Understanding Diffusion Models by Feynman's Path Integral

Yuji Hirono [1] **Akinori Tanaka** [2 3 4] **Kenji Fukushima** [5]

Abstract

Score-based diffusion models have proven effective in image generation and have gained widespread usage; however, the underlying factors contributing to the performance disparity between stochastic and deterministic (i.e., the probability flow ODEs) sampling schemes remain unclear. We introduce a novel formulation of diffusion models using Feynman's path integral, which is a formulation originally developed for quantum physics. We find this formulation providing comprehensive descriptions of score-based generative models, and demonstrate the derivation of backward stochastic differential equations and loss functions. The formulation accommodates an interpolating parameter connecting stochastic and deterministic sampling schemes, and we identify this parameter as a counterpart of Planck's constant in quantum physics. This analogy enables us to apply the Wentzel–Kramers–Brillouin (WKB) expansion, a well-established technique in quantum physics, for evaluating the negative log-likelihood to assess the performance disparity between stochastic and deterministic sampling schemes.

1. Introduction

Diffusion models have demonstrated impressive performance on image generation tasks (Dhariwal & Nichol, 2021) and they have earned widespread adoption across various applications (Yang et al., 2023). While the predominant contemporary application of diffusion models lies in conditional sampling driven by natural language (Rombach et al., 2022), the mathematical framework underlying their training, i.e., the score-based scheme (Hyvärinen et al., 2009; Vincent, 2011; Song & Ermon, 2019) or the denoising scheme (Sohl-Dickstein et al., 2015; Ho et al., 2020), is not inherently dependent on prior conditions. The work (Kingma et al., 2021) showed the equivalence of these two schemes from a variational perspective. The work (Song et al., 2021b) developed a unified description based on stochastic differential equations (SDEs)[1]. In both cases, the sampling process is given by a Markovian stochastic process.

Another type of probabilistic models employs deterministic sampling schemes using ordinary differential equations (ODEs) such as the probability flow ODE (Song et al., 2021b), which can be understood as a continuous normalizing flow (CNF) (Chen et al., 2018; Lipman et al., 2023). A notable advantage of deterministic sampling schemes is that there is a bijective map between the latent space and the data space. This bijection not only facilitates intricate tasks like manipulation of latent representations for image editing (Su et al., 2023) but also enables the direct computation of negative log-likelihoods (NLLs). Within stochastic sampling processes in contrast to deterministic ones, a direct evaluation method of the NLL remains elusive, though a theoretical bound exists for the NLL (Kong et al., 2023).

In general, stochastic sampling schemes require more number of function evaluations (NFE) than deterministic schemes, and is inferior in terms of sample generation speed. However, the beneficial impact of stochastic generation on certain metrics, such as the Fréchet Inception Distance (FID) (Heusel et al., 2017) is a well-known property in practice. For example, (Karras et al., 2022) reports improvements in these metrics with the incorporation of stochastic processes both in Variance-Exploding and Variance-Preserving pretrained models in (Song et al., 2021b). Intuitively, one can conceptualize noise within stochastic generation as perturbation to propel particles out of local minima, potentially enhancing the diversity and quality of the generated samples. However, beyond this intuitive level, thoroughly quantitative analysis or rigorous theoretical framework to explain this phenomenon is missing.

[1]Department of Physics, Kyoto University, Kyoto 606-8502, Japan [2]RIKEN AIP, RIKEN, Nihonbashi 103-0027, Japan [3]RIKEN iTHEMS, RIKEN, Wako 351-0198, Japan [4]Department of Mathematics, Keio University, Hiyoshi 223-8522, Japan [5]Department of Physics, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-0033, Japan. Correspondence to: Yuji Hirono <yuji.hirono@gmail.com>, Akinori Tanaka <akinori.tanaka@riken.jp>, Kenji Fukushima <fuku@nt.phys.s.u-tokyo.ac.jp>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

[1] See (Oksendal, 2013) for basics of SDE and (Yang et al., 2023) for diffusion models if readers are unfamiliar with them.

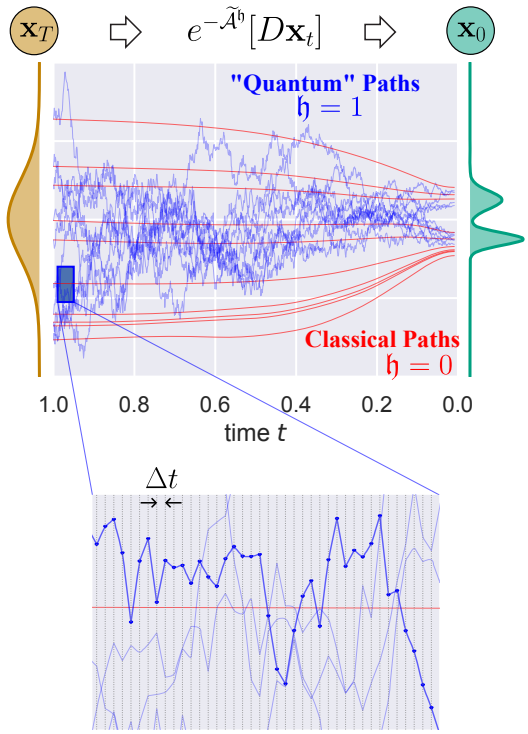

Figure 1. Schematic of the correspondence between the diffusion model and the path integral formulation. The initial 10 points are randomly sampled from $\pi(\mathbf{x}_T)$ and $\mathbf{x}_t$s evolve with Eq. (9) in which the score function is trained with a 2-Gaussian distribution. **Blue zigzag lines:** "Quantum" trajectories with $\mathfrak{h} = 1$. **Red smooth lines:** "Classical" trajectories with $\mathfrak{h} = 0$.

We tackle these two issues by making use of the *path integral formalism*[2], a framework originally developed in quantum physics by Feynman (Feynman, 1948; Feynman & Hibbs, 1965). In classical physics, a particle's motion draws a trajectory, i.e., a deterministic path in space and time. To accommodate quantum effects, the path integral formalism generalizes the trajectory including quantum fluctuations by comprising all possible paths, $\{\mathbf{x}_t\}_{t\in[0,T]}$, of a particle between two points, $\mathbf{x}_0$ at time $t = 0$ and $\mathbf{x}_T$ at $t = T$; see a magnified panel in Fig. 1 for a counterpart in diffusion models. The quantum expectation value of observable $\mathcal{O}(\mathbf{x}_t)$ is computed as a weighted sum: $\sum_{\text{paths}} \mathcal{O}(\mathbf{x}_t)e^{i\mathcal{A}[\mathbf{x}_t]/\hbar}$, where $\hbar$ is Planck's constant. In addition to quantum fluctuations, the path integral can be extended to incorporate stochastic fluctuations as well (Onsager & Machlup, 1953). We demonstrate that diffusion models can be formulated in terms of path integrals, which not only deepens our understanding about diffusion model formulations but also allows for the application of various techniques advanced in quantum physics. Importantly, this framework provides an innovative method for calculating the NLL in stochastic generation processes of diffusion models.

[2] For a brief introduction to the concept of the path integral and a physical meaning of quantum fluctuations, see Appendix A.

Our contributions are as follows.

- We reformulate diffusion models using path integral techniques. We exemplify applications including a re-derivation of the time-reversed SDE (Anderson, 1982) and an estimate of loss functions (Song et al., 2021a) based on Girsanov's theorem (Oksendal, 2013).
- Following (Zhang & Chen, 2023), we introduce an interpolating parameter $\mathfrak{h}$ connecting the stochastic generation ($\mathfrak{h} = 1$) and the probability flow ODE ($\mathfrak{h} = 0$). In the path integral language, the limit $\mathfrak{h} \to 0$ corresponds to the *classical limit* under which quantum fluctuations are dropped off. The path integral formulation of diffusion models reveals the role of $\mathfrak{h}$ as Planck's constant $\hbar$ in quantum physics.
- We apply the Wentzel–Kramers–Brillouin (WKB) expansion (Messiah, 1970), that is formulated in terms of Planck's constant $\hbar$ in quantum physics, with respect to $\mathfrak{h}$ to the likelihood calculation. Based on the first order NLL expression, we quantify the merit of noise in the sampling process by computing the NLL as well as the 2-Wasserstein distance.

Building upon the analogy with quantum physics, these contributions unveil a far more profound connection to physics beyond a classic viewpoint of the Brownian motion.

2. Related Works

Diffusion Models Key contributions in this field of diffusion models include (Sohl-Dickstein et al., 2015) and (Song et al., 2021b) which have laid the foundational principles for these models. Our reformulation of diffusion models by path integrals captures the basic mathematical characteristics of score-based diffusion models. The idea of implementing stochastic variables to represent quantum fluctuations is traced back to (Nelson, 1966), and mathematical foundation has been established as stochastic quantization (Damgaard & Hüffel, 1987). Recent works (Wang et al., 2023; Premkumar, 2023) suggested similarity between diffusion models and quantum physics. However, the path integral derivation of the basic aspects of diffusion models has not been discussed. There is no preceding work to explore the WKB expansion applied in diffusion models.

Likelihood Calculation in Diffusion Models In deterministic sampling schemes based on ODEs, one can directly calculate log-likelihood based on the change-of-variables formula (Chen et al., 2018; Song et al., 2021b; Lipman et al., 2023). Once we turn on noise in the sampling process,

only a formal expression is available from the celebrated Feynman-Kac formula (Huang et al., 2021b). To our best knowledge, there is no stochastic case where the values of log-likelihood have ever been calculated. Our approach of the perturbative expansion with respect to $\mathfrak{h}$ can access the log-likelihood explicitly even in the presence of noise. Note that an exact formula for the log-likelihood has been derived based on the information theory connecting to the Minimum Mean Square Error (MMSE) regression (Kong et al., 2023). However, this method still necessitates the computation of expectation values and cannot be implemented using a single trajectory.

Stochastic and Deterministic Sampling Procedures The role of stochasticity in the sampling process has been investigated in (Karras et al., 2022) through empirical studies. The gap between the score-matching objective depending on the sampling schemes has been pointed out in (Huang et al., 2021a). To bridge the gap between the model distribution generated by the probability flow and the actual data distribution, (Lu et al., 2022) introduced a higher-order score-matching objective. Besides these efforts, (Lai et al., 2023) derived an equation to be satisfied by a score function and introduced a regulator in the loss function to enforce this relation. The present approach is complementary to these works; we employ the perturbative $\mathfrak{h}$-expansion and directly evaluate the noise-strength dependence of the NLL for pretrained models.

3. Reformulation of Diffusion Models by Path Integral Formalism

In this section, we describe the reformulation of score-based diffusion models in terms of path integrals.

3.1. Forward and Reverse Processes

For a given datapoint $\mathbf{x}_0$ sampled from an underlying data distribution p_0, the first step of a score-based diffusion model is to gradually modify the data by adding noise via a forward SDE,

$$\mathrm{d}\mathbf{x}_t = \boldsymbol{f}(\mathbf{x}_t, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}_t. \tag{1}$$

One can view this diffusion process as a collection of stochastic trajectories, and consider the "path-probability" $P(\{\mathbf{x}_t\}_{t\in[0,T]})$ associated with Eq. (1). Intuitively, it corresponds to the joint probability for the "path" $\{\mathbf{x}_t\}_{t\in[0,T]}$ (see also Fig. 1 for schematic illustration of paths). In the path integral formulation of quantum mechanics, the expectation value of observables is expressed as a summation over all possible paths weighted by an exponential factor with a quantity called an *action*. In the following proposition, $[D\mathbf{x}_t]$ is defined later in Eq. (3). We observe that the process (1) of diffusion models can be represented as a path integral:

Proposition 3.1. *The path-probability $P(\{\mathbf{x}_t\}_{t\in[0,T]})$ can be represented in the following path integral form:*

$$P(\{\mathbf{x}_t\}_{t\in[0,T]}) = p_0(\mathbf{x}_0)e^{-\mathcal{A}}[D\mathbf{x}_t], \tag{2}$$

with $\mathcal{A} := \int_0^T L(\dot{\mathbf{x}}_t, \mathbf{x}_t)\mathrm{d}t + J$, where $L(\dot{\mathbf{x}}_t, \mathbf{x}_t)$ is called Onsager–Machlup function (Onsager & Machlup, 1953) defined by

$$L(\dot{\mathbf{x}}_t, \mathbf{x}_t) := \frac{\|\dot{\mathbf{x}}_t - \boldsymbol{f}(\mathbf{x}_t, t)\|^2}{2g(t)^2},$$

and J is the Jacobian associated with the chosen descretization scheme in stochastic process.

Here, the overdot indicates the time derivative. In the physics literature, $L(\dot{\mathbf{x}}_t, \mathbf{x}_t)$ is called the *Lagrangian* and $\mathcal{A}$ is the *action*. For a detailed derivation of Eq. (2) and the explicit expression for J, see Appendix B.2.

Using the path probability (2), the expectation value of any observable $\mathcal{O}(\mathbf{x}_t)$ depending on $\mathbf{x}_t$ obeying Eq. (1) is represented as $\mathbb{E}[O(\mathbf{x}_t)] = \int O(\mathbf{x}_t)P(\{\mathbf{x}_t\}_{t\in[0,T]})$. This expression is commonly referred to as a *path integral* as it involves the summation over infinitely many paths.

Here, we present an intuitive explanation of the expression (2). Let us start from a discretized version of the SDE (1) by Euler-Maruyama scheme:

$$\mathbf{x}_{t+\Delta t} = \mathbf{x}_t + \boldsymbol{f}(\mathbf{x}_t, t)\Delta t + g(t)\sqrt{\Delta t}\mathbf{v}_t, \quad \mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, I),$$
$$\Leftrightarrow p(\mathbf{x}_{t+\Delta t}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t+\Delta t}|\mathbf{x}_t + \boldsymbol{f}(\mathbf{x}_t, t)\Delta t, g(t)^2\Delta t I),$$

where Δt is time interval and I is the identity matrix. Now the time evolution of the SDE is described by a conditional Gaussian distribution. In the following, we take $\mathbf{x}_{t+\Delta t} - \mathbf{x}_t \approx \dot{\mathbf{x}}_t\Delta t$, and deform the conditional Gaussian as

$$p(\mathbf{x}_{t+\Delta t}|\mathbf{x}_t) \propto e^{-\frac{\|\dot{\mathbf{x}}_t - \boldsymbol{f}(\mathbf{x}_t,t)\|^2}{2g(t)^2}\Delta t} = e^{-L(\dot{\mathbf{x}}_t, \mathbf{x}_t)\Delta t}.$$

Now let us consider the probability for realizing an explicit "path" $\{\mathbf{x}_t\} := [\mathbf{x}_0, \mathbf{x}_{\Delta t}, \mathbf{x}_{2\Delta t}, \cdots, \mathbf{x}_T]$, and regarding the summation as the Riemannian sum, we get the following path integral expression for the path probability:

$$\begin{aligned} &p_0(\mathbf{x}_0)p(\mathbf{x}_{\Delta t}|\mathbf{x}_0)\cdots p(\mathbf{x}_T|\mathbf{x}_{T-\Delta t})\prod_t \mathrm{d}\mathbf{x}_t \\ &=: p_0(\mathbf{x}_0)e^{-\sum_{n=0}^{T/\Delta t-1} L(\dot{\mathbf{x}}_{n\Delta t}, \mathbf{x}_{n\Delta t})\Delta t}[D\mathbf{x}_t]_{\Delta t}, \end{aligned} \tag{3}$$

where $[D\mathbf{x}_t]_{\Delta t}$ contains a normalization constant depending on the discretization step Δt. We take the $\Delta t \to 0$ limit at the end of the calculation and omit Δt in the subscript in later discussions. In the expression (3), we need to include an additional contribution denoted by J in Proposition 3.1, depending on the choice of the discretization scheme (see Appendix B.2 for details).

The sampling process of score-based diffusion models is realized by the time-reversed version of the forward SDE (1). The path integral reformulation is beneficial in furnishing an alternative derivation of the reverse-time SDE:

Proposition 3.2. *Let $P(\{\mathbf{x}_t\}_{t\in[0,T]})$ be the path-probability, and $p_t(\mathbf{x})$ be the distribution at time t determined by the Fokker-Planck equation corresponding to Eq. (1). The path-probability is written using the reverse-time action $\widetilde{\mathcal{A}}$ as*

$$P(\{\mathbf{x}_t\}_{t\in[0,T]}) = e^{-\widetilde{\mathcal{A}}}\, p_T(\mathbf{x}_T)[D\mathbf{x}_t],$$

with $\widetilde{\mathcal{A}} := \int_0^T \widetilde{L}(\dot{\mathbf{x}}_t, \mathbf{x}_t)\mathrm{d}t + \widetilde{J}$, where

$$\widetilde{L}(\dot{\mathbf{x}}_t, \mathbf{x}_t) := \frac{\|\dot{\mathbf{x}}_t - \boldsymbol{f}(\mathbf{x}_t, t) + g(t)^2\nabla \log p_t(\mathbf{x}_t)\|^2}{2g(t)^2},$$

and $\widetilde{J}$ is the Jacobian for reverse process depending on the discretization scheme.

We provide the proof in Appendix B.3. We emphasize that the path integral derivation does not rely on the reverse-time SDE (Anderson, 1982). In fact, $\widetilde{L}$ involves the score-function $\nabla \log p_t(\mathbf{x}_t)$, so that Proposition 3.2 gives us another derivation of the reverse-time SDE,

$$\mathrm{d}\mathbf{x}_t = [\boldsymbol{f}(\mathbf{x}_t, t) - g(t)^2\nabla \log p_t(\mathbf{x}_t)]\mathrm{d}t + g(t)\mathrm{d}\bar{\mathbf{w}}_t, \quad (4)$$

by inverting the discussion from Eq. (1) to Proposition 3.1.

3.2. Models and Objectives

In score-based models (Song & Ermon, 2019; Song et al., 2021a;b), the score function $\nabla \log p_t(\mathbf{x}_t)$ is approximated by a neural network $\boldsymbol{s}_\theta(\mathbf{x}_t, t)$, and samplings are performed based on

$$\mathrm{d}\mathbf{x}_t = [\boldsymbol{f}(\mathbf{x}_t, t) - g(t)^2\boldsymbol{s}_\theta(\mathbf{x}_t, t)]\mathrm{d}t + g(t)\mathrm{d}\bar{\mathbf{w}}_t, \quad (5)$$

that is a surrogate for the reverse-time (4). By repeating the same argument as Proposition 3.2, we can conclude that the path-probability $Q_\theta(\{\mathbf{x}_t\})$ corresponding to Eq. (5) takes the following path integral representation:

$$Q_\theta(\{\mathbf{x}_t\}) = e^{-\widetilde{\mathcal{A}}_\theta}\pi(\mathbf{x}_T)[D\mathbf{x}_t], \quad (6)$$

where $\pi(\cdot)$ is a prior distribution, typically chosen to be the standard normal distribution. Here, $\widetilde{\mathcal{A}}_\theta := \int_0^T \widetilde{L}_\theta(\dot{\mathbf{x}}_t, \mathbf{x}_t)\mathrm{d}t + \widetilde{J}_\theta$ with $\widetilde{L}_\theta$ the Onsager–Machlup function for the SDE (5) defined by

$$\widetilde{L}_\theta(\dot{\mathbf{x}}_t, \mathbf{x}_t) := \frac{\|\dot{\mathbf{x}}_t - \boldsymbol{f}(\mathbf{x}_t, t) + g(t)^2\boldsymbol{s}_\theta(\mathbf{x}_t, t)\|^2}{2g(t)^2}, \quad (7)$$

and $\widetilde{J}_\theta$ is the Jacobian contribution. Figure 1 depicts a schematic picture of these reverse-time processes.

Now, we can follow the training scheme based on bound of the log-likelihood or equivalently, KL divergence via data-processing inequality (Song et al., 2021a):

$$D_{\mathrm{KL}}(p_0\|q_{\theta 0}) \le D_{\mathrm{KL}}(P\|Q_\theta). \quad (8)$$

The r.h.s. of Eq. (8) can be calculated by using *Girsanov's theorem* (Oksendal, 2013). An equivalent computation can also be performed in the path integral formulation:

Proposition 3.3. *The KL divergence of path-probabilities can be represented by the path integral form,*

$$D_{\mathrm{KL}}(P\|Q_\theta) = \int e^{-\widetilde{\mathcal{A}}} p_T(\mathbf{x}_T)\log\frac{e^{-\widetilde{\mathcal{A}}}\, p_T(\mathbf{x}_T)}{e^{-\widetilde{\mathcal{A}}_\theta}\pi(\mathbf{x}_T)}[D\mathbf{x}_t],$$

and it can be computed as

$$D_{\mathrm{KL}}(P\|Q_\theta) = D_{\mathrm{KL}}(p_T\|\pi) + \int_0^T \frac{g(t)^2}{2}\mathbb{E}_{p_t}\|\nabla \log p_t(\mathbf{x}_t) - \boldsymbol{s}_\theta(\mathbf{x}_t, t)\|^2\mathrm{d}t.$$

Indeed, Proposition 3.3 yields the same contribution as the calculation based on Girsanov's theorem. We give the derivation of Proposition 3.3 in Appendix B.4.

The discussion so far can be straightforwardly extended to the cases with fixed initial conditions. One can basically make the replacement, $p_t(\mathbf{x}_t) \to p_t(\mathbf{x}_t|\mathbf{x}_0)$; see Appendix B.5 for details.

4. Log-Likelihood by WKB Expansion

We have so far discussed reformulation of score-based diffusion models using the path integral formalism. This reformulation allows for the techniques developed in quantum physics for analyzing the properties of diffusion models. As an illustrative example, we present the calculation of log-likelihood in the presence of noise in the sampling process by pretrained models.

4.1. Interpolating SDE and Probability Flow ODE

Following (Zhang & Chen, 2023), we consider a family of generating processes defined by the following SDE parametrized by $\mathfrak{h} \in \mathbb{R}_{\ge 0}$:

$$\mathrm{d}\mathbf{x}_t = \left[\boldsymbol{f}(\mathbf{x}_t, t) - \frac{1+\mathfrak{h}}{2}g(t)^2\boldsymbol{s}_\theta(\mathbf{x}_t, t)\right]\mathrm{d}t + \sqrt{\mathfrak{h}}g(t)\mathrm{d}\bar{\mathbf{w}}_t. \quad (9)$$

If we take $\mathfrak{h} = 0$, the noise term vanishes, and the process reduces to the *probability flow ODE* (Song et al., 2021b). The situation at $\mathfrak{h} = 1$ corresponds to the original SDE (5). The path-probability corresponding to this process (9) can be expressed as

$$Q_\theta^{\mathfrak{h}}(\{\mathbf{x}_t\}_{t\in[0,T]}) = e^{-\widetilde{\mathcal{A}}_\theta^{\mathfrak{h}}}\pi(\mathbf{x}_T)[D\mathbf{x}_t], \quad (10)$$

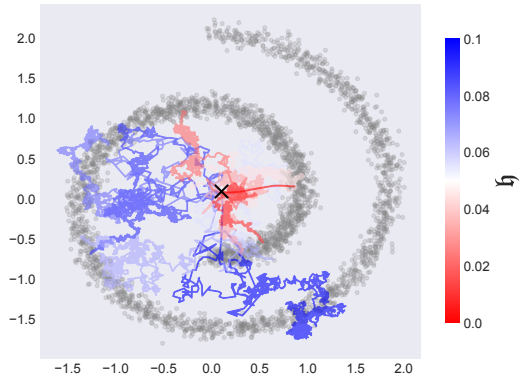

Figure 2. **Gray dots:** training data sampled from the Swiss-roll distribution. **Colored lines:** generative trajectories based on Eq. (9) from the identical initial vector $\mathbf{x}_T$ shown by $\times$ near the center of the figure. The color represents the noise level $\hbar$; the deepest red corresponds to the ODE path in the $\hbar \to 0$ limit and the trajectories become more stochastic with stronger blue.

with $\widetilde{\mathcal{A}}_\theta^\hbar := \int_0^T \widetilde{L}_\theta^\hbar(\dot{\mathbf{x}}_t, \mathbf{x}_t)\mathrm{d}t + \widetilde{J}_\theta^\hbar$, where

$$\widetilde{L}_\theta^\hbar(\dot{\mathbf{x}}_t, \mathbf{x}_t) := \frac{\|\dot{\mathbf{x}}_t - \boldsymbol{f}(\mathbf{x}_t, t) + \frac{1+\hbar}{2} g(t)^2 \boldsymbol{s}_\theta(\mathbf{x}_t, t)\|^2}{2\hbar g(t)^2}, \tag{11}$$

and $\widetilde{J}_\theta^\hbar$ is the Jacobian contribution.

The way how $\hbar$ enters Eq. (11) is quite suggestive: the action $\widetilde{\mathcal{A}}_\theta^\hbar$ is inversely proportional to $\hbar$, similarly to quantum mechanics where the path integral weight takes a form of $e^{-i\mathcal{A}/\hbar}$. This structural similarity provides us with a physical interpretation of the limit, $\hbar \to 0$; it realizes the *classical limit* in the path integral representation. Moreover, this analogy between $\hbar$ and $\hbar$ naturally leads us to explore a perturbative expansion in terms of $\hbar$, i.e., the *WKB expansion*, a well-established technique to treat the asymptotic series expansion in mathematical physics.

In the classical limit of $\hbar \to 0$, the dominant contribution comes from the path that minimizes the action, realizing the *principle of least action* in the physics context. Since the Lagrangian $\widetilde{L}_\theta^\hbar$ of this model is nonnegative, $e^{-\widetilde{\mathcal{A}}_\theta^\hbar}$ goes vanishing in the limit $\hbar \to 0$ unless the path satisfies $\widetilde{L}_\theta^{\hbar=0} = 0$ or equivalently $\dot{\mathbf{x}} = \boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t, t)$, where $\boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t, t)$ is the drift term for the probability flow ODE defined by

$$\boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t, t) := \boldsymbol{f}(\mathbf{x}_t, t) - \frac{1}{2} g(t)^2 \boldsymbol{s}_\theta(\mathbf{x}_t, t). \tag{12}$$

Consequently, the path-probability reduces to the product of delta functions as $\hbar \to 0$:

$$e^{-\widetilde{\mathcal{A}}_\theta^\hbar} \pi(\mathbf{x}_T)[D\mathbf{x}_t] \to \left(\prod_{t=0}^T \delta\big(\dot{\mathbf{x}}_t - \boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t, t)\big) \right) e^{-\widetilde{J}_\theta^{\hbar=0}} \pi(\mathbf{x}_T)[D\mathbf{x}_t].$$

This limiting behavior is visualized in Fig. 2: we plot trajectories generated by the SDE (9) from identical initial $\mathbf{x}_T$ with various $\hbar$s with a pretrained model by the Swiss-roll distribution. The trajectories are concentrated near the ODE path when the noise level is low ($\hbar \approx 0$), which means that the path-probability for $\hbar \to 0$ reduces the classical path represented by Dirac's delta function. In this way, the realization of the probability flow ODE in $\hbar \to 0$ can be regarded as a reminiscent of the reduction from quantum mechanics to classical mechanics in $\hbar \to 0$. The 1-dim. trajectories for $\hbar = 0$ (classical paths) and $\hbar = 1$ ("quantum" paths) are also visualized in Fig. 1.

For $\hbar = 0$, it is well-known that the log-likelihood can be written exactly. Employing the Itô scheme, we can recover the log-likelihood by the path integral with fixed initial condition with $\mathbf{x}_0$ as

$$\begin{aligned} &\log q_\theta^{\hbar=0}(\mathbf{x}_0) \\ &= \log \int_{\mathbf{x}_0} \prod_{t=0}^T \delta\big(\dot{\mathbf{x}}_t - \boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t, t)\big) e^{-\tilde{J}_\theta^{\hbar=0}} \pi(\mathbf{x}_T)[D\mathbf{x}_t] \\ &= \log \pi(\mathbf{x}_T) + \int_0^T \nabla \cdot \boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t, t)\mathrm{d}t, \end{aligned} \tag{13}$$

where $\mathbf{x}_t$ in the last line is obtained from the solution of $\dot{\mathbf{x}}_t = \boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t, t)$ with initial condition $\mathbf{x}_{t=0} = \mathbf{x}_0$, and we have used the explicit expression of $\tilde{J}_\theta^{\hbar=0}$ (see Appendix B). Equation (13) is nothing but the instantaneous change-of-variables formula of an ODE flow (Chen et al., 2018). The computations presented here can be equivalently performed in different discretization schemes, and it should be noted that the final results are free from such scheme dependences.

4.2. $\hbar \neq 0$ in Path Integral

When the score estimation is imperfect, the probability distribution $q_0^\hbar(\mathbf{x}_0)$ of a model acquires nontrivial dependency on parameter $\hbar$. This implies that the quality of sampled images varies depending on the level of noisiness parametrized by $\hbar$.

As we discussed earlier, in quantum physics, $\hbar \to 0$ corresponds to the classical limit, and the effect of small but nonzero $\hbar$ can be taken into account as a series expansion with respect to $\hbar$, commonly referred to as the WKB expansion or the semi-classical approximation. In the path integral formulation of diffusion models, $\hbar$ plays a role of Planck's constant $\hbar$, which quantifies the degrees of "quantumness." As a basis for the WKB expansion of the log-likelihood, we have found the following result:

Theorem 4.1. *The log-likelihood for the process* (9) *satisfies*

$$\begin{aligned} &\log q_0^\hbar(\mathbf{x}_0) = \log \pi(\mathbf{x}_T) \\ &+ \int_0^T \mathrm{d}t\, \nabla \cdot \Big(\boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t, t) - \frac{\hbar g(t)^2}{2} [\boldsymbol{s}_\theta(\mathbf{x}_t, t) - \nabla \log q_t^\hbar(\mathbf{x}_t)] \Big), \end{aligned} \tag{14}$$

where $\pi(\cdot)$ is a prior and $\mathbf{x}_t$ is the solution for the modified probability flow ODE,

$$\dot{\mathbf{x}}_t = \boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t, t) - \frac{\hbar g(t)^2}{2}[\boldsymbol{s}_\theta(\mathbf{x}_t, t) - \nabla \log q_t^\hbar(\mathbf{x}_t)], \quad (15)$$

with initial condition $\mathbf{x}_{t=0} = \mathbf{x}_0$.

We provide the proof of this theorem in Appendix B.6

Note that Eq. (14) is valid for any value of $\hbar \in \mathbb{R}_{\geq 0}$. The modified probability flow ODE (15) indicates that, when the score deviates from the ground-truth value, the deterministic trajectory should also be modified by $\hbar \neq 0$. In Eq. (14), the density $q_t^\hbar(\mathbf{x})$ appears on both sides and this is a self-consistent equation[3]. This relation provides us with the basis for perturbative evaluation of the NLL in power series of $\hbar$.

Theorem 4.2. *The perturbative expansion of the log-likelihood (9) to the first order in $\hbar$ reads*

$$\log q_0^\hbar(\mathbf{x}_0) = \log q_0^{\hbar=0}(\mathbf{x}_0) + \hbar\left(\delta\mathbf{x}_T \cdot \nabla \log \pi(\mathbf{x}_T) + \int_0^T \mathrm{d}t\, \nabla \cdot \delta \boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t, \delta\mathbf{x}_t, t)\right), \quad (16)$$

where $(\mathbf{x}_t, \delta\mathbf{x}_t)$ is the solution for the coupled probability flow ODE,

$$\begin{aligned} \dot{\mathbf{x}}_t &= \boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t, t), \\ \dot{\delta\mathbf{x}}_t &= \delta\boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t, \delta\mathbf{x}_t, t), \end{aligned} \quad (17)$$

with initial condition $\mathbf{x}_{t=0} = \mathbf{x}_0$ and $\delta\mathbf{x}_{t=0} = \mathbf{0}$, where

$$\delta\boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t, \delta\mathbf{x}_t, t) := (\delta\mathbf{x}_t \cdot \nabla)\boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t, t) - \frac{g(t)^2}{2}[\boldsymbol{s}_\theta(\mathbf{x}_t, t) - \nabla \log q_t^{\hbar=0}(\mathbf{x}_t)].$$

We provide the proof in Appendix B.7. Note that we treat $\mathbf{x}_t$ and $\delta\mathbf{x}_t$ as *independent* variables, so the gradient ∇ does not act on $\delta\mathbf{x}_t$. Because Eq. (16) is no longer a self-consistent equation, this theorem allows us to compute $O(\hbar^1)$ correction of the log-likelihood based on $\log q_t^{\hbar=0}(\mathbf{x}_t)$, i.e., log-likelihood defined by the usual probability flow ODE.

A unique feature of the $\hbar$-expansion in this model is that $\hbar$ appears in the denominator of Eq. (11) as well as in the numerator, which contrasts quantum physics where $\hbar$ only appears in the denominator as an overall factor. As a result, once we consider finite $\hbar$ corrections, the "classical" path deviates from the classical path obtained in the $\hbar \to 0$ limit. This deviation caused by $\hbar \neq 0$ is quantified by $\delta\mathbf{x}_t$, which represents how noise influences the bijective relationship between the data points and their latent counterparts.

[3] Note that $q_t^\hbar(\mathbf{x}_t)$ is obtained by replacing variable t in $\int$ to τ and all instances of 0 with t in Theorem 4.1.

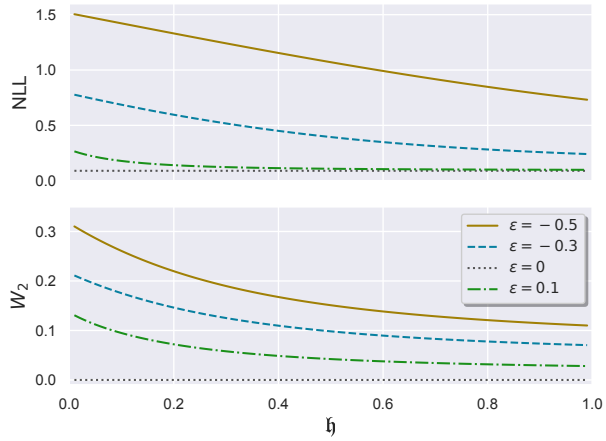

Figure 3. **(Top)** Negative log-likelihood (NLL) of the 1-dim. Gaussian toy model. **(Bottom)** 2-Wasserstein metric (W_2) between the data distribution and the distribution obtained by the same model. Both panels are plotted as a function of parameter $\hbar$.

5. Experiments

5.1. Analysis of 1-dim. Gaussian Data

Let us first illustrate the results in a simple example of one-dimensional (1-dim.) Gaussian distribution, in which everything is analytically tractable. We take the data distribution to be 1-dim. Gaussian distribution with zero mean and variance v_0. For the forward process, we employ $f = -\beta x/2$ and $g(t) = \sqrt{\beta}$, which corresponds to *denoising diffusion probabilistic model* (DDPM) for general β (Ho et al., 2020). Here, we specifically take β to be constant for simplicity. To study the relation between the imperfect score estimation and stochasticity in the sampling process, we parametrize the score model as $s(x,t) = -x(1+\epsilon)/v_t$, where $\epsilon \in \mathbb{R}$ quantifies the deviation from the perfect score, and v_t is the variance of the distribution in the forward process. In this simple model, the distribution remains Gaussian in both forward and backward processes. When $\epsilon = 0$, the backward time evolution exactly matches the forward one, and $q_t(x) = \mathcal{N}(x|0, v_t)$. When $\epsilon \neq 0$, the model distribution $q_t^\hbar(x)$ nontrivially depends on $\hbar$. We evaluate $q_t^\hbar(x)$ and examine the effect of noisiness in the sampling process. In this model, we can compute $q_t^\hbar(x)$ analytically and verify that Eq. (14) is satisfied, as we detail in Appendix D.

In Fig. 3 (Top), we plot the NLL, $-\mathbb{E}_{p_0}[\log q_0^\hbar(x_0)]$, as a function of parameter $\hbar$. Different lines correspond to different values of ϵ. The NLL is computed analytically, and we give the details in Appendix D. For nonzero ϵ, the score estimation is imperfect and the model distribution $q_0^\hbar(x_0)$ differs from the data distribution. In these situations at $\epsilon \neq 0$, the NLL acquires nontrivial dependence on $\hbar$, meaning that the quality of generated images depends the noise level in the sampling process. In Fig. 3 (Top), the

NLLs are decreasing functions of $\mathfrak{h}$, and the presence of noise improves the output quality. Depending on the choice of ϵ, the NLL could be an increasing function of $\mathfrak{h}$ as well. For reference, in Fig. 3 (Bottom), we plot the 2-Wasserstein distance, W_2, between the data distribution and $q_0^{\mathfrak{h}}(x_0)$ for different values of ϵ. The qualitative behavior is consistent with that of the NLL.

5.2. Experiment of 2-dim. Synthetic Data

Let us show some examples[4] of log-likelihood computation for noisy generating process with pre-trained diffusion models trained by two-dimensional (2-dim.) synthetic distributions, Swiss-roll and 25-Gaussian (see Appendix E.1).

Pretrained Models We train simple neural network: $(\mathbf{x}, t) \to \texttt{concat}([\mathbf{x}, t]) \to \big(\texttt{Dense}(128) \to \texttt{swish}\big)^3 \to \texttt{Dense}(2) \to \boldsymbol{s}_\theta(\mathbf{x}, t)$ by choosing one of the following SDE schedulings:

$$\text{SIMPLE:}\ \boldsymbol{f}(\mathbf{x}, t) = -\tfrac{\beta}{2} t\mathbf{x}, \quad g(t) = \sqrt{\beta t}, \quad (\beta = 20)$$

$$\text{COSINE:}\ \boldsymbol{f}(\mathbf{x}, t) = -\tfrac{\pi}{2}\tan(\tfrac{\pi}{2}t)\mathbf{x}, \quad g(t) = \sqrt{\pi\tan(\tfrac{\pi}{2}t)},$$

with time interval $t \in [T_{\text{min}}, T_{\text{max}}]$, where $T_{\text{min}} \in \mathbb{R}$ is small but nonzero to avoid singular behavior around $t = 0$. We take $T_{\text{min}} = 0.01$, $T_{\text{max}} = 1$ here. We train each model by using the following loss function:

$$\frac{1}{N_{\text{batch}}} \sum_{i=1}^{N_{\text{batch}}} \frac{g(t_i)^2}{2} \left\| \frac{\mathbf{x}_{i,t} - \alpha(t_i)\mathbf{x}_i}{\sigma(t_i)^2} + \boldsymbol{s}_\theta(\mathbf{x}_{i,t}, t_i) \right\|^2, \tag{18}$$

where the signal and noise functions $\alpha(t)$ and $\sigma(t)$ are fixed by the chosen SDE scheduling, and t_i represents discretized time. To compute the above, we take the Monte-Carlo sampling with N_{batch} batches of data according to $\mathbf{x}_{i,t} \sim \mathcal{N}(\alpha(t_i)\mathbf{x}_i, \sigma(t_i)^2 I)$. For more details about pre-training with explicit forms of $\alpha(t)$, $\sigma(t)$, t_i, and N_{batch}, see explanations around Eq. (114) in Appendix E.2.

NLL Calculation Our basic strategy is based on Theorem 4.2, the perturbative expansion of the log-likelihood with respect to the parameter $\mathfrak{h}$. To calculate the $O(\mathfrak{h})$ correction, we need the values of $\log q_t^{\mathfrak{h}=0}(\mathbf{x})$ and its derivatives. We can obtain $\log q_t^{\mathfrak{h}=0}(\mathbf{x})$ by solving the probability flow ODE; however, there is no closed formula for its derivatives. In this experiment, the dimension of $\mathbf{x} = [x, y]^\top$ is 2, and this relatively low dimensionality allows us to approximate derivatives by discretized differential operators with Δx, a change in the value of x and y, that needs to be prefixed in practice:

$$\hat{\nabla} \log q_t^{\mathfrak{h}=0}(\mathbf{x}) = \begin{bmatrix} \frac{\log q_t^{\mathfrak{h}=0}(\mathbf{x}_t + [\Delta x, 0]^\top) - \log q_t^{\mathfrak{h}=0}(\mathbf{x}_t - [\Delta x, 0]^\top)}{2\Delta x} \\ \frac{\log q_t^{\mathfrak{h}=0}(\mathbf{x}_t + [0, \Delta x]^\top) - \log q_t^{\mathfrak{h}=0}(\mathbf{x}_t - [0, \Delta x]^\top)}{2\Delta x} \end{bmatrix}, \tag{19}$$

$$\begin{aligned} \hat{\nabla}^2 \log q_t^{\mathfrak{h}=0}(\mathbf{x}) &= \Big(\log q_t^{\mathfrak{h}=0}(\mathbf{x}_t + [\Delta x, 0]^\top) + \log q_t^{\mathfrak{h}=0}(\mathbf{x}_t - [\Delta x, 0]^\top) \\ &\quad + \log q_t^{\mathfrak{h}=0}(\mathbf{x}_t + [0, \Delta x]^\top) + \log q_t^{\mathfrak{h}=0}(\mathbf{x}_t - [0, \Delta x]^\top) \\ &\quad - 4 \log q_t^{\mathfrak{h}=0}(\mathbf{x}_t) \Big) / \Delta x^2. \end{aligned} \tag{20}$$

There are inherent discretization errors that we need to carefully evaluate. More on this will be addressed later in our analysis.

Consequently, to obtain the $O(\mathfrak{h}^1)$ correction, we need to evaluate a nested integral. The pseudocode detailing our calculation method is presented in Algorithm 1. We use `scipy.integrate.solve_ivp` (Virtanen et al., 2020) both in `0th-logqSolver` and discrete-time update in `1th-logqSolver`. We take $T_{\text{min}} > 0$ as the initial time and T_{max} as the terminal time for probability flow ODE. This amounts to calculating the NLL for generation by SDE (9) in time interval $[T_{\text{min}}, T_{\text{max}}]$, and does not affect the sampling quality if we take sufficiently small T_{min}. We show our results in Table 1.

Numerical Errors In Algorithm 1, we made two approximations in discrete differential operators and ODE solvers. To ensure reliability of our results, it is essential to assess and estimate the associated errors. Let us call the numerical value of $\log q_t^{\mathfrak{h}=0}(\mathbf{x})$ calculated by `0th-logqSolver` in Algorithm 1 as $N[\log q_t^{\mathfrak{h}=0}(\mathbf{x})]$, then we have two errors:

$$\begin{aligned} \text{err}\big(\hat{\nabla} N[\log q_t^{\mathfrak{h}=0}(\mathbf{x})]\big) &= \nabla \log q_t^{\mathfrak{h}=0}(\mathbf{x}) - \hat{\nabla} N[\log q_t^{\mathfrak{h}=0}(\mathbf{x})], \\ \text{err}\big(\hat{\nabla}^2 N[\log q_t^{\mathfrak{h}=0}(\mathbf{x})]\big) &= \nabla^2 \log q_t^{\mathfrak{h}=0}(\mathbf{x}) - \hat{\nabla}^2 N[\log q_t^{\mathfrak{h}=0}(\mathbf{x})]. \end{aligned} \tag{21}$$

Exact calculation of these values is unachievable, but nevertheless, we should somehow estimate them. We propose two estimation schemes, `subtraction` (Appendix F.2) and `model` (Appendix F.3). Here, we show results based on the latter method, which operates at a higher speed.

In addition, we should integrate these local errors[5] to estimate the error piled up in the final results. This can also be calculated numerically by the ODE solver, and the numerical errors estimated in this way are shown in Table 1. See Appendix F for more details.

[4] The code to reproduce experiments here can be found at https://github.com/AkinoriTanaka-phys/diffusion_path_integral.

[5] Related to this, we apply certain tips to ensure stable numerical calculations in practice, such as excluding unstable solutions that occur with negligible frequency. See README file in the repository mentioned at Footnote 4 for more details.

Algorithm 1 `1st-logqSolver`

Input: data $\mathbf{x}$, a change in the value of x and y for calculating differentials Δx, SDE info $[\boldsymbol{f}(\cdot,t), g(t), T_{\text{min}}, T_{\text{max}}]$, solver `0th-logqSolver`
Initialize: $t = T_{\text{min}}$, $\mathbf{x}_t = \mathbf{x}$, $\delta\mathbf{x}_t = \mathbf{0}$, $\delta \log q_t = 0$
Calculate $\log q_0 \leftarrow$ `0th-logqSolver`$(\mathbf{x}, t = T_{\text{min}})$
repeat
Calculate

$$\log q_t \leftarrow \texttt{0th-logqSolver}(\mathbf{x}_t, t)$$
$$\log q_t^{x+} \leftarrow \texttt{0th-logqSolver}(\mathbf{x}_t + [\Delta x, 0]^\top, t)$$
$$\log q_t^{x-} \leftarrow \texttt{0th-logqSolver}(\mathbf{x}_t - [\Delta x, 0]^\top, t)$$
$$\log q_t^{y+} \leftarrow \texttt{0th-logqSolver}(\mathbf{x}_t + [0, \Delta x]^\top, t)$$
$$\log q_t^{y-} \leftarrow \texttt{0th-logqSolver}(\mathbf{x}_t - [0, \Delta x]^\top, t)$$

Calculate $\boldsymbol{f}_\theta^{\text{PF}}(\mathbf{x}_t, t)$
Calculate $\delta\boldsymbol{f}_\theta^{\text{PF}}(\mathbf{x}_t, \delta\mathbf{x}_t, t)$ by $\log q_t^{x/y\pm}$ and (19)
Calculate $\nabla \cdot \delta\boldsymbol{f}_\theta^{\text{PF}}(\mathbf{x}_t, \delta\mathbf{x}_t, t)$ by $\log q_t^{x/y\pm}$ and (20)
Update $t, \mathbf{x}_t, \delta\mathbf{x}_t, \delta \log q_t$ by time-discretized version (e.g. Runge-Kutta update) based on

$$\dot{\mathbf{x}}_t = \boldsymbol{f}_\theta^{\text{PF}}(\mathbf{x}_t, t)$$
$$\dot{\delta\mathbf{x}}_t = \delta\boldsymbol{f}_\theta^{\text{PF}}(\mathbf{x}_t, \delta\mathbf{x}_t, t)$$
$$\dot{\delta \log q_t} = \nabla \cdot \delta\boldsymbol{f}_\theta^{\text{PF}}(\mathbf{x}_t, \delta\mathbf{x}_t, t)$$

until $t = T_{\text{max}}$
1st-correction $= \delta\mathbf{x}_{T_{\text{max}}} \cdot \nabla \log \pi(\mathbf{x}_{T_{\text{max}}}) + \delta \log q_{T_{\text{max}}}$
Output: $\log q_0$, 1st-correction

Comparison to 2-Wasserstein We also show 2-Wasserstein distance, W_2, between validation data and generated data in Fig. 4 with stochastic SDE (9). We see that the W_2 values typically decrease especially in the small $\hbar$ region, which signifies the improvement of generated data quality. This observation is consistent with negative-valued NLL corrections in 1ST-CORR column in Table 1. The overall trend is the same as the results from the analytical study in Sec. 5.1; the tendency of enhancing sampled data quality by noise has been confirmed by our experiment even in intricate cases where exact calculation is not accessible. In addition, considering that the values in 1ST-CORR column are the first-order derivative of NLL (cross-entropy) with respect to $\hbar$, one might say that the values for COSINE-SDE are a few times larger than the values for SIMPLE-SDE, which agrees with the behavior of the first-order derivative near $\hbar = 0$ in Fig. 4.

6. Conclusions

We presented a novel formulation of diffusion models utilizing the path integral framework, originally developed in quantum physics. This formulation provides a unified perspective on various aspects of score-based generative models, and we gave the re-derivation of reverse-time SDEs and loss functions for training. In particular, one can introduce a continuous parameter linking different sampling schemes: the probability flow ODEs and stochastic generation. We have performed the expansion with respect to this parameter and perturbatively evaluated the negative log-likelihood, which is a reminiscent of the WKB expansion in quantum physics. In this way, this formulation has presented a new method for scrutinizing the role of noise in the sampling process.

An interesting future direction is an extension of the analysis based on path integral formalism for diffusion Schrödinger bridges (De Bortoli et al., 2021), in which the prior can be more general. Another direction is to understand the cases mentioned in (Karras et al., 2022), where injecting noise would rather degrade the quality of the generated data. This is an open problem, and there is room for deeper study in terms of log-likelihood calculations based on WKB expansions.

Table 1. NLL (cross-entropy) and its $O(\hbar^1)$ corrections. We apply Δx = 0.01 to compute (19) and (20). `tol` represents the value of absolute and relative tolerances of updates in Algorithm 1. We set absolute and relative tolerances of `0th-logqSolver` as order `1e-5`.

SWISS-ROLL

SDE (NLL)	`tol`	1ST-CORR	ERRORS
SIMPLE	`1e-3`	-0.31±0.21	0.13±0.00
(1.39± 0.05)	`1e-5`	-0.44±0.38	0.13±0.00
COSINE	`1e-3`	-1.59±0.57	0.35±0.00
(1.42± 0.02)	`1e-5`	-3.27±1.11	0.37±0.02

25-GAUSSIAN

SDE (NLL)	`tol`	1ST-CORR	ERRORS
SIMPLE	`1e-3`	-3.64±0.49	0.31±0.00
(-1.22± 0.01)	`1e-5`	-3.61±0.64	0.32±0.01
COSINE	`1e-3`	-17.57±5.56	0.70±0.01
(-1.71± 0.02)	`1e-5`	-19.65±17.46	0.67±0.03

Limitations Our experiments did not involve actual image data. This omission is primarily due to the current evaluation method's limitations regarding NLLs, which are not directly applicable to high-dimensional data such as images on account of the uses of explicit discrete differentials[6]. An-

[6] However, we believe this limitation can be resolved with the following strategy. The main bottleneck was the computation of $\nabla^2 \log q_t(\mathbf{x})$ using numerical discretization. We can sidestep this by introducing a parametrized function $\mathbf{r}_\theta(\mathbf{x}, t)$ by a neural

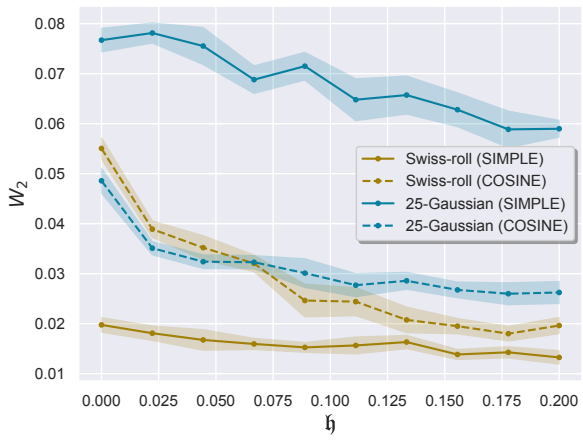

Figure 4. 2-Wasserstein metrics (W_2) by `POT` library (Flamary et al., 2021) between validation data and generated data via (9) with Swiss-roll data and 25-Gaussian data with SIMPLE and COSINE SDE scheduling. The dots and the errorbars represent the mean values over 10 independent trials and $\pm\text{std}/\sqrt{10}$, respectively.

other limitation is the potential of underestimated numerical error in our computed NLLs. Although our estimated local errors look safe (Appendix F.5), they are potentially underestimated because the estimation is based on score-based model that does not match to $\nabla \log q_t(\mathbf{x})$ exactly in general.

Acknowledgements

The work of Y. H. was supported in part by JSPS KAKENHI Grant No. JP22H05111. The work of A. T. was supported in part by JSPS KAKENHI Grant No. JP22H05116. This work of K. F. was supported by JSPS KAKENHI Grant Nos. JP22H01216 and JP22H05118.

Impact Statement

This paper presents a novel approach for understanding diffusion models through the path-integral formalism of quantum physics. By offering a new perspective on the behaviors of score-based diffusion models grounded in physical principles, our work bridges the fields of Machine Learning and Physics. We believe this interdisciplinary approach contributes significantly to advancing both fields and could inspire innovative developments. Although our study primarily addresses theoretical aspects, we recognize the potential societal implications of AI advancements and underscore the necessity of ethical considerations and responsible application of these technologies.

network, which is to be used as a proxy for $\nabla \log q_t(\mathbf{x})$. To optimize this neural network, we use an objective function defined as:

$$\mathcal{L}(\theta) = \int dt \lambda(t) \mathbb{E}_Q[\|\mathbf{r}_\theta(\mathbf{x}, t) - \nabla \log q_t(\mathbf{x})\|^2]. \tag{22}$$

The minimization of $\mathcal{L}(\theta)$ is equivalent to that of the following:

$$\mathcal{L}'(\theta) = \int dt \lambda(t) \mathbb{E}_Q[\|\mathbf{r}_\theta(\mathbf{x}, t)\|^2 + 2\nabla \cdot \mathbf{r}_\theta(\mathbf{x}, t)]. \tag{23}$$

The loss function $\mathcal{L}'(\theta)$ can be evaluated via a sample average over paths generated by ODEs. The divergence of $\mathbf{r}_\theta(\mathbf{x}, t)$ can be efficiently computed by Skilling-Hutchinson trace estimation. We believe this approach enables us to apply our algorithm to higher-dimensional settings such as realistic images.

References

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

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax.

Chen, R. T., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Damgaard, P. H. and Hüffel, H. Stochastic quantization. *Physics Reports*, 152(5):227–398, 1987. ISSN 0370-1573. doi: https://doi.org/10.1016/0370-1573(87)90144-X. URL https://www.sciencedirect.com/science/article/pii/037015738790144X.

De Bortoli, V., Thornton, J., Heng, J., and Doucet, A. Diffusion schrödinger bridge with applications to score-based generative modeling. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 17695–17709. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/940392f5f32a7ade1cc201767cf83e31-Paper.pdf.

DeepMind, Babuschkin, I., Baumli, K., Bell, A., Bhupatiraju, S., Bruce, J., Buchlovsky, P., Budden, D., Cai, T., Clark, A., Danihelka, I., Dedieu, A., Fantacci, C., Godwin, J., Jones, C., Hemsley, R., Hennigan, T., Hessel, M., Hou, S., Kapturowski, S., Keck, T., Kemaev, I., King, M., Kunesch, M., Martens, L., Merzic, H., Mikulik, V., Norman, T., Papamakarios, G., Quan, J., Ring, R., Ruiz, F., Sanchez, A., Sartran, L., Schneider, R., Sezener, E., Spencer, S., Srinivasan, S., Stanojević, M., Stokowiec, W., Wang, L., Zhou, G., and Viola, F. The DeepMind JAX Ecosystem, 2020. URL http://github.com/google-deepmind.

Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Feynman, R. P. Space-time approach to non-relativistic quantum mechanics. *Rev. Mod. Phys.*, 20:367–387, Apr 1948. doi: 10.1103/RevModPhys.20.367. URL https://link.aps.org/doi/10.1103/RevModPhys.20.367.

Feynman, R. P. and Hibbs, A. R. *Quantum mechanics and path integrals*. International series in pure and applied physics. McGraw-Hill, New York, NY, 1965.

Flamary, R., Courty, N., Gramfort, A., Alaya, M. Z., Boisbunon, A., Chambon, S., Chapel, L., Corenflos, A., Fatras, K., Fournier, N., Gautheron, L., Gayraud, N. T., Janati, H., Rakotomamonjy, A., Redko, I., Rolet, A., Schutz, A., Seguy, V., Sutherland, D. J., Tavenard, R., Tong, A., and Vayer, T. Pot: Python optimal transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021. URL http://jmlr.org/papers/v22/20-451.html.

Heek, J., Levskaya, A., Oliver, A., Ritter, M., Rondepierre, B., Steiner, A., and van Zee, M. Flax: A neural network library and ecosystem for JAX, 2023. URL http://github.com/google/flax.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Huang, C.-W., Lim, J. H., and Courville, A. C. A variational perspective on diffusion-based generative models and score matching. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 22863–22876. Curran Associates, Inc., 2021a. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/c11abfd29e4d9b4d4b566b01114d8486-Paper.pdf.

Huang, C.-W., Lim, J. H., and Courville, A. C. A variational perspective on diffusion-based generative models and score matching. *Advances in Neural Information Processing Systems*, 34:22863–22876, 2021b.

Hyvärinen, A., Hurri, J., Hoyer, P. O., Hyvärinen, A., Hurri, J., and Hoyer, P. O. Estimation of non-normalized statistical models. *Natural Image Statistics: A Probabilistic Approach to Early Computational Vision*, pp. 419–426, 2009.

Karras, T., Aittala, M., Aila, T., and Laine, S. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 35:26565–26577, 2022.

Kingma, D., Salimans, T., Poole, B., and Ho, J. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021.

Kingma, D. P. and Gao, R. Understanding the diffusion objective as a weighted integral of elbos. *arXiv preprint arXiv:2303.00848*, 2023.

Kong, X., Brekelmans, R., and Steeg, G. V. Information-theoretic diffusion. *11th International Conference on Learning Representations*, 2023.

Lai, C.-H., Takida, Y., Murata, N., Uesaka, T., Mitsufuji, Y., and Ermon, S. FP-diffusion: Improving score-based diffusion models by enforcing the underlying score fokker-planck equation. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 18365–18398. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/lai23d.html.

Lipman, Y., Chen, R. T., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. *11th International Conference on Learning Representations*, 2023.

Lu, C., Zheng, K., Bao, F., Chen, J., Li, C., and Zhu, J. Maximum likelihood training for score-based diffusion ODEs by high order denoising score matching. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 14429–14460. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/lu22f.html.

Machlup, S. and Onsager, L. Fluctuations and irreversible process. ii. systems with kinetic energy. *Phys. Rev.*, 91:1512–1515, Sep 1953. doi: 10.1103/PhysRev.91.1512. URL https://link.aps.org/doi/10.1103/PhysRev.91.1512.

Messiah, A. *Quantum Mechanics Volume I*. North-Holland Publishing Company, 1970.

Nelson, E. Derivation of the schrödinger equation from newtonian mechanics. *Phys. Rev.*, 150:1079–1085, Oct 1966. doi: 10.1103/PhysRev.150.1079. URL https://link.aps.org/doi/10.1103/PhysRev.150.1079.

Oksendal, B. *Stochastic differential equations: an introduction with applications*. Springer Science & Business Media, 2013.

Onsager, L. and Machlup, S. Fluctuations and irreversible processes. *Phys. Rev.*, 91:1505–1512, Sep 1953. doi: 10.1103/PhysRev.91.1505. URL https://link.aps.org/doi/10.1103/PhysRev.91.1505.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Petzka, H., Fischer, A., and Lukovnicov, D. On the regularization of wasserstein gans. *6th International Conference on Learning Representations*, 2018.

Premkumar, A. Generative diffusion from an action principle. *arXiv preprint arXiv:2310.04490*, 2023.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. PMLR, 2015.

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

Song, Y., Durkan, C., Murray, I., and Ermon, S. Maximum likelihood training of score-based diffusion models. *Advances in Neural Information Processing Systems*, 34: 1415–1428, 2021a.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. *9th International Conference on Learning Representations*, 2021b.

Su, X., Song, J., Meng, C., and Ermon, S. Dual diffusion implicit bridges for image-to-image translation. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=5HLoTvVGDe.

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

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

Wang, L., Aarts, G., and Zhou, K. Diffusion models as stochastic quantization in lattice field theory. *arXiv preprint arXiv:2309.17082*, 2023.

Yang, L., Zhang, Z., Song, Y., Hong, S., Xu, R., Zhao, Y., Zhang, W., Cui, B., and Yang, M.-H. Diffusion models: A comprehensive survey of methods and applications. *ACM Computing Surveys*, 56(4):1–39, 2023.

Zhang, Q. and Chen, Y. Fast sampling of diffusion models with exponential integrator. *11th International Conference on Learning Representations*, 2023.

A. Overview of quantum mechanics and the WKB approximation

The quantum physical system is defined by the Hamiltonian H which is for example $H(\mathbf{x}, \mathbf{p}) = \mathbf{p}^2/2 + V(\mathbf{x})$ for a one-particle potential problem (with the mass normalized to the unity). Then, the wave function ψ should satisfy the Schrödinger equation,

$$H\Big(\mathbf{x}, \frac{\hbar}{i}\nabla\Big)\psi = -\frac{\hbar^2}{2}\nabla^2\psi + V(\mathbf{x})\psi = i\hbar\frac{\partial}{\partial t}\psi\,. \tag{24}$$

We can introduce a specific form without loss of generality, $\psi(\mathbf{x}, t) = e^{i\mathcal{A}(\mathbf{x},t)/\hbar}$, up to irrelevant normalization, and plug this into Eq. (24). Then, the limit of $\hbar \to 0$ gives

$$\frac{1}{2}(\nabla\mathcal{A})^2 + V(\mathbf{x}) = -\frac{\partial\mathcal{A}}{\partial t}\,, \tag{25}$$

which is called the Hamilton-Jacobi equation in classical mechanics derived from the stationary condition of the action, i.e., $\delta\mathcal{A} = 0$.

Importantly, the wave functions at different times, t and t_0, are connected by a kernel convolution as

$$\psi(\mathbf{x}, t) = \int d\mathbf{x}_0\psi(\mathbf{x}_0, t_0)K(\mathbf{x}, t|\mathbf{x}_0, t_0)\,, \tag{26}$$

where $K(\mathbf{x}, t|\mathbf{x}_0, t_0)$ is given by $\sum_{\text{paths}} e^{i\mathcal{A}[\mathbf{x}_t]/\hbar}$ with paths with boundaries at $(\mathbf{x}, t)$ and $(\mathbf{x}_0, t_0)$. This representation provides us with a useful intuition that $\psi(\mathbf{x}, t)$ is furiously fluctuating and canceling on average over trajectories unless trajectories in $K(\mathbf{x}, t|\mathbf{x}_0, t_0)$ are close enough within $\delta\mathcal{A} = O(\hbar)$. Therefore, $\hbar$ is a parameter to characterize the magnitude of quantum fluctuations. In our discussions in the text, what is called "Euclidean" formalism is adopted in which the imaginary time, it, is regarded as the time, and then the weight no longer fluctuates and $e^{i\mathcal{A}/\hbar}$ is replaced by $e^{-\mathcal{A}/\hbar}$. Then, the kernel is also given an interpretation as the conditional probability which is denoted by $p(\mathbf{x}_t|\mathbf{x}_0)$ in the text.

This strategy to expand the solution of Eq. (24) in powers of $\hbar$ can be systematically performed, which is generally referred to as the WKB expansion. As we discuss in the main part, in the diffusion models, a counterpart of $\hbar$ denoted by $\mathfrak{h}$ appears in a slightly more complicated way than quantum mechanics, but the expansion works similarly.

B. From SDE to path integral

In this appendix, we describe the details of the path integral formulation of diffusion models.

B.1. Discretization schemes

In later discussions, we will introduce discretized summations in different ways, which do not seem apparent in their continuum counterpart. We here introduce discretization schemes that will be used in later analyses.

Suppose we have $\{\mathbf{x}_t\}_{t\in[0,T]}$ obeying SDE (1). We discretize the time interval $[0, T]$ with width Δt and $M \coloneqq T/\Delta t$, which we take to be an integer. Let us take the following summation:

$$\sum_{n=0}^{M-1} \boldsymbol{f}_{t_n} \cdot (\mathbf{x}_{t_n+\Delta t} - \mathbf{x}_{t_n}), \tag{27}$$

where $t_n \coloneqq n\Delta t$ and $\boldsymbol{f}_t = \boldsymbol{f}(\mathbf{x}_t, t)$. We will consider the continuum limit $\Delta t \to 0$ and will refer to this as the Itô scheme, which will be denoted by

$$\int \boldsymbol{f} \cdot \mathrm{d}\mathbf{x} \coloneqq \int \boldsymbol{f}_t \cdot (\mathbf{x}_{t+\mathrm{d}t} - \mathbf{x}_t). \tag{28}$$

A common prescription in the physics literature is the Stratonovich scheme,

$$\int \boldsymbol{f} \circ \mathrm{d}\mathbf{x} \coloneqq \int \frac{\boldsymbol{f}_t + \boldsymbol{f}_{t+\mathrm{d}t}}{2} \cdot (\mathbf{x}_{t+\mathrm{d}t} - \mathbf{x}_t), \tag{29}$$

which should be understood as a short-hand notation for the following equation,

$$\lim_{\Delta t\to 0} \sum_{n=0}^{M-1} \frac{\boldsymbol{f}_{t_n} + \boldsymbol{f}_{t_n+\Delta t}}{2} \cdot (\mathbf{x}_{t_n+\Delta t} - \mathbf{x}_{t_n}). \tag{30}$$

We also encounter the reverse-Itô scheme, which is written as

$$\int \boldsymbol{f} \tilde{\cdot} \, \mathrm{d}\mathbf{x} \coloneqq \int \boldsymbol{f}(\mathbf{x}_{t+\mathrm{d}t}) \cdot (\mathbf{x}_{t+\mathrm{d}t} - \mathbf{x}_t). \tag{31}$$

This can be seen as the Itô scheme in reverse time.

B.2. Derivation of the path integral expression

Here, we give a derivation of the path integral representation of a diffusion model (Proposition 3.1). We consider a forward process described by the SDE (1). In the physics literature, Eq. (1) is commonly expressed in the following form,

$$\dot{\mathbf{x}}_t = \boldsymbol{f}(\mathbf{x}_t, t) + \boldsymbol{\xi}_t, \tag{32}$$

where the noise term satisfies

$$\mathbb{E}[\xi_t^i \xi_{t'}^j] = g(t)^2 \delta^{ij} \delta(t - t'). \tag{33}$$

The expectation value of a generic observable $\mathcal{O}(\{\mathbf{x}_t\})$ over noise realizations can be expressed as

$$\mathbb{E}_{P[\{\mathbf{x}_t\}]}[\mathcal{O}(\{\mathbf{x}_t\})] = \int [D\mathbf{x}_t][D\boldsymbol{\xi}_t] \, \mathcal{O}(\{\mathbf{x}_t\}) \left(\prod_t \delta(\mathbf{x}_t - \mathbf{x}_t^{\mathrm{sol}}) \right) e^{-\int_0^T \frac{1}{2g(t)^2} \|\boldsymbol{\xi}_t\|^2 \mathrm{d}t} p_0(\mathbf{x}_0), \tag{34}$$

where $p_0(\mathbf{x}_0)$ is the probability distribution of initial states (i.e., the data distribution), $\delta(\cdot)$ is Dirac's delta function, and $\mathbf{x}_t^{\mathrm{sol}}$ is the solution of Eq. (32). The symbol $[D\mathbf{x}_t]_{\Delta t}$ denotes $[D\mathbf{x}_t]_{\Delta t} \coloneqq \prod_{n=0}^{M} C(\Delta t) \mathrm{d}\mathbf{x}_{t_n}$, where $C(\Delta t)$ is a normalization constant, and similarly for $[D\boldsymbol{\xi}_t]$. When the weight is sufficiently well-behaved (like $e^{-\int_0^T \frac{1}{2g(t)^2} \|\boldsymbol{\xi}_t\|^2 \mathrm{d}t}$ in Eq. (34)), the limit $\Delta t \to 0$ the path integral is well-defined. We will be considering this limit at the end of the calculation do not explicitly indicate the dependence on Δt hereafter. The delta function imposes that $\mathbf{x}_t$ is a solution of Eq. (32) under a given noise realization. Using the change-of-variable formula for the delta function,

$$\delta(\mathbf{x}_t - \mathbf{x}_t^{\mathrm{sol}}) = \left| \det \frac{\delta \mathrm{EOM}_t}{\delta \mathbf{x}_{t'}} \right| \delta(\mathrm{EOM}_t), \tag{35}$$

where $\mathrm{EOM}_t \coloneqq \dot{\mathbf{x}}_t - \boldsymbol{f}(\mathbf{x}_t, t) - \boldsymbol{\xi}_t$. The Jacobian part gives a nontrivial contribution (see Appendix C for a detailed derivation) in the case of the Stratonovich scheme,

$$\begin{aligned} \left| \det \frac{\delta \mathrm{EOM}_t}{\delta \mathbf{x}_{t'}} \right| &= \det(\partial_t \delta_{ij} - \partial_i f_j) \\ &= \det \partial_t \, \det(\delta_{ij} - \partial_t^{-1} \partial_i f_j) \\ &\propto e^{-\frac{1}{2} \int \nabla \cdot \boldsymbol{f}(\mathbf{x}_t, t) \mathrm{d}t}, \end{aligned} \tag{36}$$

while it gives a trivial factor for the Itô scheme. Writing this factor as e^{-J}, the expectation value can now be written as

$$\mathbb{E}_{P[\{\mathbf{x}_t\}]}[\mathcal{O}(\{\mathbf{x}_t\})] = \int [D\mathbf{x}_t][D\boldsymbol{\xi}_t] \, \mathcal{O}(\{\mathbf{x}_t\}) e^{-J} \left(\prod_t \delta(\dot{\mathbf{x}}_t - \boldsymbol{f}_t - \boldsymbol{\xi}_t) \right) e^{-\int_0^T \frac{1}{2g(t)^2} \|\boldsymbol{\xi}_t\|^2 \mathrm{d}t} p_0(\mathbf{x}_0). \tag{37}$$

Performing the integration over $\boldsymbol{\xi}_t$, we arrive at the expression

$$\mathbb{E}_{P[\{\mathbf{x}_t\}]}[\mathcal{O}(\{\mathbf{x}_t\})] = \int [D\mathbf{x}_t] \, \mathcal{O}(\{\mathbf{x}_t\}) \, e^{-\mathcal{A}} p_0(\mathbf{x}_0), \tag{38}$$

Here, we defined the action $\mathcal{A}$ by

$$\mathcal{A} \coloneqq \int_0^T L(\dot{\mathbf{x}}_t, \mathbf{x}_t) \mathrm{d}t + J \tag{39}$$

where $L(\dot{\mathbf{x}}_t, \mathbf{x}_t)$ is the Onsager-Machlup function (Onsager & Machlup, 1953; Machlup & Onsager, 1953) given by

$$L(\dot{\mathbf{x}}_t, \mathbf{x}_t) = \frac{1}{2g(t)^2} \|\dot{\mathbf{x}}_t - \boldsymbol{f}(\mathbf{x}_t, t)\|^2, \tag{40}$$

and J is the term coming from the Jacobian, which depends on the choice of the discretization scheme:

$$J = \begin{cases} 0 & \text{Itô} \\ \int_0^T \frac{1}{2}\nabla \cdot \boldsymbol{f}_t(\mathbf{x}_t, t)\mathrm{d}t & \text{Stratonovich} \\ \int_0^T \nabla \cdot \boldsymbol{f}_t(\mathbf{x}_t, t)\mathrm{d}t & \text{Reverse-Itô} \end{cases}. \tag{41}$$

This concludes the derivation of Proposition 3.1. Any observable can be computed using the weight given by $\mathcal{A}$. For example, one can compute the joint probability for the variables on discrete time points $\{t_n\}_{n=1,\ldots,N}$ as

$$P(\mathbf{y}_{t_1}, \mathbf{y}_{t_2}, \ldots, \mathbf{y}_{t_N}) = \int [D\mathbf{x}_t]\, e^{-\mathcal{A}}\, p_0(\mathbf{x}_0) \prod_{n=1}^{N} \delta(\mathbf{y}_{t_n} - \mathbf{x}_{t_n}). \tag{42}$$

Before ending this section, let us comment on the mathematical interpretation of the Onsager-Machlup function (40). A reader might be concerned with the meaning of the time derivative $\dot{\mathbf{x}}_t$, since the paths in question are almost surely non-differentiable. Indeed, we should note that the notation of the time derivative $\dot{\mathbf{x}}_t$ appearing in the action is only symbolic. Precisely speaking, the action,

$$\mathcal{A} = \int_0^T dt \frac{1}{2g(t)^2} \|\dot{\mathbf{x}}_t - \mathbf{f}(\mathbf{x}_t, t)\|^2, \tag{43}$$

should be understood through its discretized form,

$$\mathcal{A}_{\text{disc}} = \sum_i \Delta t \frac{1}{2g(t_i)^2} \|(\mathbf{x}_{t_i+\Delta t} - \mathbf{x}_{t_i})/\Delta t - \mathbf{f}(\mathbf{x}_{t_i}, t_i)\|^2. \tag{44}$$

The path-integral measure is obtained in the limit $\Delta t \to 0$, i.e.,

$$\lim_{\Delta t \to 0} e^{-\mathcal{A}_{\text{disc}}}. \tag{45}$$

This limit is well-defined. In this form, it is apparent that the paths do not need to be differentiable.

B.3. Time-reversed SDE

Here we present the derivation of the time-reversed dynamics in the path integral formalism (Proposition 3.2). We will use the Itô scheme, and later comment on other discretization schemes. We start by rewriting Eq. (38) as

$$\begin{aligned} \mathbb{E}_{P[\{\mathbf{x}_t\}]}[\mathcal{O}(\{\mathbf{x}_t\})] &= \int [D\mathbf{x}_t]\, \mathcal{O}(\{\mathbf{x}_t\})\, p_T(\mathbf{x}_T)\, e^{\int_0^T [-L - \ln p_T(\mathbf{x}_T) + \ln p_0(\mathbf{x}_0)]\mathrm{d}t} \\ &= \int [D\mathbf{x}_t]\, p_T(\mathbf{x}_T)\, e^{-\widetilde{\mathcal{A}}}, \end{aligned} \tag{46}$$

where $\widetilde{\mathcal{A}} \coloneqq \int_0^T \widetilde{L}(\dot{\mathbf{x}}_t, \mathbf{x}_t)\mathrm{d}t$ with

$$\widetilde{L}(\dot{\mathbf{x}}_t, \mathbf{x}_t) \coloneqq L(\dot{\mathbf{x}}_t, \mathbf{x}_t) + \frac{\mathrm{d}}{\mathrm{d}t} \ln p_t(\mathbf{x}_t). \tag{47}$$

The total time derivative of $\ln p_t(\mathbf{x}_t)$ is written as

$$\mathrm{d} \ln p_t(\mathbf{x}_t) = \partial_t \ln p_t(\mathbf{x}_t)\mathrm{d}t + \mathrm{d}\mathbf{x}_t \cdot \nabla \ln p_t(\mathbf{x}_t) + \frac{1}{2} g(t)^2 \nabla^2 \ln p_t(\mathbf{x}_t)\mathrm{d}t, \tag{48}$$

where we have used Itô's formula. For $\partial_t p_t(\mathbf{x}_t)$, we use the Fokker-Planck equation,

$$\partial_t p_t(\mathbf{x}) = -\nabla \cdot \left(\boldsymbol{f}(\mathbf{x}_t, t) p_t(\mathbf{x}) - \frac{g(t)^2}{2} \nabla p_t(\mathbf{x}) \right), \tag{49}$$

which we rewrite as

$$\partial_t \ln p_t(\mathbf{x}) = -\nabla \cdot \boldsymbol{f}(\mathbf{x}_t, t) - \boldsymbol{f}(\mathbf{x}_t, t) \cdot \nabla \ln p_t(\mathbf{x}) + \frac{g(t)^2}{2} \left[\nabla^2 \ln p_t(\mathbf{x}) + (\nabla \ln p_t(\mathbf{x}))^2 \right]. \tag{50}$$

The new Lagrangian $\widetilde{L}(\dot{\mathbf{x}}_t, \mathbf{x}_t)$ is now written as

$$\begin{aligned}\widetilde{L}(\dot{\mathbf{x}}_t, \mathbf{x}_t) &= \frac{1}{2g_t^2}(\dot{\mathbf{x}}^2 - 2\dot{\mathbf{x}}\cdot(\boldsymbol{f}(\mathbf{x}_t,t) - g(t)^2\nabla \ln p_t(\mathbf{x}_t)) + \boldsymbol{f}(\mathbf{x}_t,t)^2) + \frac{g(t)^2}{2}\nabla^2 \ln p_t(\mathbf{x}_t) \\ &\quad - \nabla\cdot\boldsymbol{f}(\mathbf{x}_t,t) - \boldsymbol{f}(\mathbf{x}_t,t)\cdot\nabla\ln p_t(\mathbf{x}_t) + \frac{g(t)^2}{2}\left[\nabla^2 \ln p_t(\mathbf{x}_t) + (\nabla \ln p_t(\mathbf{x}_t))^2\right] \\ &= \frac{1}{2g_t^2}(\dot{\mathbf{x}} - \boldsymbol{f}(\mathbf{x}_t,t) + g(t)^2\nabla\ln p_t(\mathbf{x}_t))^2 - \nabla\cdot\left(\boldsymbol{f}(\mathbf{x}_t,t) - g(t)^2\nabla\ln p_t(\mathbf{x}_t)\right).\end{aligned} \tag{51}$$

Thus, the action $\widetilde{\mathcal{A}}$ is given by

$$\int_0^T \widetilde{L}(\dot{\mathbf{x}}_t, \mathbf{x}_t)\mathrm{d}t = \int_0^T \left[\frac{1}{2g(t)^2}\|\dot{\mathbf{x}}_t - \widetilde{\boldsymbol{f}}(\mathbf{x}_t,t)\|^2 - \nabla\cdot\widetilde{\boldsymbol{f}}(\mathbf{x}_t,t)\right]\mathrm{d}t, \tag{52}$$

where

$$\widetilde{\boldsymbol{f}}(\mathbf{x}_t,t) \coloneqq \boldsymbol{f}(\mathbf{x}_t,t) - g(t)^2\nabla\ln p_t(\mathbf{x}_t). \tag{53}$$

Note that there appears a nontrivial Jacobian term in Eq. (52). This term disappears if we rewrite the integral using the inverse time τ in the Itô convention. The action $\widetilde{\mathcal{A}}$ contains the following contribution,

$$\widetilde{\mathcal{A}} \supset -\int \frac{1}{g(t)^2}\widetilde{\boldsymbol{f}}(\mathbf{x}_t,t)\cdot\mathrm{d}\mathbf{x}_t. \tag{54}$$

Currently, this product written in the Itô scheme. We rewrite this term in the reverse-Itô scheme,

$$\begin{aligned}\widetilde{\boldsymbol{f}}_t\cdot\mathrm{d}\mathbf{x}_t &= \left[\widetilde{\boldsymbol{f}}_{t+\mathrm{d}t} - \left(\widetilde{\boldsymbol{f}}_{t+\mathrm{d}t} - \widetilde{\boldsymbol{f}}_t\right)\right]\cdot\mathrm{d}\mathbf{x}_t = \widetilde{\boldsymbol{f}}_{t+\mathrm{d}t}\cdot\mathrm{d}\mathbf{x}_t - \sum_{i,j}\partial_i(\widetilde{f}_t)_j\mathrm{d}x_i\mathrm{d}x_j \\ &= \widetilde{\boldsymbol{f}}_{t+\mathrm{d}t}\cdot\mathrm{d}\mathbf{x}_t - g(t)^2\nabla\cdot\widetilde{\boldsymbol{f}}_t\,\mathrm{d}t.\end{aligned} \tag{55}$$

The second term of this equation cancels the Jacobian term. Thus, the time-reversed action can be naturally interpreted as an Itô integral in reverse time.

The same procedure can be also done for the Stratonovich scheme. The difference is the presence of the Jacobian term in the original action and the total time derivative of $p_t(\mathbf{x}_t)$ is written as

$$\mathrm{d}\ln p_t(\mathbf{x}_t) = \partial_t \ln p_t(\mathbf{x}_t)\mathrm{d}t + \mathrm{d}\mathbf{x}_t\cdot\nabla\ln p_t(\mathbf{x}_t) \tag{56}$$

instead of Eq. (48). Following similar steps, the time-reversed action is found to be given by

$$\widetilde{\mathcal{A}} = \int_0^T \widetilde{L}(\dot{\mathbf{x}}_t, \mathbf{x}_t)\mathrm{d}t = \int_0^T \left[\frac{1}{2g(t)^2}\|\dot{\mathbf{x}}_t - \widetilde{\boldsymbol{f}}_t\|^2 - \frac{1}{2}\nabla\cdot\widetilde{\boldsymbol{f}}_t\right]\mathrm{d}t. \tag{57}$$

Note that the sign of the Jacobian term is flipped compared with the original action, which allows us to interpret the process as a time-reversed one.

B.4. Evaluation of KL divergence

We here give the derivation of Proposition 3.3. We evaluate the upper limit of the KL divergence of the data distribution and a model,

$$D_{\mathrm{KL}}(p_0(\mathbf{x}_0)\|q_0(\mathbf{x}_0)) \le D_{\mathrm{KL}}(P(\{\mathbf{x}_t\}_{t\in[0,T]})\|Q(\{\mathbf{x}_t\}_{t\in[0,T]})), \tag{58}$$

where we used the data processing inequality. Below, we evaluate the RHS of Eq. (58).

The time-reversed action of the data distribution (in the reverse-Itô scheme) reads

$$\widetilde{\mathcal{A}} \coloneqq \int_0^T \widetilde{L}(\dot{\mathbf{x}}_t, \mathbf{x}_t)\mathrm{d}t = \int_0^T \left[\frac{1}{2g(t)^2}\|\dot{\mathbf{x}}_t - \widetilde{\boldsymbol{f}}(\mathbf{x}_t,t)\|^2\right]\mathrm{d}t, \tag{59}$$

The time-reversed action of a model is given by

$$\widetilde{\mathcal{A}}_\theta \coloneqq \int_0^T \widetilde{L}_\theta(\dot{\mathbf{x}}_t, \mathbf{x}_t)\mathrm{d}t = \int_0^T \left[\frac{1}{2g(t)^2}\|\dot{\mathbf{x}}_t - \widetilde{\boldsymbol{f}}_\theta(\mathbf{x}_t, t)\|^2\right]\mathrm{d}t, \tag{60}$$

with

$$\widetilde{\boldsymbol{f}}_\theta(\mathbf{x}_t, t) \coloneqq \boldsymbol{f}(\mathbf{x}_t, t) - g(t)^2\boldsymbol{s}_\theta(\mathbf{x}_t, t). \tag{61}$$

We note that, we employ the reverse-Itô scheme in the following computation and that is why there is no Jacobian term in Eqs. (59) and (60). We rewrite the joint probability of paths as

$$P(\{\mathbf{x}_t\}_{t\in[0,T]}) = e^{-\mathcal{A}}p_0(\mathbf{x}_0) = p_T(\mathbf{x}_T)e^{-\widetilde{\mathcal{A}}}, \tag{62}$$

$$Q(\{\mathbf{x}_t\}_{t\in[0,T]}) = e^{-\mathcal{A}_\theta}q_0(\mathbf{x}_0) = q_T(\mathbf{x}_T)e^{-\widetilde{\mathcal{A}}_\theta}. \tag{63}$$

The KL divergence of the path-probability $P(\{\mathbf{x}_t\}_t)$ from $Q(\{\mathbf{x}_t\}_t)$ is written as

$$\begin{aligned} D_{\mathrm{KL}}\left(P(\{\mathbf{x}_t\}_{t\in[0,T]})\|Q(\{\mathbf{x}_t\}_{t\in[0,T]})\right) &= \mathbb{E}_{P(\{\mathbf{x}_t\})}\left[\ln\frac{P(\{\mathbf{x}_t\}_{t\in[0,T]})}{Q(\{\mathbf{x}_t\}_{t\in[0,T]})}\right] \\ &= \mathbb{E}_{P(\{\mathbf{x}_t\}_t)}\left[\ln\frac{p_T(\mathbf{x}_T)}{q_T(\mathbf{x}_T)} - \widetilde{\mathcal{A}} + \widetilde{\mathcal{A}}_\theta\right] \\ &= D_{\mathrm{KL}}\left(p_T(\mathbf{x}_T)\|q_T(\mathbf{x}_T)\right) + \mathbb{E}_{P(\{\mathbf{x}_t\}_t)}\left[\widetilde{\mathcal{A}}_\theta - \widetilde{\mathcal{A}}\right]. \end{aligned} \tag{64}$$

The second term of the RHS can be written as

$$\begin{aligned} \mathbb{E}_{P(\{\mathbf{x}_t\}_t)}\left[\widetilde{\mathcal{A}}_\theta - \widetilde{\mathcal{A}}\right] &= \mathbb{E}_{P(\{\mathbf{x}_t\}_t)}\int_0^T \frac{1}{2g(t)^2}\left[-2\dot{\mathbf{x}}_t\tilde{}(\widetilde{\boldsymbol{f}}_\theta(\mathbf{x}_t, t) - \widetilde{\boldsymbol{f}}(\mathbf{x}_t, t)) + (\widetilde{\boldsymbol{f}}_\theta(\mathbf{x}_t, t))^2 - (\widetilde{\boldsymbol{f}}(\mathbf{x}_t, t))^2\right]\mathrm{d}t \\ &= \mathbb{E}_{P(\{\mathbf{x}_t\}_t)}\int_0^T \frac{1}{2g(t)^2}[\widetilde{\boldsymbol{f}}_\theta(\mathbf{x}_{t+\mathrm{d}t}, t+\mathrm{d}t) - \widetilde{\boldsymbol{f}}(\mathbf{x}_{t+\mathrm{d}t}, t+\mathrm{d}t)]\cdot[-2\dot{\mathbf{x}}_t + \widetilde{\boldsymbol{f}}_\theta(\mathbf{x}_t, t) + \widetilde{\boldsymbol{f}}(\mathbf{x}_t, t)]\mathrm{d}t. \end{aligned} \tag{65}$$

We discretize this and look at the contribution from the neighboring part $(t, t+\Delta t)$,

$$\begin{aligned} &\mathbb{E}_{P(\mathbf{x}_t, \mathbf{x}_{t+\Delta t})}\left[\Delta t\,\frac{1}{2g(t)^2}(\widetilde{\boldsymbol{f}}_{\theta,t+\Delta t} - \widetilde{\boldsymbol{f}}_{t+\Delta t})\cdot\left[-2(\mathbf{x}_{t+\Delta t} - \mathbf{x}_t)/\Delta t + \widetilde{\boldsymbol{f}}_{\theta,t} + \widetilde{\boldsymbol{f}}_t\right]\right] \\ &= \mathbb{E}_{p_t(\mathbf{x}_t)}\left[\Delta t\,\frac{1}{2g(t)^2}(\widetilde{\boldsymbol{f}}_{\theta,t+\Delta t} - \widetilde{\boldsymbol{f}}_{t+\Delta t})\cdot(-2\widetilde{\boldsymbol{f}}_t + \widetilde{\boldsymbol{f}}_{\theta,t} + \widetilde{\boldsymbol{f}}_t)\right] \\ &= \mathbb{E}_{p_t(\mathbf{x}_t)}\left[\Delta t\,\frac{1}{2g(t)^2}\|\widetilde{\boldsymbol{f}}_{\theta,t} - \widetilde{\boldsymbol{f}}_t\|^2 + O(\Delta t^2)\right], \end{aligned} \tag{66}$$

where we performed the summation over $\delta = \mathbf{x}_{t+\Delta t} - \mathbf{x}_t$. Summing up these contributions for $[0, T]$, we have

$$\begin{aligned} \mathbb{E}_{P(\{\mathbf{x}_t\}_t)}\left[\widetilde{\mathcal{A}}_\theta - \widetilde{\mathcal{A}}\right] &= \mathbb{E}_{P(\{\mathbf{x}_t\}_t)}\left[\int_0^T \frac{1}{2g(t)^2}\|\widetilde{\boldsymbol{f}}_\theta(\mathbf{x}_t, t) - \widetilde{\boldsymbol{f}}(\mathbf{x}_t, t)\|^2\mathrm{d}t\right] \\ &= \mathbb{E}_{P(\{\mathbf{x}_t\}_t)}\left[\int_0^T \frac{g(t)^2}{2}\|\nabla\ln p_t(\mathbf{x}_t) - \boldsymbol{s}_\theta(\mathbf{x}_t, t)\|^2\mathrm{d}t\right]. \end{aligned} \tag{67}$$

Thus, we have obtained the following inequality,

$$D_{\mathrm{KL}}(p_0(\mathbf{x}_0)\|q_0(\mathbf{x}_0)) \leq D_{\mathrm{KL}}\left(p_T(\mathbf{x}_T)\|q_T(\mathbf{x}_T)\right) + \int_0^T \frac{g(t)^2}{2}\mathbb{E}_{p_t}\left[\|\nabla\ln p_t(\mathbf{x}_t) - \boldsymbol{s}_\theta(\mathbf{x}_t, t)\|^2\right]\mathrm{d}t. \tag{68}$$

The distribution $q_T(\mathbf{x}_T)$ is taken to be a prior, $\pi(\mathbf{x}_T)$. This concludes the proof of Proposition 3.3.

B.5. Conditional variants

A similar argument applies to the case when the initial state is fixed. The derivation of the reverse process can be obtained by replacing the probability densities with conditional ones on a chosen initial state $\mathbf{x}_0'$. The expectation value of a general observable in this situation can be written as

$$\mathbb{E}_{P(\{\mathbf{x}_t\}_{t\in(0,T]}|\mathbf{x}_0')}[\mathcal{O}(\{\mathbf{x}_t\})] = \int [D\mathbf{x}_t]\, \mathcal{O}(\{\mathbf{x}_t\})\, e^{-\int L\mathrm{d}t} \delta(\mathbf{x}_0 - \mathbf{x}_0'). \tag{69}$$

Similarly to the previous section, we can rewrite this as

$$\mathbb{E}_{P(\{\mathbf{x}_t\}_{t\in(0,T]}|\mathbf{x}_0')}[\mathcal{O}(\{\mathbf{x}_t\})] = \int [D\mathbf{x}_t]\, \mathcal{O}(\{x_t\})\, P_T(\mathbf{x}_T|\mathbf{x}_0')\, e^{-\widetilde{\mathcal{A}}(\mathbf{x}_0')} \delta(\mathbf{x}_0 - \mathbf{x}_0'), \tag{70}$$

where $\widetilde{\mathcal{A}}(\mathbf{x}_0') \coloneqq \int_0^T \widetilde{L}(\dot{\mathbf{x}}_t, \mathbf{x}_t|\mathbf{x}_0')\, \mathrm{d}t$ with

$$\widetilde{L}(\dot{\mathbf{x}}_t, \mathbf{x}_t|\mathbf{x}_0') \coloneqq L(\dot{\mathbf{x}}_t, \mathbf{x}_t|\mathbf{x}_0') + \frac{\mathrm{d}}{\mathrm{d}t} p_t(\mathbf{x}_t|\mathbf{x}_0'). \tag{71}$$

Following similar calculations with the unconditional case, the force of the reverse process turns out to be given by

$$\widetilde{\boldsymbol{f}}(\mathbf{x}_t, t) \coloneqq \boldsymbol{f}(\mathbf{x}_t, t) - g(t)^2 \nabla \ln p_t(\mathbf{x}_t|\mathbf{x}_0'). \tag{72}$$

One can also repeat a similar argument for the evaluation of the KL divergence with fixed $\mathbf{x}_0'$, which corresponds to the ELBO-based loss (Kingma et al., 2021; Kingma & Gao, 2023):

Proposition B.1. *Let $p_t(\mathbf{x}|\mathbf{x}_0)$ as the Markov kernel from time 0 to t determined by the Fokker-Planck equation. Determining the Onsager–Machlup function for reverse process $\widetilde{L}(\dot{\mathbf{x}}_t, \mathbf{x}_t|\mathbf{x}_0')$ to satisfy*

$$p_0(\mathbf{x}_0|\mathbf{x}_0') e^{-\int_0^T L(\dot{\mathbf{x}}_t, \mathbf{x}_t)\mathrm{d}t + J}[D\mathbf{x}_t] = e^{-\int_0^T \widetilde{L}(\dot{\mathbf{x}}_t, \mathbf{x}_t|\mathbf{x}_0')\mathrm{d}t + \widetilde{J}(\mathbf{x}_0')} p_T(\mathbf{x}_T|\mathbf{x}_0')[D\mathbf{x}_t], \tag{73}$$

yields

$$\widetilde{L}(\dot{\mathbf{x}}_t, \mathbf{x}_t|\mathbf{x}_0') = \frac{\|\dot{\mathbf{x}}_t - \boldsymbol{f}(\mathbf{x}_t, t) + g(t)^2 \nabla \log p_t(\mathbf{x}_t|\mathbf{x}_0')\|^2}{2g(t)^2}, \tag{74}$$

where $\widetilde{J}(\mathbf{x}_0')$ is the conditional Jacobian for the reverse process depending on discretization scheme.

This representation proves to be practically valuable, especially since in most cases, we do not have access to the score function $\nabla \log p_t(\mathbf{x})$.

Proposition B.2. *The KL divergence of path-probabilities can be represented by the path integral form,*

$$D_{\mathrm{KL}}(P(\cdot|\mathbf{x}_0')\|Q_\theta) = \int e^{-\widetilde{\mathcal{A}}(\mathbf{x}_0')} p_T(\mathbf{x}_T) \log \frac{e^{-\widetilde{\mathcal{A}}(\mathbf{x}_0')}\, p_T(\mathbf{x}_T)}{e^{-\widetilde{\mathcal{A}}_\theta} \pi(\mathbf{x}_T)} [D\mathbf{x}_t], \tag{75}$$

where π is a prior distribution, and it can be computed as

$$D_{\mathrm{KL}}(P(\cdot|\mathbf{x}_0')\|Q_\theta) = D_{\mathrm{KL}}(p_T\|\pi) + \int_0^T \frac{g(t)^2}{2} \mathbb{E}_{p_t}\left[\|\nabla \log p_t(\mathbf{x}_t|\mathbf{x}_0') - \boldsymbol{s}_\theta(\mathbf{x}_t, t)\|^2\right] \mathrm{d}t. \tag{76}$$

B.6. Likelihood evaluation

We here give the proof of Theorem 4.1.

The probability distribution $q_t^\hbar(\mathbf{x})$ corresponding to the backward process (5) satisfies the following Fokker-Planck equation,

$$\partial_t q_t^\hbar(\mathbf{x}) = -\nabla \cdot \left[\boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}, t) q_t^\hbar(\mathbf{x}) - \frac{\hbar g(t)^2}{2} (\boldsymbol{s}_{\theta,t}(\mathbf{x}) - \nabla \log q_t^\hbar(\mathbf{x})) q_t^\hbar(\mathbf{x})\right]. \tag{77}$$

Note that $q_t^{\mathfrak{h}}(\mathbf{x})$ depends on $\mathfrak{h}$ nontrivially when $\boldsymbol{s}_\theta(\mathbf{x},t) \neq \nabla \log q_t^{\mathfrak{h}}(\mathbf{x})$. Introducing a new parameter $\gamma \in \mathbb{R}_{\geq 0}$, Eq. (77) can be written as

$$\begin{aligned}\partial_t q_t^{\mathfrak{h}}(\mathbf{x}) &= -\nabla \cdot \left[\boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x},t) q_t^{\mathfrak{h}}(\mathbf{x}) - \frac{\mathfrak{h} g(t)^2}{2} (\boldsymbol{s}_{\theta,t}(\mathbf{x}) - \nabla \log q_t^{\mathfrak{h}}(\mathbf{x})) q_t^{\mathfrak{h}}(\mathbf{x}) - \frac{\gamma g(t)^2}{2} \nabla q_t^{\mathfrak{h}}(\mathbf{x}) + \frac{\gamma g(t)^2}{2} \nabla q_t^{\mathfrak{h}}(\mathbf{x}) \right] \\ &= -\nabla \cdot \left[\left(\boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x},t) - \frac{g(t)^2}{2} \left[\mathfrak{h}(\boldsymbol{s}_{\theta,t}(\mathbf{x}) - \nabla \log q_t^{\mathfrak{h}}(\mathbf{x})) + \gamma \nabla \log q_t^{\mathfrak{h}}(\mathbf{x}) \right] \right) q_t^{\mathfrak{h}}(\mathbf{x}) + \frac{\gamma g(t)^2}{2} \nabla q_t^{\mathfrak{h}}(\mathbf{x}) \right].\end{aligned} \tag{78}$$

Let us define

$$\boldsymbol{F}_{\mathfrak{h},\gamma}(\mathbf{x},t) \coloneqq \boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x},t) - \frac{\mathfrak{h} g(t)^2}{2} \boldsymbol{s}_\theta(\mathbf{x},t) + (\mathfrak{h} - \gamma) \frac{g(t)^2}{2} \nabla \log q_t^{\mathfrak{h}}(\mathbf{x}). \tag{79}$$

We consider the path integral expression for Eq. (78). The corresponding action is given by

$$\widetilde{\mathcal{A}}_{\mathfrak{h},\gamma} = \int_0^T \mathrm{d}t\, \widetilde{L}_{\mathfrak{h},\gamma}(\dot{\mathbf{x}}_t, \mathbf{x}_t) = \int_0^T \frac{1}{2\gamma g(t)^2} \|\dot{\mathbf{x}} - \boldsymbol{F}_{\mathfrak{h},\gamma}(\mathbf{x}_t,t)\|^2 \,\mathrm{d}t. \tag{80}$$

Using the Itô scheme, one can obtain the likelihood for finite $\mathfrak{h}$ by taking the limit $\gamma \to 0$,

$$\begin{aligned} q_0^{\mathfrak{h}}(\mathbf{x}_0') &\propto \int \left(\prod_{t \in [0,T]} \mathrm{d}\mathbf{x}_t \right) \delta(\mathbf{x}_0 - \mathbf{x}_0') e^{-\widetilde{\mathcal{A}}_{\mathfrak{h},\gamma} + \int_0^T \nabla \cdot \boldsymbol{F}_{\mathfrak{h},\gamma} \mathrm{d}t} q_T^{\mathfrak{h}}(\mathbf{x}_T) \\ &\xrightarrow{\gamma \to 0} e^{\int_0^T \nabla \cdot \boldsymbol{F}_{\mathfrak{h},\gamma=0} \mathrm{d}t} q_T^{\mathfrak{h}}(\mathbf{x}_T). \end{aligned} \tag{81}$$

Thus, we obtain the formula for the likelihood for finite $\mathfrak{h}$,

$$\log q_0^{\mathfrak{h}}(\mathbf{x}_0) = \log q_T^{\mathfrak{h}}(\mathbf{x}_T) + \int_0^T \nabla \cdot \left(\boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t,t) - \frac{\mathfrak{h} g(t)^2}{2} \left[\boldsymbol{s}_\theta(\mathbf{x}_t,t) - \nabla \log q_t^{\mathfrak{h}}(\mathbf{x}_t) \right] \right) \mathrm{d}t. \tag{82}$$

Since $q_T^{\mathfrak{h}}(\mathbf{x}_T)$ is taken to be a prior, $q_T^{\mathfrak{h}}(\mathbf{x}_T) = \pi(\mathbf{x}_T)$, this concludes the proof of Theorem 4.1.

B.7. Likelihood up to first order of $\mathfrak{h}$

We here give a proof of Theorem 4.2.

First, we rename the solution for the modified probability flow ODE (15) as $\mathbf{x}_t^{\mathfrak{h}}$, i.e.

$$\dot{\mathbf{x}}_t^{\mathfrak{h}} = \boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t^{\mathfrak{h}},t) - \frac{\mathfrak{h} g(t)^2}{2} \left[\boldsymbol{s}_\theta(\mathbf{x}_t^{\mathfrak{h}},t) - \nabla \log q_t^{\mathfrak{h}}(\mathbf{x}_t^{\mathfrak{h}}) \right], \quad \mathbf{x}_{t=0}^{\mathfrak{h}} = \mathbf{x}_0 \tag{83}$$

and it can be represented by the formal integral

$$\mathbf{x}_t^{\mathfrak{h}} = \mathbf{x}_0 + \int_0^t \left(\boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t^{\mathfrak{h}},t) - \frac{\mathfrak{h} g(t)^2}{2} \left[\boldsymbol{s}_\theta(\mathbf{x}_t^{\mathfrak{h}},t) - \nabla \log q_t^{\mathfrak{h}}(\mathbf{x}_t^{\mathfrak{h}}) \right] \right) \mathrm{d}t. \tag{84}$$

Next, we consider Taylor expansion of $\boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t^{\mathfrak{h}},t)$ with $\mathfrak{h}$:

$$\boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t^{\mathfrak{h}},t) = \boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t,t) + \mathfrak{h} \underbrace{\partial_{\mathfrak{h}} \boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t,t)|_{\mathfrak{h}=0}}_{(\partial_{\mathfrak{h}} \mathbf{x}_t^{\mathfrak{h}}|_{\mathfrak{h}=0} \cdot \nabla) \boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t,t)} + O(\mathfrak{h}^2) \tag{85}$$

Now we define $\partial_{\mathfrak{h}} \mathbf{x}_t^{\mathfrak{h}}|_{\mathfrak{h}=0}$ as $\delta \mathbf{x}_t$ for simplicity, then by taking differential of (84), we get

$$\begin{aligned} \delta \mathbf{x}_t &= \partial_{\mathfrak{h}} \mathbf{x}_t^{\mathfrak{h}}|_{\mathfrak{h}=0} \\ &= \int_0^t \left(\underbrace{\partial_{\mathfrak{h}} \boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t^{\mathfrak{h}},t)|_{\mathfrak{h}=0}}_{(\partial_{\mathfrak{h}} \mathbf{x}_t^{\mathfrak{h}}|_{\mathfrak{h}=0} \cdot \nabla) \boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t,t)} - \frac{g(t)^2}{2} \left[\boldsymbol{s}_\theta(\mathbf{x}_t,t) - \nabla \log q_t^{\mathfrak{h}=0}(\mathbf{x}_t) \right] \right) \mathrm{d}t \\ &= \int_0^t \left((\delta \mathbf{x}_t \cdot \nabla) \boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t,t) - \frac{g(t)^2}{2} \left[\boldsymbol{s}_\theta(\mathbf{x}_t,t) - \nabla \log q_t^{\mathfrak{h}=0}(\mathbf{x}_t) \right] \right) \mathrm{d}t, \end{aligned} \tag{86}$$

which is the integral of

$$\dot{\delta \mathbf{x}_t} = \underbrace{(\delta \mathbf{x}_t \cdot \nabla) \boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t, t) - \frac{g(t)^2}{2}\left[\boldsymbol{s}_\theta(\mathbf{x}_t, t) - \nabla \log q_t^{\hbar=0}(\mathbf{x}_t)\right]}_{\delta \boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t, \delta \mathbf{x}_t, t)}. \tag{87}$$

By combining it to $O(\hbar^0)$ ODE for $\mathbf{x}_t$, i.e., the probability flow ODE, we get Eq. (17).

Next, we apply the generic formula that is valid with arbitrary function $f(\mathbf{x})$,

$$f(\mathbf{x}_t^\hbar) = f(\mathbf{x}_t) + \hbar \delta \mathbf{x}_t \cdot \nabla f(\mathbf{x}_t) + O(\hbar^2), \tag{88}$$

to the right hand side of the self-consistent equation of log-likelihood in Theorem 4.1.

$$\begin{aligned}
&\log q_0^\hbar(\mathbf{x}_0) \\
&= \log \pi(\mathbf{x}_T^\hbar) + \int_0^T \nabla \cdot \left(\boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t^\hbar, t) - \frac{\hbar g(t)^2}{2}[\boldsymbol{s}_\theta(\mathbf{x}_t^\hbar, t) - \nabla \log q_t^\hbar(\mathbf{x}_t^\hbar)] \right) \mathrm{d}t \\
&= \begin{matrix} \log \pi(\mathbf{x}_T) \\ + \\ \hbar \delta \mathbf{x}_t \cdot \nabla \log \pi(\mathbf{x}_T) \end{matrix} + \int_0^T \left(\begin{matrix} \nabla \cdot \boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t, t) \\ + \\ \hbar (\delta \mathbf{x}_t \cdot \nabla) \nabla \cdot \boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t, t) \end{matrix} - \frac{\hbar g(t)^2}{2} \nabla \cdot [\boldsymbol{s}_\theta(\mathbf{x}_t, t) - \nabla \log q_t^{\hbar=0}(\mathbf{x}_t)]) \right) + O(\hbar^2),
\end{aligned} \tag{89}$$

then, this expression is equivalent to what we wanted to prove, i.e., Eq. (16).

C. Computation of determinant

Let us detail on the computation of the determinant of an operator of the form,

$$\det\left(\partial_t - M\right). \tag{90}$$

We first factorize this as $\det\left(\partial_t - M\right) = \det \partial_t \cdot \det\left(1 - \partial_t^{-1} M\right)$. The latter factor is computed as

$$\det\left(1 - \partial_t^{-1} M\right) = \exp \log \det\left(1 - \partial_t^{-1} M\right) = \exp \mathrm{tr}\, \ln\left(1 - \partial_t^{-1} M\right) = \exp \mathrm{tr} \sum_{n=1}^{\infty} \left(-\frac{1}{n} (\partial_t^{-1} M)^n \right). \tag{91}$$

Noting that $\partial_t^{-1} \delta(t - t') = \theta(t - t')$, the term with $n = 1$ gives

$$\exp\left(-\theta(0) \int \mathrm{d}t\, \mathrm{tr}\, M \right), \tag{92}$$

where $\theta(x)$ is the step function. Its value at 0 depends on the discretization scheme as

$$\theta(0) = \begin{cases} 0 & \text{Ito} \\ \frac{1}{2} & \text{Stratonovich} \\ 1 & \text{Reverse-Ito} \end{cases}. \tag{93}$$

All the higher-order terms vanish. For example, the term with $n = 2$ reads

$$-\int \theta(t - t')\theta(t' - t) \mathrm{tr}\, M^2\, \mathrm{d}t \mathrm{d}t' = 0. \tag{94}$$

Thus, we have

$$\det\left(\partial_t - M\right) \propto \exp\left(-\theta(0) \int \mathrm{d}t\, \mathrm{tr}\, M \right), \tag{95}$$

with $\theta(0)$ given by Eq. (93).

D. Detail of the example in Sec. 5.1

We here describe the details of the simple example discussed in Sec. 5.1.

The data distribution is taken to be a one-dimensional Gaussian distribution,

$$P_0(x_0) = \mathcal{N}(x_0 \,|\, 0, v_0). \tag{96}$$

For the forward process, we use the following SDE,

$$\mathrm{d}x_t = -\frac{1}{2}\beta x_t \,\mathrm{d}t + \sqrt{\beta}\mathrm{d}w_t. \tag{97}$$

Namely, we have chosen the force and noise strength as

$$f(x) = -\frac{\beta}{2}x, \quad g = \sqrt{\beta}. \tag{98}$$

We here take β to be constant. In the current situation, the distribution stays Gaussian with a zero mean throughout the time evolution, and the distribution $p_t(x)$ is fully specified by its variance. Using Ito's formula,

$$\mathrm{d}(x_t^2) = 2x_t \,\mathrm{d}x_t + \frac{1}{2}\cdot 2(\mathrm{d}x_t)^2 = -\beta x_t^2 \,\mathrm{d}t + \beta \mathrm{d}t. \tag{99}$$

Thus, the time evolution of the variance v_t is described by

$$\mathrm{d}v_t = -\beta(v_t - 1)\mathrm{d}t, \tag{100}$$

which can be solved with initial condition $v_{t=0} = v_0$ as

$$v_t = 1 + e^{-\beta t}(v_0 - 1). \tag{101}$$

The distribution at t is given by

$$p_t(x) = \mathcal{N}(x \,|\, 0, v_t). \tag{102}$$

with v_t given by Eq. (101).

D.1. Likelihood evaluation

Suppose that the estimated score $s(x,t)$ is given by

$$s(x,t) = \partial_x \log q_t(x) = -\frac{x}{v_t}(1+\epsilon). \tag{103}$$

where $\epsilon \in \mathbb{R}$ is a constant. The parameters ϵ quantifies the deviation from the ideal estimation and when $\epsilon = 0$ we can recover the original data distribution perfectly. If $\epsilon \neq 0$, the likelihood $q_0(x_0)$ depends nontrivially on parameter $\mathfrak{h}$. The force of the reverse process is written as

$$\widetilde{f}_{\mathfrak{h},\epsilon}(x) = f(x) - \frac{1+\mathfrak{h}}{2}g^2 s(x,t) = -\frac{\beta}{2}x - \frac{1+\mathfrak{h}}{2}\beta\left(-\frac{x}{v_t}(1+\epsilon_t)\right) = \frac{\beta}{2}\left(\frac{(1+\mathfrak{h})(1+\epsilon_t)}{v_t} - 1\right)x. \tag{104}$$

Let us denote the variance of the model distribution $q_t^{\mathfrak{h}}(x)$ by v_t', which differs from v_t when $\epsilon \neq 0$. The variance v_t' obeys

$$\mathrm{d}v_t' = \beta\left(\frac{(1+\mathfrak{h})(1+\epsilon)}{v_t} - 1\right)v_t'\mathrm{d}t - \mathfrak{h}\beta\mathrm{d}t. \tag{105}$$

If we solve this with the boundary condition $v_T' = v_T$, we have

$$v_t' = \frac{1}{(1+\mathfrak{h})(1+\epsilon) - 1}\left[\mathfrak{h}v_t + \epsilon(1+\mathfrak{h})e^{\beta(T-t)}v_T\left(e^{-\beta(T-t)}\frac{v_t}{v_T}\right)^{(1+\mathfrak{h})(1+\epsilon)}\right]. \tag{106}$$

Note that, when $\epsilon = 0$, we have $v'_t = v_t$ and it is independent of $\mathfrak{h}$. For nonzero ϵ, v'_0 depends on $\mathfrak{h}$ nontrivially. The model distribution at t is given by

$$q_t^{\mathfrak{h}}(x) = \mathcal{N}(x \,|\, 0, v'_t). \tag{107}$$

The NLL can be expressed as

$$-\mathbb{E}_{p_0(x_0)}[\log q_0^{\mathfrak{h}}(x_0)] = -\int \mathrm{d}x\, \mathcal{N}(x \,|\, 0, v_0) \log \mathcal{N}(x \,|\, 0, v'_{t=0}) = \frac{1}{2}\left[\log 2\pi v'_{t=0} + \frac{v_0}{v'_{t=0}}\right]. \tag{108}$$

As another measure, we can compute the 2-Wasserstein distance,

$$W_2(p_0, p_T)^2 = (\sqrt{v_0} - \sqrt{v'_{t=0}})^2. \tag{109}$$

Let us check the formula for the likelihood is satisfied. We shall check that the following formula is satisfied:

$$\log q_0^{\mathfrak{h}}(x_0) - \log q_T^{\mathfrak{h}}(x_T(x_0)) = \int_0^T \partial_x \left[\widetilde{f}^{\mathrm{PF}}(x_t) - \frac{\mathfrak{h}}{2} g^2 (s(x_t, t) - \nabla \ln q_t^{\mathfrak{h}}(x_t))\right] \mathrm{d}t, \tag{110}$$

where $\widetilde{f}^{\mathrm{PF}}(x) = \widetilde{f}_{\mathfrak{h}=0,\epsilon}(x)$. Noting that

$$x_t = \sqrt{\frac{v'_t}{v_0}} x_0, \tag{111}$$

and the LHS can be computed to give

$$\mathrm{LHS} = \frac{1}{2} \log \frac{v_T}{v'_0}. \tag{112}$$

The integrand on the RHS reads

$$\begin{aligned} \partial_x \left[\widetilde{f}^{\mathrm{PF}}(x) - \frac{\mathfrak{h}}{2} g^2 (s(x,t) - \nabla \ln q_t^{\mathfrak{h}}(x))\right] &= \partial_x \left[-\frac{\beta}{2} x - \frac{1+\mathfrak{h}}{2} \beta \left(-\frac{x}{v_t}(1+\epsilon)\right) + \frac{\mathfrak{h}}{2} \beta \left(-\frac{x}{v'_t}\right)\right] \\ &= \frac{\beta}{2}\left[\frac{(1+\mathfrak{h})(1+\epsilon)}{v_t} - 1 - \frac{\mathfrak{h}}{v'_t}\right]. \end{aligned} \tag{113}$$

The integration can be performed analytically, and we can check that the LHS indeed coincides with the RHS.

E. Setting of the pretraining in 2d synthetic data

E.1. Data

We use synthetic data similar to the synthetic data shown in Fig. 5 used in (Petzka et al., 2018) to train our model.

Swiss-roll data is generated by `sklearn.datasets.make_swiss_roll` (Pedregosa et al., 2011) with `noise = 0.5, hole = False`. This data itself is 3-dimensional data, so we project them to 2-dimensional data by using `[0, 2]` axes. After getting data, we normalize it by its `std` ($\approx$ `6.865`) to get data with `std = 1`.

25-Gaussian data is generated by mixture of gaussians. We first generate gaussian distributions with mean in $\{-4, 2, 0, 2, 4\}^2$, std = 0.05, and again divide each sample vector component by its `std` ($\approx$ `2.828`) to get data with `std = 1`.

E.2. Training

We used JAX (Bradbury et al., 2018) and Flax (Heek et al., 2023) to implement our score-based models with neural networks, and Optax (DeepMind et al., 2020) for the training.

As we write in the main part of this paper, we use simple neural network: $(\mathbf{x}, t) \to \mathtt{concat}([\mathbf{x}, t]) \to \left(\mathtt{Dense}(128) \to \mathtt{swish}\right)^3 \to \mathtt{Dense}(2) \to \boldsymbol{s}_\theta(\mathbf{x}, t)$ with default initialization both in training with Swiss-roll data and 25-Gaussian data. The loss function is calculated by Monte-Carlo sampling in each training step:

1. First we divide the time interval $[T_{\min}, T_{\max}]$ into 1,000 equal parts in advance to get a discretized diffusion time array.

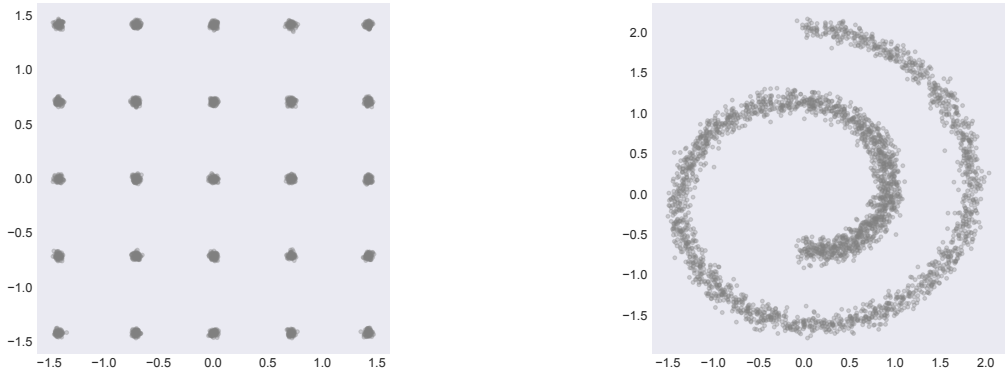

Figure 5. **(Left)** 25-Gaussian data (3,000 samples), and **(Right)** Swiss-roll data (3,000 samples).

2. We take $N_{\text{batch}} = 512$ batches of data $\{\mathbf{x}_i\}_{i=1,2,\ldots,N_{\text{batch}}}$ and for each data point we take a discretized diffusion time t_i uniformly from the array that we discretized in the first step.

3. We compute the signal $\alpha(t_i)$ and noise $\sigma(t_i)$ at time t_i, computed from the definition of SDE, and take the following quantity as loss function

$$\frac{1}{N_{\text{batch}}}\sum_{i=1}^{N_{\text{batch}}}\frac{g(t_i)^2}{2}\left\|\frac{\mathbf{x}_{i,t}-\alpha(t_i)\mathbf{x}_i}{\sigma(t_i)^2}+\boldsymbol{s}_\theta(\mathbf{x}_{i,t},t_i)\right\|^2, \tag{114}$$

where $\mathbf{x}_{i,t}\sim\mathcal{N}(\alpha(t_i)\mathbf{x}_i,\sigma(t_i)^2 I)$ is the Monte-Carlo sample.

The signal and noise functions $\alpha(t)$ and $\sigma(t)$ are:

$$\begin{aligned}&\text{SIMPLE:}\quad \alpha(t)=e^{-\frac{\beta}{4}t^2},\quad \sigma(t)^2=1-e^{-\frac{\beta}{2}t^2},\\ &\text{COSINE:}\quad \alpha(t)=\cos\left(\frac{\pi}{2}t\right),\quad \sigma(t)^2=1-\cos\left(\frac{\pi}{2}t\right).\end{aligned} \tag{115}$$

These expressions are derived from general argument of SDE. In general, once the SDE

$$d\mathbf{x}_t = f(t)\mathbf{x}+g(t)d\mathbf{w}_t \tag{116}$$

is given, the conditional probability from time $t=0$ to t is given by

$$p_{t|0}(\mathbf{x}_t|\mathbf{x}_0)=\mathcal{N}(\mathbf{x}_t|\alpha(t)\mathbf{x}_0,\sigma(t)^2 I), \tag{117}$$

where

$$\alpha(t)=e^{\int_0^t f(\xi)\mathrm{d}\xi},\quad \sigma(t)^2=\alpha(t)^2\int_0^t\frac{g(\xi)^2}{\alpha(\xi)^2}\mathrm{d}\xi, \tag{118}$$

which is essentially equivalent to the formula in (Karras et al., 2022).

In training, 3,000 data points are taken in advance, and stochastic gradients are computed based on the Monte-Carlo loss function (114) with batch size 512 mini-batches. The gradients are used for optimization of the neural network $\boldsymbol{s}_\theta(\mathbf{x}_t,t)$ with Adam-optimizer with learning rate `1e-3` and default values determined by Optax (DeepMind et al., 2020). We train our models 16,000 epochs.

E.3. Inference

Here we show some instances on generated data by our pretrained model in Figs. 6 to 9. The figures are plotted with identical points in discretized time, $t = 1,\ 0.18,\ 0.1,\ 0.01$.

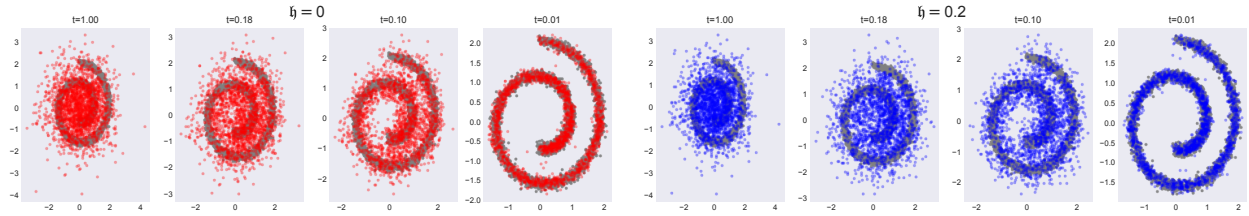

Figure 6. SIMPLE-SDE pretrained model on Swiss-roll data with **(Left)** probability flow ODE ($\hbar = 0$), **(Right)** SDE (9) with $\hbar = 0.2$.

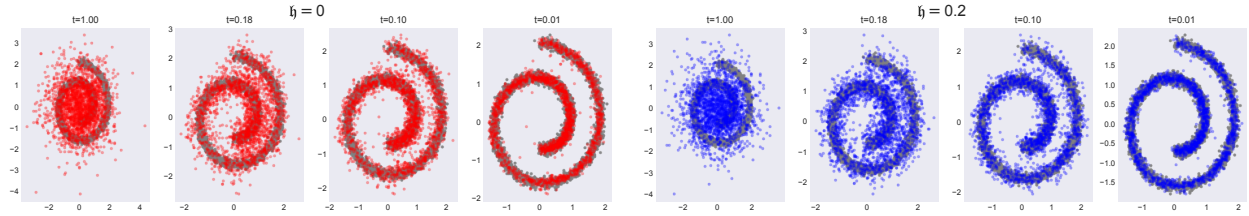

Figure 7. COSINE-SDE pretrained model on Swiss-roll data with **(Left)** probability flow ODE ($\hbar = 0$), **(Right)** SDE (9) with $\hbar = 0.2$.

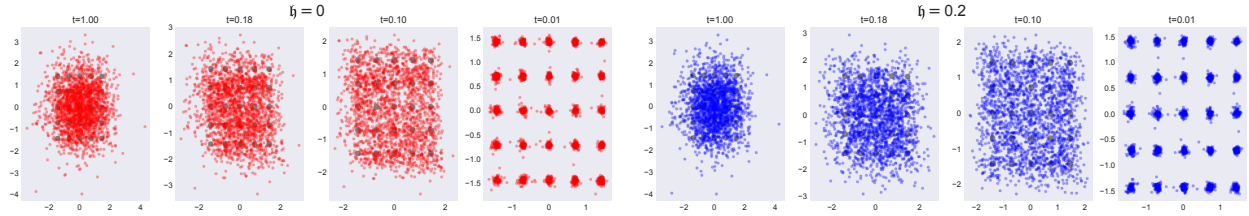

Figure 8. SIMPLE-SDE pretrained model on 25-Gaussian data with **(Left)** probability flow ODE ($\hbar = 0$), **(Right)** SDE (9) with $\hbar = 0.2$.

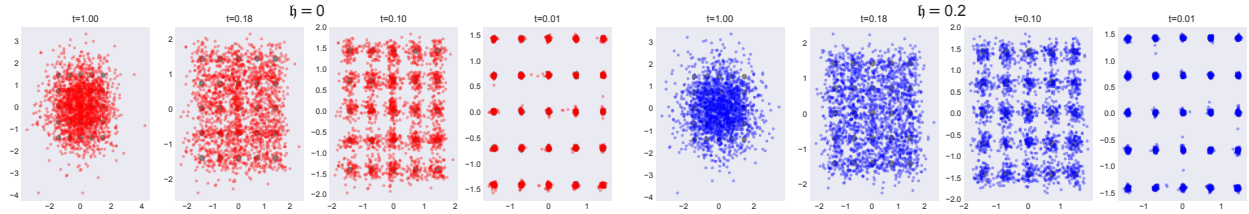

Figure 9. COSINE-SDE pretrained model on 25-Gaussian data with **(Left)** probability flow ODE ($\hbar = 0$), **(Right)** SDE (9) with $\hbar = 0.2$.

F. Derivative numerical error estimations in Sec. 5.2

F.1. Estimation of the local errors (21)

For simplicity, we omit "$\hbar = 0$" script here. In this notation, error of $\log q_t(\mathbf{x})$ and local errors (21) are

$$\begin{aligned}
\text{err}(N[\log q_t(\mathbf{x})]) &= \log q_t(\mathbf{x}) - N[\log q_t(\mathbf{x})], \\
\text{err}(\hat{\nabla} N[\log q_t(\mathbf{x})]) &= \nabla \log q_t(\mathbf{x}) - \hat{\nabla} N[\log q_t(\mathbf{x})], \\
\text{err}(\hat{\nabla}^2 N[\log q_t(\mathbf{x})]) &= \nabla^2 \log q_t(\mathbf{x}) - \hat{\nabla}^2 N[\log q_t(\mathbf{x})].
\end{aligned} \tag{119}$$

We use a third-party solver `scipy.integrate.solve_ivp` (Virtanen et al., 2020), and it has inputs `atol` and `rtol` that control the errors in the subroutine, to make it clear, let us call the numerical value as $N_{\texttt{atol},\texttt{rtol}}[...]$. It is plausible to expect the order of errors are same between numerical calculations based on the same order of tolerances, say

$$O\Big(\text{err}(\hat{\nabla}^{0 \text{ or } 1 \text{ or } 2} N_{(\texttt{1.1*atol}),(\texttt{1.1*rtol})}[\log q_t(\mathbf{x})])\Big) = O\Big(\text{err}(\hat{\nabla}^{0 \text{ or } 1 \text{ or } 2} N_{\texttt{atol},\texttt{rtol}}[\log q_t(\mathbf{x})])\Big), \tag{120}$$

where O means order estimate. Based on this observation, we can estimate each error. For example,

$$\begin{aligned}
&\underbrace{N_{(\texttt{1.1*atol}),(\texttt{1.1*rtol})}[\log q_t(\mathbf{x})]}_{\log q_t(\mathbf{x})+\text{err}(N_{(\texttt{1.1*atol}),(\texttt{1.1*rtol})}[\log q_t(\mathbf{x})])} - \underbrace{N_{\texttt{atol},\texttt{rtol}}[\log q_t(\mathbf{x})]}_{\log q_t(\mathbf{x})+\text{err}(N_{\texttt{atol},\texttt{rtol}}[\log q_t(\mathbf{x})])} \\
&= \text{err}(N_{(\texttt{1.1*atol}),(\texttt{1.1*rtol})}[\log q_t(\mathbf{x})]) - \text{err}(N_{\texttt{atol},\texttt{rtol}}[\log q_t(\mathbf{x})]) \\
&= O\Big(\text{err}(N_{\texttt{atol},\texttt{rtol}}[\log q_t(\mathbf{x})])\Big).
\end{aligned} \tag{121}$$

F.2. `subtraction`: Estimation of the local errors (21) by subtraction

On the order estimations for differential operators, we have 2 choices. First choice is estimating them simply by

$$\text{err}(\hat{\nabla}^{1\text{ or }2}N[\log q_t(\mathbf{x})]) = \hat{\nabla}^{1\text{ or }2}N_{(\texttt{1.1*atol}),(\texttt{1.1*rtol})}[\log q_t(\mathbf{x})] - \hat{\nabla}^{1\text{ or }2}N_{\texttt{atol},\texttt{rtol}}[\log q_t(\mathbf{x})] \tag{122}$$

We name error estimation scheme based on this by `subtraction`. this method is straightforward, however, we need doubled computation time, and we introduce more time-efficient estimation in the next section.

F.3. `model`: Estimation of the local errors (21) by Taylor expansion and score models

Next choice is simply based on the error

$$N[\log q_t(\mathbf{x})] = \log q_t(\mathbf{x}) + \text{err}(N[\log q_t(\mathbf{x})]). \tag{123}$$

By applying the discrete differential (19), we can get

$$\hat{\nabla}N[\log q_t(\mathbf{x})] = \begin{bmatrix} \frac{N[\log q_t(\mathbf{x}_t+[\Delta x,0]^\top)]-N[\log q_t(\mathbf{x}_t-[\Delta x,0]^\top)]}{2\Delta x} \\ \frac{N[\log q_t(\mathbf{x}_t+[0,\Delta x]^\top)]-N[\log q_t(\mathbf{x}_t-[0,\Delta x]^\top)]}{2\Delta x} \end{bmatrix}. \tag{124}$$

It would be sufficient to show x component:

$$\begin{aligned}
&\hat{\partial}_x N[\log q_t(\mathbf{x})] \\
&= \frac{N[\log q_t(\mathbf{x}+[\Delta x,0]^\top)] - N[\log q_t(\mathbf{x}-[\Delta x,0]^\top)]}{2\Delta x} \\
&= \frac{\log q_t(\mathbf{x}+[\Delta x,0]^\top) - \log q_t(\mathbf{x}-[\Delta x,0]^\top)}{2\Delta x} + \frac{\text{err}(N[\log q_t(\mathbf{x}+[\Delta x,0]^\top)]) - \text{err}(N[\log q_t(\mathbf{x}-[\Delta x,0]^\top)])}{2\Delta x} \\
&= \frac{\partial_x \log q_t(\mathbf{x}) 2\Delta x + O(\Delta x^3)}{2\Delta x} + \frac{O(\text{err}(N[\log q_t(\mathbf{x})]))}{\Delta x} \\
&= \partial_x \log q_t(\mathbf{x}) + O(\Delta x^2) + \frac{O(\text{err}(N[\log q_t(\mathbf{x})]))}{\Delta x},
\end{aligned} \tag{125}$$

where we use the facts: 1) $O(\Delta x^2)$ term vanishes from the numerator thanks to the symmetric descrete differential, 2) orders of $\text{err}(N[\log q_t(\mathbf{x}))]$ and $\text{err}(N[\log q_t(\mathbf{x}\pm[\Delta x,0]^\top)])$ are same, in the 3rd line. This fact is achieved by the Runge-Kutta integrator. Therefore, by subtracting the true value from both side, we get

$$\text{err}(\hat{\nabla}N[\log q_t(\mathbf{x})]) = \nabla \log q_t(\mathbf{x}) - \hat{\nabla}N[\log q_t(\mathbf{x})] = O(\Delta x^2) + \frac{O(\text{err}(N[\log q_t(\mathbf{x})]))}{\Delta x} \tag{126}$$

We can derive the error for ∇^2 similarly:

$$
\begin{aligned}
&\hat{\nabla}^2 N[\log q_t(\mathbf{x})] \\
&= \frac{N[\log q_t(\mathbf{x}+[\Delta x,0]^\top)] + N[\log q_t(\mathbf{x}-[\Delta x,0]^\top)] + N[\log q_t(\mathbf{x}+[0,\Delta x]^\top)] + N[\log q_t(\mathbf{x}-[0,\Delta x]^\top)] - 4N[\log q_t(\mathbf{x})]}{\Delta x^2} \\
&= \frac{\log q_t(\mathbf{x}+[\Delta x,0]^\top) + \log q_t(\mathbf{x}-[\Delta x,0]^\top) + \log q_t(\mathbf{x}+[0,\Delta x]^\top) + \log q_t(\mathbf{x}-[0,\Delta x]^\top) - 4\log q_t(\mathbf{x})}{\Delta x^2} \\
&\quad + \frac{\begin{array}{r}\mathrm{err}(N[\log q_t(\mathbf{x}+[\Delta x,0]^\top)]) + \mathrm{err}(N[\log q_t(\mathbf{x}-[\Delta x,0]^\top)]) + \mathrm{err}(N[\log q_t(\mathbf{x}+[0,\Delta x]^\top)]) + \mathrm{err}(N[\log q_t(\mathbf{x}-[0,\Delta x]^\top)]) \\ -4\mathrm{err}(N[\log q_t(\mathbf{x})])\end{array}}{\Delta x^2} \\
&= \frac{\nabla^2 \log q_t(\mathbf{x})\Delta x^2 + O(\Delta x^4)}{\Delta x^2} + \frac{O(\mathrm{err}(N[\log q_t(\mathbf{x})]))}{\Delta x^2} \\
&= \nabla^2 \log q_t(\mathbf{x}) + O(\Delta x^2) + \frac{O(\mathrm{err}(N[\log q_t(\mathbf{x})]))}{\Delta x^2},
\end{aligned} \tag{127}
$$

where we use that $O(\Delta x^3)$ term in the numerator vanishes because we use the 5-point approximation of the laplacian. Now we get

$$
\mathrm{err}(\hat{\nabla}^2 N[\log q_t(\mathbf{x})]) = \nabla^2 \log q_t(\mathbf{x}) - \hat{\nabla}^2 N[\log q_t(\mathbf{x})] = O(\Delta x^2) + \frac{O(\mathrm{err}(N[\log q_t(\mathbf{x})]))}{\Delta x^2} \tag{128}
$$

In summary, we conclude each error as

$$
\begin{aligned}
\mathrm{err}(\hat{\nabla} N[\log q_t(\mathbf{x})]) &= \boldsymbol{C}_t^{(1)}(\mathbf{x})\Delta x^2 + \frac{O(\mathrm{err}(N[\log q_t(\mathbf{x})]))\mathbf{1}}{\Delta x}, \\
\mathrm{err}(\hat{\nabla}^2 N[\log q_t(\mathbf{x})]) &= C_t^{(2)}(\mathbf{x})\Delta x^2 + \frac{O(\mathrm{err}(N[\log q_t(\mathbf{x})]))}{\Delta x^2},
\end{aligned} \tag{129}
$$

where $\mathbf{1}$ is the vector with all 1 components, and $\boldsymbol{C}_t^{(1)}(\mathbf{x}), C_t^{(2)}(\mathbf{x})$ are determined by the Taylor expansion remainder terms. Typically, these are approximated by

$$
\begin{aligned}
\boldsymbol{C}_t^{(1)}(\mathbf{x}) &\approx \text{3rd derivative of } \log q_t(\mathbf{x}), \\
C_t^{(2)}(\mathbf{x}) &\approx \text{4th derivative of } \log q_t(\mathbf{x}).
\end{aligned} \tag{130}
$$

Of course, we cannot access to the functions (130), however, it would be plausible to regard

$$
O(\nabla \log q_t(\mathbf{x})) = O(\boldsymbol{s}_\theta(\mathbf{x},t)), \tag{131}
$$

because the objective of the diffusion model training is to achieve $\boldsymbol{s}_\theta(\mathbf{x},t) \approx \nabla \log p_t(\mathbf{x})$ and in the ideal case, $q_t = p_t$. By using this assumption, we regard

$$
\begin{aligned}
\boldsymbol{C}_t^{(1)}(\mathbf{x}) &\approx -\nabla(\nabla \cdot \boldsymbol{s}_\theta(\mathbf{x},t)) \\
C_t^{(2)}(\mathbf{x}) &\approx -\nabla^2(\nabla \cdot \boldsymbol{s}_\theta(\mathbf{x},t))
\end{aligned} \tag{132}
$$

and estimate the local errors by

$$
\begin{aligned}
\mathrm{err}(\hat{\nabla} N[\log q_t(\mathbf{x})]) &\approx |\nabla(\nabla \cdot \boldsymbol{s}_\theta(\mathbf{x},t))\Delta x^2| + \left|\frac{O(\mathrm{err}(N[\log q_t(\mathbf{x})]))\mathbf{1}}{\Delta x}\right|, \\
\mathrm{err}(\hat{\nabla}^2 N[\log q_t(\mathbf{x})]) &\approx |\nabla^2(\nabla \cdot \boldsymbol{s}_\theta(\mathbf{x},t))\Delta x^2| + \left|\frac{O(\mathrm{err}(N[\log q_t(\mathbf{x})]))}{\Delta x^2}\right|,
\end{aligned} \tag{133}
$$

where $O(\mathrm{err}(N[\log q_t(\mathbf{x})]))$ can be estimated by using (121). We name error estimation scheme based on this by `model`.

F.4. Integral of the local errors

Now we go back to Theorem 4.2 to explain our error estimation for $O(\mathfrak{h})$ correction to the (negative) log-likelihood. To calculate $O(\mathfrak{h})$ terms, we consider the paired ODE constructed by

$$
\begin{aligned}
\dot{\mathbf{x}}_t &= \boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t,t), \\
\dot{\delta\mathbf{x}_t} &= \delta\boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t,\delta\mathbf{x}_t,t) \quad (\text{by } \hat{\nabla} N[\log q_t(\mathbf{x}_t)]), \\
\dot{\delta\log q_t} &= \nabla \cdot \delta\boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t,\delta\mathbf{x}_t,t) \quad (\text{by } \hat{\nabla}^2 N[\log q_t(\mathbf{x}_t)]),
\end{aligned} \tag{134}
$$

as we wrote in Algorithm 1. In this expression, there is no numerical error in the RHS of $\dot{\mathbf{x}}_t$ except for the `float32` precision that is the default of the deep learning framework. In the RHS of $\dot{\delta\mathbf{x}_t}$, we have discretization of differential $\hat{\nabla}$ and numerical integral for $\log q_t^{\mathfrak{h}=0}(\mathbf{x}_t)$, and as we noted, we omit $\mathfrak{h}=0$ and call it as $\log q_t(\mathbf{x}_t)$. By using the error estimation (133), we rewrite ODE for $\delta\mathbf{x}_t$ as

$$\begin{aligned}\dot{\delta\mathbf{x}_t} &= (\delta\mathbf{x}_t\cdot\nabla)\boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t,t) - \frac{g(t)^2}{2}(\boldsymbol{s}_\theta(\mathbf{x}_t,t) - \underbrace{\hat{\nabla}N\log q_t(\mathbf{x}_t)}_{\nabla\log q_t(\mathbf{x}_t)+\mathrm{err}(\hat{\nabla}N[\log q_t(\mathbf{x}_t)])}) \\ &= (\delta\mathbf{x}_t\cdot\nabla)\boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t,t) - \frac{g(t)^2}{2}(\boldsymbol{s}_\theta(\mathbf{x}_t,t) - \nabla\log q_t(\mathbf{x}_t)) + \frac{g(t)^2}{2}\mathrm{err}(\hat{\nabla}N[\log q_t(\mathbf{x}_t)]).\end{aligned} \tag{135}$$

When we apply the solver, basically it discretize the system as

$$\delta\mathbf{x}_{t+\epsilon} = \delta\mathbf{x}_t + \epsilon\Big((\delta\mathbf{x}_t\cdot\nabla)\boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t,t) - \frac{g(t)^2}{2}(\boldsymbol{s}_\theta(\mathbf{x}_t,t) - \nabla\log q_t(\mathbf{x}_t)) + \frac{g(t)^2}{2}\mathrm{err}(\hat{\nabla}N[\log q_t(\mathbf{x}_t)])\Big) + \dots, \tag{136}$$

and truncate $\dots$ terms up to certain finite order, and apply this recursive equation from $\delta\mathbf{x}_0$. For simplicity, we consider the liner term only here, and split $\delta\mathbf{x}_t = \delta\mathbf{x}_t^{\mathrm{true}} + \mathrm{err}_t^{(1)}$, then

$$\begin{aligned}&\begin{pmatrix}\delta\mathbf{x}_{t+\epsilon}^{\mathrm{true}}\\ +\\ \mathrm{err}_{t+\epsilon}^{(1)}\end{pmatrix} \\ &= \begin{pmatrix}\delta\mathbf{x}_t^{\mathrm{true}}\\ +\\ \mathrm{err}_t^{(1)}\end{pmatrix} + \epsilon\Bigg(\left(\begin{pmatrix}\delta\mathbf{x}_t^{\mathrm{true}}\\ +\\ \mathrm{err}_t^{(1)}\end{pmatrix}\cdot\nabla\right)\boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t,t) - \frac{g(t)^2}{2}(\boldsymbol{s}_\theta(\mathbf{x}_t,t) - \nabla\log q_t(\mathbf{x}_t)) + \frac{g(t)^2}{2}\mathrm{err}(\hat{\nabla}N[\log q_t(\mathbf{x}_t)])\Bigg) \\ &= \begin{pmatrix}\delta\mathbf{x}_t^{\mathrm{true}} + \epsilon\Big((\delta\mathbf{x}_t^{\mathrm{true}}\cdot\nabla)\boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t,t) - \frac{g(t)^2}{2}(\boldsymbol{s}_\theta(\mathbf{x}_t,t) - \nabla\log q_t(\mathbf{x}_t))\Big)\\ +\\ \mathrm{err}_t^{(1)} + \epsilon\Big((\mathrm{err}_t^{(1)}\cdot\nabla)\boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t,t) + \frac{g(t)^2}{2}\mathrm{err}(\hat{\nabla}N[\log q_t(\mathbf{x}_t)])\Big)\end{pmatrix},\end{aligned} \tag{137}$$

and we get the ODE for time t error by taking continuous time limit:

$$\dot{\mathrm{err}}_t^{(1)} = (\mathrm{err}_t^{(1)}\cdot\nabla)\boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t,t) + \frac{g(t)^2}{2}\mathrm{err}(\hat{\nabla}N[\log q_t(\mathbf{x}_t)]). \tag{138}$$

Note that $\mathrm{err}_t^{(1)}$ is a vector with same dimension to $\mathbf{x}_t, \delta\mathbf{x}_t, \boldsymbol{s}_\theta$, and $\boldsymbol{f}_\theta^{\mathrm{PF}}$. As same, we can get the ODE for $\dot{\delta\log q_t}$ term. The discrete ODE is

$$\begin{aligned}&\delta\log q_{t+\epsilon} \\ &= \delta\log q_t + \epsilon\Big((\delta\mathbf{x}_t\cdot\nabla)\nabla\cdot\boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t,t) - \frac{g(t)^2}{2}(\nabla\cdot\boldsymbol{s}_\theta(\mathbf{x}_t,t) - \nabla^2\log q_t(\mathbf{x}_t)) + \frac{g(t)^2}{2}\mathrm{err}(\hat{\nabla}^2N[\log q_t(\mathbf{x}_t)])\Big) + \dots,\end{aligned} \tag{139}$$

and as same, by splitting true value and error value, we get

$$\begin{aligned}&\begin{pmatrix}\delta\log q_{t+\epsilon}^{\mathrm{true}}\\ +\\ \mathrm{err}_{t+\epsilon}^{(2)}\end{pmatrix} \\ &= \begin{pmatrix}\delta\log q_t^{\mathrm{true}} + \epsilon\Big((\delta\mathbf{x}_t^{\mathrm{true}}\cdot\nabla)\nabla\cdot\boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t,t) - \frac{g(t)^2}{2}(\nabla\cdot\boldsymbol{s}_\theta(\mathbf{x}_t,t) - \nabla^2\log q_t(\mathbf{x}_t))\Big)\\ +\\ \mathrm{err}_t^{(2)} + \epsilon\Big((\mathrm{err}_t^{(1)}\cdot\nabla)\nabla\cdot\boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t,t) + \frac{g(t)^2}{2}\mathrm{err}(\hat{\nabla}^2N[\log q_t(\mathbf{x}_t)])\Big)\end{pmatrix},\end{aligned} \tag{140}$$

and get the ODE for $\mathrm{err}_t^{(2)}$ as

$$\dot{\mathrm{err}}_t^{(2)} = (\mathrm{err}_t^{(1)}\cdot\nabla)\nabla\cdot\boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t,t) + \frac{g(t)^2}{2}\mathrm{err}(\hat{\nabla}^2N[\log q_t(\mathbf{x}_t)]). \tag{141}$$

In summary, we get the ODE system

$$\begin{aligned}
\dot{\mathbf{x}}_t &= \boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t, t), \\
\dot{\delta\mathbf{x}}_t &= \delta\boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t, \delta\mathbf{x}_t, t) \quad (\text{by } \hat{\nabla} N[\log q_t(\mathbf{x}_t)]), \\
\dot{\delta\log q_t} &= \nabla\cdot\delta\boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t, \delta\mathbf{x}_t, t) \quad (\text{by } \hat{\nabla}^2 N[\log q_t(\mathbf{x}_t)]), \\
\dot{\mathrm{err}}_t^{(1)} &= (\mathrm{err}_t^{(1)}\cdot\nabla)\boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t, t) + \frac{g(t)^2}{2}\mathrm{err}(\hat{\nabla} N[\log q_t(\mathbf{x}_t)]), \\
\dot{\mathrm{err}}_t^{(2)} &= (\mathrm{err}_t^{(1)}\cdot\nabla)\nabla\cdot\boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t, t) + \frac{g(t)^2}{2}\mathrm{err}(\hat{\nabla}^2 N[\log q_t(\mathbf{x}_t)]).
\end{aligned} \tag{142}$$

The ERRORS in Table 1 are calculated by solving more conservative (or over-estimated) ODE:

$$\begin{aligned}
\dot{\mathrm{err}}_t^{(1)} &= \left|(\mathrm{err}_t^{(1)}\cdot\nabla)\boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t, t)\right| + \left|\frac{g(t)^2}{2}\mathrm{err}(\hat{\nabla} N[\log q_t(\mathbf{x}_t)])\right|, \\
\dot{\mathrm{err}}_t^{(2)} &= \left|(\mathrm{err}_t^{(1)}\cdot\nabla)\nabla\cdot\boldsymbol{f}_\theta^{\mathrm{PF}}(\mathbf{x}_t, t)\right| + \left|\frac{g(t)^2}{2}\mathrm{err}(\hat{\nabla}^2 N[\log q_t(\mathbf{x}_t)])\right|,
\end{aligned} \tag{143}$$

by using approximations (133) and the error final value for $\log q_T$ as

$$|\mathrm{err}_T^{(1)}\cdot\nabla\log\pi(\mathbf{x}_T)| + |\mathrm{err}_T^{(2)}| \tag{144}$$

F.5. Visualization of local error estimates

Of course, the degree of final error (144) strongly depends on the order of the local errors $\mathrm{err}(\hat{\nabla} N[\log q_t(\mathbf{x}_t)])$ and $\mathrm{err}(\hat{\nabla}^2 N[\log q_t(\mathbf{x}_t)])$. To see its order, we plot $\log_{10}$ scales of the local error functions in Fig. 10.

If we believe $\mathrm{err}(N[\log q_t(\mathbf{x})])$ is suppressed within the small tolerance value, we take it as 10^{-5} by the Runge-Kutta algorithm, the order of local errors are depending on the coefficients of Δx^2. To check these values, we plot mean and std values of $\mathrm{tr}\nabla(\nabla\cdot\boldsymbol{s}_\theta(\mathbf{x}, t))$ and $\nabla^2(\nabla\cdot\boldsymbol{s}_\theta(\mathbf{x}, t))$ by sampling 500 points $\mathbf{x}\sim\mathrm{Uniform}(\min(\texttt{validation set})-0.1, \max(\texttt{validation set})-0.1)$ at each time t in Figs. 11 to 14. Simultaneously, we plot colored contours that corresponds the value of local error estimates based on (132) with $\Delta x = 0.01$, that are exactly same as the values on dashed line in Fig. 10.

From these figures, one can see that the estimated local errors are almost located at safe region, i.e., errors are negative in log scale. As one can see, the 25-Gaussian case Figs. 13 and 14, the maximum values are slightly inside red regions, and we may be careful about it, but we leave further study as a future work.

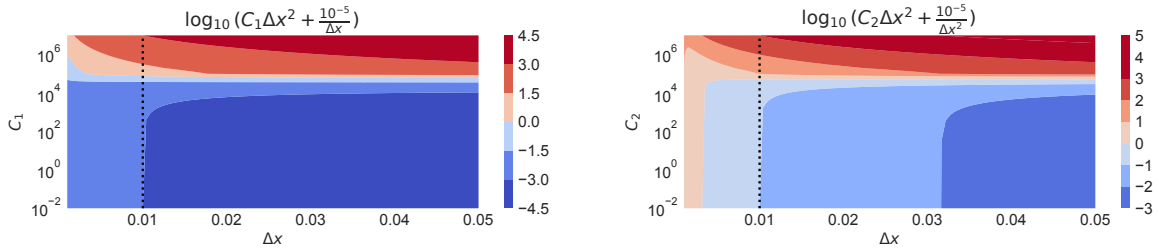

Figure 10. The order of local errors. **(Left)** Contour plot corresponding to $\mathrm{err}(\hat{\nabla} N[\log q_t(\mathbf{x}_t)])$. **(Right)** Contour plot corresponding to $\mathrm{err}(\hat{\nabla}^2 N[\log q_t(\mathbf{x}_t)])$, based on (132). Blue region has negative power, and relatively safe. Red region has positive power, and dangerous. The dotted line corresponds $\Delta x = 0.01$ that is the value used in Table 1. The value 10^{-5} in the 2nd-term numerator is the typical tolerance value of `0th-logqSolver` in Algorithm 1, that corresponds to $\mathrm{err}(N[\log q_t(\mathbf{x})])$.

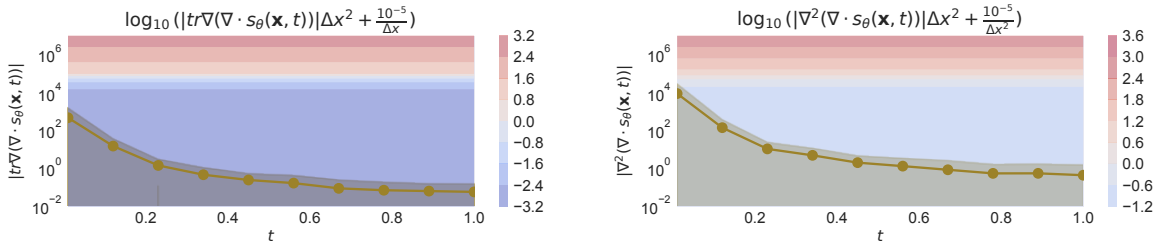

Figure 11. Local error mean/std plot with pretrained model with SIMPLE-SDE trained by Swiss-roll.

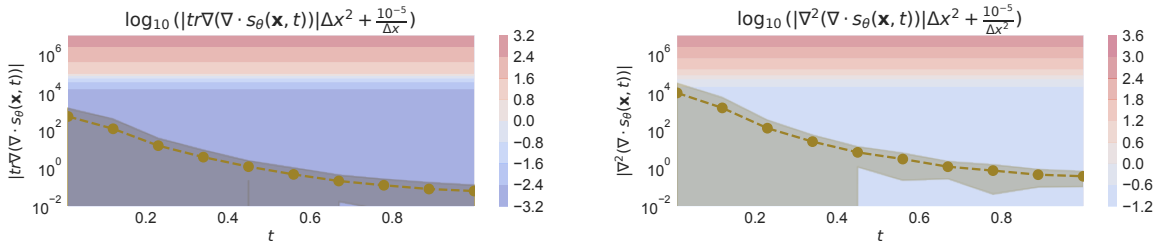

Figure 12. Local error mean/std plot with pretrained model with COSINE-SDE trained by Swiss-roll.

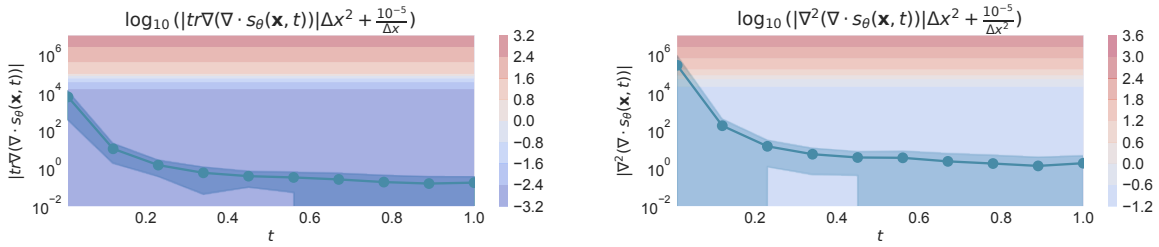

Figure 13. Local error mean/std plot with pretrained model with SIMPLE-SDE trained by 25-Gaussian.

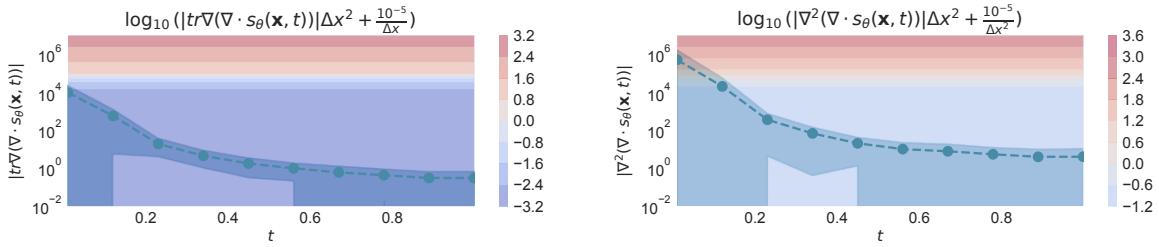

Figure 14. Local error mean/std plot with pretrained model with COSINE-SDE trained by 25-Gaussian.