). For the VP-Diffusion,

$$R^{-1}_{nor} \equiv \text{diag}\{1, \ldots, 1\}.$$

However, in FP-Diffusion model, the diagonal elements of R^{-1}_{nor} are allowed to be inhomogeneous and greater than one (under the condition that $\text{Tr}(R^{-1}_{nor}) < n$). This will lead to a smaller r_{min}, which will speed up the convergence rate by our analysis.

A.1.5. Verification of Corollary 3.4

The intuition of Corollary 3.4 can be stated as follows: To guarantee the geometric ergodicity property of FP-Diffusion on the **phase space**, we need enough noise such that the diffusion process can transverse the whole space. Suppose $R^{-1}(x)$ degenerates along the i-th direction (corresponding to a zero eigenvalue), then no randomness (noise) is imposed on this direction.

To remedy the issue, we require the symplectic form ω to be non-zero along the i-th direction, which makes it possible to mix the noise originated along other directions (where $R^{-1}(x)$ is strictly positive-definite) with the i-th direction. Now we give the formal proof:

Proof. We only prove for the simplified case when A and B are both diagonal matrices with two sets of positive eigenvalues $\{a_i\}_{i=1}^d$, $\{b_i\}_{i=1}^d$. The general situation can be handled by trivial linear transformation. By proposition 8.1 of (Bellet, 2006), the proof boils down to prove that the Hörmander's condition (Hörmander, 1967) holds for the forward process X_t. When $R^{-1}(x)$ is a constant matrix, the infinitesimal generator L of (12) is:

$$L = \sum_i \sum_j \frac{1}{2}[-R_{ij}^{-1} - 2\omega_{ij}]x_j \frac{\partial}{\partial x_j} + \frac{1}{2}\sum_{ij} R_{ij}^{-1}\frac{\partial^2}{\partial x_i \partial x_j}.$$

For notation simplicity, denote $x := (u, v) \in \mathbb{R}^{2d}$, where $u, v \in \mathbb{R}^d$. To put the second-order differential operator L in Hörmander's form, set

$$Y_j(u, v) = -\frac{1}{2}\sqrt{b_j}\frac{\partial}{\partial v_j}, \quad 1 \le j \le n,$$

and

$$Y_0(u.v) = \sum_i (-a_i v_i \frac{\partial}{\partial u_i} + a_i u_i \frac{\partial}{\partial v_i}).$$

Then it suffices to show that the vector fields $\{[Y_0, Y_j], Y_j\}_{1\le j\le d}$ span the whole $\mathbb{R}^{2d}$. By direct calculation,

$$[Y_0, Y_j] = \frac{1}{2}a_j\sqrt{b_j}\frac{\partial}{\partial u_j},$$

for $\forall j$. Therefore, we conclude that the Hörmander's condition holds for X_t. Then the ergodic proposition 8.1 of (Bellet, 2006) implies that the forward diffusion X_s converges to the standard Gaussian distribution. □

Remark A.2. A recent study (Dockhorn et al., 2022) proposed to improve the diffusion model by enlarging the spatial space (where the generated samples lie in) to the "phase" space: $x \to (x, v)$. Then the corresponding joint forward diffusion (x_t, v_t) satisfies the Critically-Damped Langevin diffusion:

$$\begin{pmatrix} dx_t \\ dv_t \end{pmatrix} = \begin{pmatrix} M^{-1}v_t \\ -x_t \end{pmatrix} dt + \begin{pmatrix} \mathbf{0}_d \\ -\Gamma M^{-1}v_t \end{pmatrix} dt + \begin{pmatrix} 0 \\ \sqrt{2\Gamma} \end{pmatrix} dW_t. \tag{25}$$

If the coupling mass $M = 1$, the drift part of Eq. 25 can be decomposed to a symmetric part R^{-1} and an **non-trivial** anti-symmetric part ω of (19) by setting:

$$R^{-1} := \begin{pmatrix} 0, & 0 \\ 0, & 2\Gamma I \end{pmatrix}, \quad \omega := \begin{pmatrix} 0, & -I \\ I, & 0 \end{pmatrix}.$$

It's straightforward to check that they rigorously fit the conditions of Corollary 3.4. Therefore, we conclude from Corollary 3.4 that the Damped Langevin diffusion converges to the standard Gaussian distribution of the enlarged phase space $(x, v) \in \mathbb{R}^{2d}$, which coincides with the results of Appendix B.2 in (Dockhorn et al., 2022).

A.2. Discussion of Section 3.2

In this section, following the arguments from (Huang et al., 2021), we demonstrate how to estimate the score gradient vector field $\nabla \log p(x)$ by the analytically tractable conditional score gradient vector field (conditioned on a previous time).

To prove (16), by adapting Eq. 31 of (Huang et al., 2021), it's enough to show that

$$\mathbb{E}_{X_t}[s_\theta^T(X_t, t)\cdot\nabla \log p_t(X_t)] = \mathbb{E}_{X_s, X_t}[s_\theta^T(X_t, t)\cdot\nabla \log p_t(X_t|X_s)].$$

Transforming the expectation to probabilistic integration, we have

$$\begin{aligned}
&\mathbb{E}_{X_t}[s_\theta^T(X_t, t)\cdot\nabla \log p_t(X_t)] && (26)\\
&= \int p_t(x) s_\theta^T(x, t)\cdot\nabla \log p_t(x) dx && (27)\\
&= \int s_\theta^T(x, t)\int \nabla p_t(x|x_s) p_s(x_s) dx dx_s && (28)\\
&= \int\int p_s(x_s) p_t(x|x_s)\nabla p_t(x|x_s) dx dx_s && (29)\\
&= \mathbb{E}_{X_s, X_t}[s_\theta^T(X_t, t)\cdot\nabla \log p_t(X_t|X_s)], && (30)
\end{aligned}$$

for $0 \le s < t$. By quadratic expanding $\mathbb{E}_{X_t}\left\|\mathbf{s}_\theta(X_t, t) - \nabla \log p_t(X_t)\right\|^2$ and plugging in (26), equality (16) follows directly.

To implement our discretized FP-diffusion forward diffusion, we usually choose $s = t - 1$, the immediate time step before t. Then from $t-1$ to t, the conditional score gradient vector field of $p_t(x_t|x_{t-1})$ is the Gaussian score function, which is analytically tractable.

A.3. Discussion of Section 3.3

In this section, we prove Theorem 3.5 by applying Ito's formula and martingale representation theorem.

Recall that the time-change of Eq. 12 satisfies

$$dX_t = \beta'(t)(-\frac{1}{2}R^{-1} - \omega)X_t dt + \sqrt{\beta'(t)R^{-1}}dW_t, \tag{31}$$

where X_0 is a fixed point. Let $Y_t := e^{(\frac{1}{2}R^{-1}+\omega)\beta(t)}X_t$, then by Ito's formula,

$$Y_t = \int_0^t e^{(\frac{1}{2}R^{-1}+\omega)\beta(s)}\sqrt{\beta'(s)R^{-1}}dW_s. \quad (32)$$

From the martingale representation theorem, Y_t is a Gaussian random variable for each t. Therefore, to fully determine the distribution of X_t, we only need to calculate the expectation and variance formulas of X_t. By the definition of stochastic integration, we have

$$\mathbf{E}[X(t)] = e^{(-\frac{1}{2}R^{-1}-\omega)\beta(t)}X_0.$$

Utilizing the Ito's isometry to (32), we get

$$Var[Y_t] = \int_0^t e^{\beta(s)(R^{-1}+2\omega)}\beta'(s)R^{-1}ds.$$

Suppose $\omega = 0$, then

$$Var[X_t] = \mathbf{I} - e^{-\beta(t)R^{-1}},$$

where $\mathbf{I}$ denotes the identity matrix of $\mathbf{R}^d$. Suppose $R^{-1} = \mathbf{I}$, since the Lie bracket $[\mathbf{I}+2\omega, \mathbf{I}-2\omega] = 0$, we further obtain

$$Var[X_t] = \mathbf{I} - e^{-\beta(t)\mathbf{I}}.$$

In conclusion, we have proved Theorem 3.5.

A.4. How to parameterize symmetric and anti-symmetric matrix

To implement FP-Drift and FP-Noise models practically, we need to find an efficient way to parameterize positive-definite symmetric and anti-symmetric matrix.

Given a full-rank anti-symmetric matrix B, there always exist an orthogonal matrix P such that

$$B = P\text{diag}\left\{\begin{bmatrix}0 & \lambda_1\\ -\lambda_1 & 0\end{bmatrix}, \cdots \begin{bmatrix}0 & \lambda_n\\ -\lambda_n & 0\end{bmatrix}\right\}P^T,$$

where $\{\lambda_1, \ldots, \lambda_n\}$ are nonzero numbers. Then, the inverse of $\mathbf{I}+B$ (appeared in subsection A.4) is:

$$(B+I)^{-1} = P\text{diag}\left\{\begin{bmatrix}\frac{1}{1+\lambda_1^2} & \frac{-\lambda_1}{1+\lambda_1^2}\\ \frac{\lambda_1}{1+\lambda_1^2} & \frac{1}{1+\lambda_1^2}\end{bmatrix}, \cdots \begin{bmatrix}\frac{1}{1+\lambda_n^2} & \frac{-\lambda_n}{1+\lambda_n^2}\\ \frac{\lambda_n}{1+\lambda_n^2} & \frac{1}{1+\lambda_n^2}\end{bmatrix}\right\}P^T.$$

For positive-definite symmetric matrices, there always exist an orthogonal matrix P such that

$$R = P\text{diag}\{\lambda_1, \ldots, \lambda_n\}P^T,$$

where $\{\lambda_1, \ldots, \lambda_n\}$ are positive numbers.

To apply the above method, we only need to parameterize orthogonal matrices in an efficient and expressive way. By treating orthogonal matrices as elements in $SO(n)$ orthogonal group, we utilize the exponential map to parameterize orthogonal matrices P:

$$P = \exp H.$$

Note that H is an element that belongs to the lie algebra $so(n)$, which can be generated by upper triangular matrices.

B. Experiments

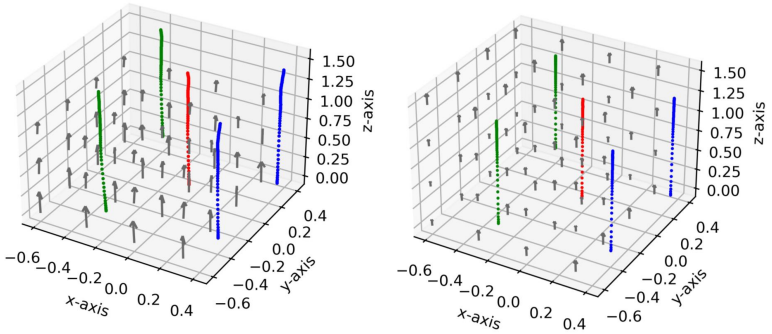
B.1. Learned FP SDEs from synthetic 3D examples

Figure 7 plots four 3D integration trajectories of the probabilistic flows (with respect to the fixed VP and learned FP-Diffusion models) starting at random initial positions. It's obvious that the trajectories of our flexible model are more straight than the fixed VP model, which demonstrates the power of selecting more regular generating paths of our FP-Diffusion model.

B.2. Image Generation

Implementation Details. Following (Ho et al., 2020) and (Song et al., 2020a), we rescale the range of the images into $[-1, 1]$ before inputting them into the model. In the FP-Diffusion model, $\beta(t)$ is an linearly increasing function with respect to the time t, i.e., $\beta(t) = \bar{\beta}_{min} + t(\bar{\beta}_{max} - \bar{\beta}_{min})$ for $t \in [0, 1]$. It's worth mentioning that DDPM adopts a discretization form of this time scheduler, where $\beta_i = \frac{\bar{\beta}_{min}}{N} + \frac{i-1}{N(N-1)}(\bar{\beta}_{max} - \bar{\beta}_{min})$. These two forms are actually equivalent when $N \to \infty$. For all experiments, we set $\bar{\beta}_{max}$ as 20 and $\bar{\beta}_{min}$ as 0.1, which are also used in (Ho et al., 2020) and (Song et al., 2020a). As discussed in A.4, we only need to parameterize the upper triangular matrices H and the diagonal elements $\Lambda = diag\{\lambda_1, \cdots, \lambda_n\}$ in the FP-Drift and FP-Noise models. Particularly, both H and Λ are initialized with a multivariate normal distribution, and we adopt a exponential operation on Λ to keep it a positive vector. As described in Section 4.2, we leverage a U-net style neural network to fit the score function of the reverse-time diffusion process. We keep the model architecture and the parameters of the score networks consistent with previous SOTA diffusion models (e.g., (Song et al., 2020a)) for a fair comparison. All models are trained with the Adam optimizer with a learning rate 2×10^{-4} and a batchsize 96.

In the MNIST experiment, we first train the whole model for 50k iterations and train the score model for another 250k iterations with our Mix training strategy. We report the NLL of the model based on the last checkpoint. In the CIFAR10 experiment, the training iterations of both stage 1 and stage 2 are 600k. We also report the FIDs and NLL of the model based on the last checkpoint.



(a) VP (b) Reg. FP

Figure 7. Integral trajectories of two SDEs

Results. We provide more random samples from our best FP-Drift model's checkpoint in Fig. 8. We also provide the learned forward process of FP-Noise model in Fig. 9.

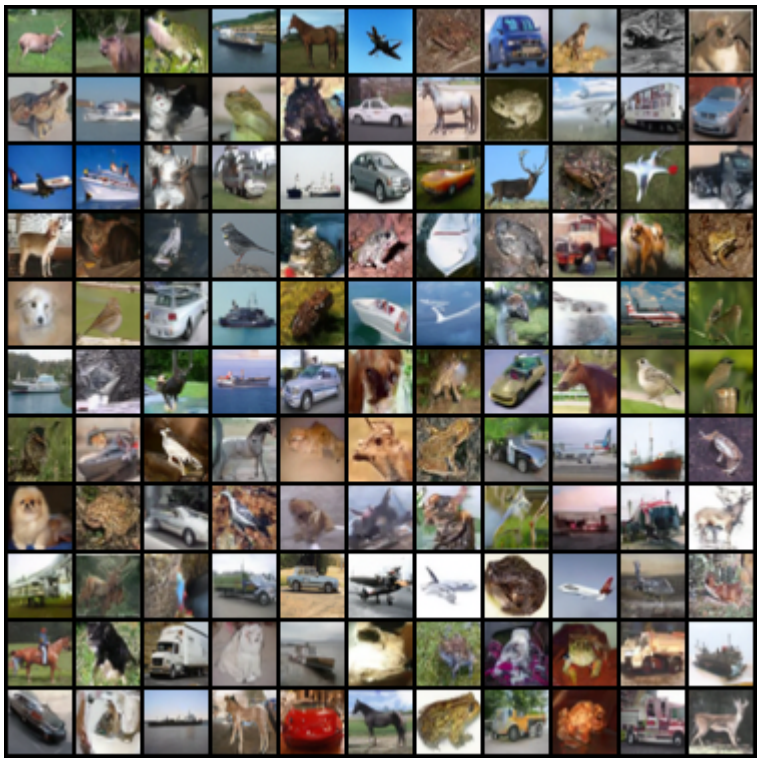


Figure 8. CIFAR-10 samples from FP-Drift

Figure 9. The learned forward process of FP-Noise on CIFAR-10