

Schrödinger bridge-type diffusion models as an extension of variational autoencoders

Kentaro Kaba^{1,*}, Reo Shimizu², Masayuki Ohzeki^{1,2,3,4} and Yuki Sughiyama²¹Department of Physics, *Institute of Science Tokyo*, Meguro-ku, Tokyo 152-9551, Japan²Graduate School of Information Sciences, *Tohoku University*, Sendai, Miyagi 980-9564, Japan³Research and Education Institute for Semiconductors and Informatics, *Kumamoto University*, Kumamoto, Kumamoto 860-8555, Japan⁴Sigma-i Co., Ltd., Minato-ku, Tokyo 108-0075, Japan

(Received 20 February 2025; accepted 4 August 2025; published 3 September 2025)

Generative diffusion models are described by time-forward and -backward stochastic differential equations to connect the data and prior distributions. While conventional diffusion models (e.g., score-based models) only learn the backward process, more flexible frameworks have been proposed to also learn the forward process by employing the Schrödinger bridge (SB). However, due to the complexity of the mathematical structure behind SB-type models, we cannot easily give an intuitive understanding of their objective function. In this work, we propose a unified framework to construct diffusion models by reinterpreting the SB-type models as an extension of variational autoencoders. In this context, the data processing inequality plays a crucial role. As a result, we find that the objective function consists of the prior loss and drift matching parts, which enable us to reduce the numerical cost of training the forward process. Furthermore, we discuss the overfitting problem in the SB-type models in this framework.

DOI: [10.1103/dxp7-4hby](https://doi.org/10.1103/dxp7-4hby)

I. INTRODUCTION

Generative diffusion model [1–8] is one of the likelihood-based generative models that have successfully synthesized high-quality samples such as image [9,10] and audio [11,12]. In the framework of the diffusion model, the model distribution that mimics the training dataset is defined by a distribution transported from a simple prior through a stochastic differential equation (SDE). This framework enables us to generate new samples by learning the transport dynamics via neural networks (NNs).

A prominent example of generative diffusion models is the score-based model (SBM) [3,13]. In SBMs, the data distribution is gradually perturbed by the addition of noise and eventually converges to a normal distribution that does not have any information. This procedure is commonly referred to as the noising process. Subsequently, the time reversal of the noising process is learned to establish a generative process, which synthesizes new samples by transporting the prior distribution (i.e., the normal distribution). Although SBMs are used in many situations [9–12,14], they have two main limitations. First, since learning of the reverse process requires computation of the score function (the gradient of the log probability distribution), the noising process must be restricted to a simple form (e.g., linear or degenerate), where the score function can be evaluated analytically. Second, when the data distribution far deviates from the prior, the time required

for the noising process becomes longer, which results in substantial delays in sample generation.

To overcome these limitations, a more flexible model has recently been proposed using the Schrödinger bridge (SB) [4–8]. The SB is a stochastic optimization technique that derives an SDE to connect any two arbitrary probability distributions over a given time horizon [15,16]. This technique allows us to use highly structured SDEs with NNs for the noising process and accelerates the generation of new samples. However, the mathematical structure underlying SB-type diffusion models is complicated, because it involves the mapping from partial differential equations that solve SB problems to SDEs of noising and generative processes by using the nonlinear Feynman-Kac formula [7,17]. This complicated structure inhibits our intuitive understanding of the objective function used in the training of SB-type models.

In actual training of SB-type models, the computational cost is much higher than that of SBMs, because we need to train not only the generative process but also the noising process. Furthermore, the overfitting problem, which is reported in SBMs [18,19], is also expected to occur in SB-type models. These problems have not been sufficiently settled yet, due to the complexity of its mathematical structure mentioned above.

In this work, we attempt to reinterpret SB-type models as a relatively straightforward extension of variational autoencoders (VAEs) [20,21]. VAEs are generative models that learn two mappings between data and latent spaces: The encoder that maps the data into the latent variables and the decoder that represents its inverse. Recent work [22] has rederived SBMs in the VAE framework by associating the noising process with the encoder and the generative process with the decoder, respectively. However, the approach in that work has omitted the training of the encoder and has limited the scope to the derivation of SBMs. In this paper, taking the training of the encoder into account, we propose a method to construct

*Contact author: kaba.k.df05@m.isct.ac.jp

the objective function for SB-type models. As a result, it is revealed that the objective function is composed of the prior loss and drift matching parts, which characterize the training of NNs in the encoder and the decoder, respectively. Owing to these two parts, we can reduce the numerical cost of the training of the SB-type models. Moreover, by interrupting the training of the encoder, it is shown that the overfitting problem can be mitigated.

This paper is organized as follows. In Sec. II, we briefly introduce the VAE framework and its training scheme. In Sec. III, we construct the diffusion models in the framework of VAEs and derive its objective function. In Sec. IV, we discuss the connection between our work and previous studies. In Sec. V, we present numerical experiments to demonstrate the reduction of the training cost and overfitting in the SB-type models. Finally, in Sec. VI, we summarize our work and discuss future directions.

II. VARIATIONAL AUTOENCODERS

We begin with a brief introduction to variational autoencoders. Let $\{x_i\}_{i=1}^n \in \mathcal{X}^n$ be a dataset that consists of i.i.d. samples from a true distribution $p_{\text{true}}(x)$ on $\mathcal{X}$. Then, we can define its empirical distribution on $\mathcal{X}$ as $\mu(x) = (1/n) \sum_i \delta(x - x_i)$. $q_\theta(x)$ denotes a generative model with a parameter θ . In addition, the training of the model based on the data distribution $\mu(x)$ is performed by solving

$$\theta^* = \arg \min_{\theta} D_{\text{KL}}(\mu(x) \| q_\theta(x)), \quad (1)$$

which represents the minimization of the Kullback-Leibler (KL) divergence:

$$D_{\text{KL}}(\mu(x) \| q_\theta(x)) := \mathbb{E}_{X \sim \mu} \left[\log \frac{\mu(X)}{q_\theta(X)} \right]. \quad (2)$$

That is equivalent to the maximum log-likelihood training [23].

In VAEs, the model $q_\theta(x)$ is defined by the marginal distribution of $Q_\theta(x, z)$ as

$$q_\theta(x) := \int dz Q_\theta(x, z) = \int dz Q_\theta(x|z) \pi(z). \quad (3)$$

Here, $\pi(z)$ is a known distribution of a latent variable $z \in \mathcal{Z}$, which is called the prior [24]; $Q_\theta(x|z)$ is a conditional distribution of x given z . Since this distribution carries information from the latent space $\mathcal{Z}$ to the data space $\mathcal{X}$, it is called the decoder.

In VAEs, instead of solving Eq. (1) directly, we consider the optimization problem of its upper bound by employing the data processing inequality [25] (see Appendix A):

$$D_{\text{KL}}(\mu(x) \| q_\theta(x)) \leq D_{\text{KL}}(P_\phi(x, z) \| Q_\theta(x, z)). \quad (4)$$

That minimizes the KL divergence on the joint space $\mathcal{X} \times \mathcal{Z}$. Here, $P_\phi(x, z)$ is the joint distribution extended from $\mu(x)$ with a conditional distribution $P_\phi(z|x)$:

$$P_\phi(x, z) = P_\phi(z|x) \mu(x). \quad (5)$$

In contrast to the decoder $Q_\theta(x|z)$, since $P_\phi(z|x)$ sends information from $\mathcal{X}$ to $\mathcal{Z}$ with the parameter ϕ , it is called the encoder. The training of VAEs is to find the optimal param-

eters (ϕ^*, θ^*) :

$$(\phi^*, \theta^*) = \arg \min_{\phi, \theta} D_{\text{KL}}(P_\phi(x, z) \| Q_\theta(x, z)). \quad (6)$$

By taking Eqs. (3) and (5) into account, the objective function of VAEs is derived as

$$\begin{aligned} D_{\text{KL}}(P_\phi(x, z) \| Q_\theta(x, z)) \\ = \mathbb{E}_{X \sim \mu} [\log \mu(X)] + \mathbb{E}_{X \sim \mu} [D_{\text{KL}}(P_\phi(z|X) \| \pi(z))] \\ - \mathbb{E}_{X \sim \mu, Z \sim P_\phi(\cdot|X)} [\log Q_\theta(X|Z)]. \end{aligned} \quad (7)$$

Its derivation is shown in Appendix B. The negative sign of the second and third terms is collectively referred to as the evidence lower bound [21]. In practical training, the encoder $P_\phi(z|x)$ and the decoder $Q_\theta(x|z)$ are usually modeled by a normal distribution whose mean and variance are given by neural networks.

After training, we can generate new samples using the optimized parameter θ^* as follows: (1) The latent variable z is sampled from a known prior $\pi(z)$. (2) A new sample x can be generated by the optimal decoder $Q_{\theta^*}(x|z)$ conditioned on z . Here, we note that ϕ^* is not used in the generating process but is used for the training of θ .

To clarify the scheme of this training, we summarize the role of each parameter. On the one hand, training of θ adjusts the model distribution q_θ closer to the data distribution μ , which is the primary goal of all generative models. On the other hand, training for ϕ does not directly contribute to optimizing q_θ because ϕ does not appear in the left-hand side of Eq. (4). That serves more to fill the gap between both sides of the inequality and helps to improve the accuracy of the training of θ . This structure of training will play an important role in revealing the mechanism of diffusion models.

III. DIFFUSION MODELS IN THE FRAMEWORK OF VAEs

In this work, we regard the generative diffusion models as an extension of VAEs. The key idea behind this extension is based on the fact that the data processing inequality (4) holds regardless of the number of latent variables in the joint distribution. Therefore, the number of variables can be extended to infinity, which forms a path along which the probability can be analyzed.

We start with the setup of diffusion models. Let $x_{[0,T]} := \{x_t\}_{t \in [0,T]}$ be a path in the time interval $[0, T]$, and $\mathbb{Q}_\theta[x_{[0,T]}|x_T = z]$ denotes a conditional path probability given the final state $x_T = z$. Since $\mathbb{Q}_\theta[x_{[0,T]}|x_T = z]$ carries information on the latent space $\mathcal{Z}$ [26] along the path $x_{[0,T]}$ time backwardly, it works as the decoder. Setting a prior to $\pi(z)$, the probability of the path $x_{[0,T]}$ is expressed as $\mathbb{Q}_\theta[x_{[0,T]}] = \mathbb{Q}_\theta[x_{[0,T]}|x_T = z] \pi(z)$. Then, the diffusion model $q_\theta(x)$ is defined by the marginal distribution at the initial state x_0 :

$$q_\theta(x) := \int \prod_{t \in [0,T]} dx_t \mathbb{Q}_\theta[x_{[0,T]}] \delta(x_0 - x). \quad (8)$$

Similarly to the previous section, to employ the data processing inequality, we define an encoder on the path $x_{[0,T]}$; $\mathbb{P}_\phi[x_{[0,T]}|x_0 = x]$ denotes a conditional path probability for the encoder. Note that $\mathbb{P}_\phi[x_{[0,T]}|x_0 = x]$ is conditioned on the initial state x_0 of the path, whereas the decoder $\mathbb{Q}_\theta[x_{[0,T]}|x_T =$

$z]$ is conditioned on the final state x_T . Using the data distribution $\mu(x)$, we obtain the path probability $\mathbb{P}_\phi[x_{[0,T]}] = \mathbb{P}_\phi[x_{(0,T)}|x_0 = x]\mu(x)$.

In the diffusion models, the encoder $\mathbb{P}_\phi[x_{(0,T)}|x_0 = x]$ and the decoder $\mathbb{Q}_\theta[x_{[0,T]}|x_T = z]$ are modeled by the following time-forward and -backward SDEs, respectively:

$$dX_t = u_\phi(t, X_t) \cdot dt + g(t) \cdot dW_t, \quad (9)$$

$$dX_t = s_\theta(t, X_t) \tilde{\cdot} dt + g(t) \tilde{\cdot} d\bar{W}_t, \quad (10)$$

where $u_\phi(t, X_t)$ and $s_\theta(t, X_t)$ represent the drifts of the diffusion processes. They are given by certain NNs whose parameters are ϕ and θ . $g(t)$ denotes a given noise intensity, and W_t and $\bar{W}_t$ are independent Wiener processes [27,28]. The two types of products in SDEs (9) and (10) represent the Itô and the reverse-Itô products [27,29], which are defined as

$$f(t, X_t) \cdot dW_t := f(t, X_t)(W_{t+dt} - W_t), \quad (11)$$

$$f(t, X_t) \tilde{\cdot} dW_t := f(t + dt, X_{t+dt})(W_{t+dt} - W_t). \quad (12)$$

The decode SDE (10) should be expressed in the reverse-Itô products because, in our setting, the decoder is conditioned on $x_T = z$ and described in the time-backward manner. For the numerical generation of new samples, we need to calculate a current event from a future event, which is only feasible with the reverse-Itô product.

The data processing inequality (4) also can be applied to the case of path probabilities (see Appendix A):

$$D_{\text{KL}}(\mu(x) \| q_\theta(x)) \leq D_{\text{KL}}(\mathbb{P}_\phi[x_{[0,T]}] \| \mathbb{Q}_\theta[x_{[0,T]}]). \quad (13)$$

Hence, if we use the same framework in VAEs, the objective function of the diffusion models is given by the KL divergence between the path probabilities. That is, we solve the following

optimization problem:

$$(\phi^*, \theta^*) = \arg \min_{\phi, \theta} D_{\text{KL}}(\mathbb{P}_\phi[x_{[0,T]}] \| \mathbb{Q}_\theta[x_{[0,T]}]). \quad (14)$$

For the practical training of the diffusion models, we represent $D_{\text{KL}}(\mathbb{P}_\phi[x_{[0,T]}] \| \mathbb{Q}_\theta[x_{[0,T]}])$ by the NNs: u_ϕ and s_θ . To this end, we need to align the time direction of the encoder $\mathbb{P}_\phi[x_{(0,T)}|x_0 = x]$ and the decoder $\mathbb{Q}_\theta[x_{[0,T]}|x_T = z]$. To reverse the time direction of the encoder, we define the conditional distribution given that $x_T = z$ by Bayes' theorem:

$$\mathbb{P}_\phi[x_{[0,T]}|x_T = z] = \frac{\mathbb{P}_\phi[x_{[0,T]}]}{p_\phi(z)}, \quad (15)$$

where $p_\phi(z)$ is the marginal distribution at the final state x_T (i.e., the latent variable z):

$$p_\phi(z) := \int \prod_{t \in [0,T]} dx_t \mathbb{P}_\phi[x_{[0,T]}] \delta(x_T - z). \quad (16)$$

The conditional probability $\mathbb{P}_\phi[x_{[0,T]}|x_T = z]$ can be evaluated by the time-reverse-encode SDE corresponding to Eq. (9) with the initial sample $X_T \sim p_\phi$:

$$dX_t = [u_\phi(t, X_t) - g(t)^2 \nabla \log \rho_\phi(t, X_t)] \tilde{\cdot} dt + g(t) \tilde{\cdot} d\bar{W}_t. \quad (17)$$

Here, $\rho_\phi(t, \cdot)$ is the solution of the Fokker-Planck equation (FPE) [27,28] with initial condition $\rho(0, \cdot) = \mu(\cdot)$:

$$\frac{\partial \rho(t, x)}{\partial t} = -\nabla \cdot [u_\phi(t, x) \rho(t, x)] + \frac{1}{2} g(t)^2 \nabla^2 \rho(t, x), \quad (18)$$

which expresses the time evolution of the probability distribution corresponding to the encode SDE (9). The derivation of Eqs. (17) and (18) is shown in Appendix C. In the context of diffusion models, $\nabla \log \rho_\phi(t, X_t)$ is known as the score function.

By using Eq. (15), the objective function in Eq. (14) is calculated as

$$D_{\text{KL}}(\mathbb{P}_\phi[x_{[0,T]}] \| \mathbb{Q}_\theta[x_{[0,T]}]) = D_{\text{KL}}(p_\phi(z) \| \pi(z)) + \mathbb{E}_{Z \sim p_\phi} [D_{\text{KL}}(\mathbb{P}_\phi[x_{[0,T]}|x_T = Z] \| \mathbb{Q}_\theta[x_{[0,T]}|x_T = Z])]. \quad (19)$$

Moreover, by using the Girsanov theorem [30] for the reverse-Itô process, the second term can be reformulated as

$$\mathbb{E}_{Z \sim p_\phi} [D_{\text{KL}}(\mathbb{P}_\phi[x_{[0,T]}|x_T = Z] \| \mathbb{Q}_\theta[x_{[0,T]}|x_T = Z])] = \frac{1}{2} \int_0^T \frac{dt}{g(t)^2} \mathbb{E}_{X_t \sim \rho_\phi(t, \cdot)} [\|u_\phi(t, X_t) - g(t)^2 \nabla \log \rho_\phi(t, X_t) - s_\theta(t, X_t)\|^2] \quad (20)$$

(see Appendix D). Therefore, our training scheme (14) is rephrased as

$$(\phi^*, \theta^*) = \arg \min_{\phi, \theta} \left\{ D_{\text{KL}}(p_\phi(z) \| \pi(z)) + \frac{1}{2} \int_0^T \frac{dt}{g(t)^2} \mathbb{E}_{X_t \sim \rho_\phi(t, \cdot)} [\|u_\phi(t, X_t) - g(t)^2 \nabla \log \rho_\phi(t, X_t) - s_\theta(t, X_t)\|^2] \right\}, \quad (21)$$

which is the main result of this work. By solving Eq. (21), we obtain the optimal parameters (ϕ^*, θ^*) in NNs. Similarly to VAE, we can generate new samples by using the parameter θ^* in the decoder: (1) Sample the latent variable z from a known prior $\pi(z)$. (2) By time-backwardly solving the decode SDE (10) with θ^* under the terminal condition $X_T = z$, we obtain a new sample x as $X_0 = x$. For the same reason as in VAEs, the parameter ϕ^* in the encoder is not directly related to the generative processes, but the training of ϕ makes the upper bound in Eq. (13) approach $D_{\text{KL}}(\mu \| q_\theta)$.

This framework inspired from VAEs lays the foundation for a construction of diffusion models. As shown in the next section, we can derive the training scheme of SBMs and SB-type models in this framework. The previous attempt to interpret diffusion models as an extension of VAEs [22] has been limited to the model that does not learn the encoder and has derived only SBMs. By contrast, our framework imposes no such restriction, which enables us to obtain the training scheme of SB-type models.

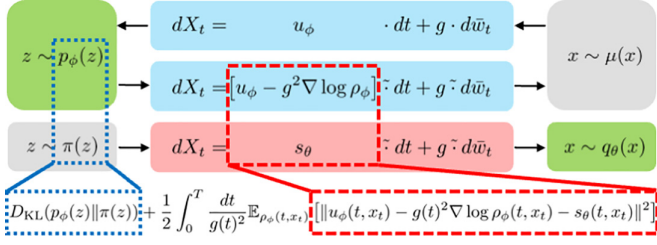


FIG. 1. The top and middle SDEs with the blue shadow denote the encode SDE with a NN u_ϕ and its time reversal, which connect the data μ and p_ϕ . By contrast, the bottom SDE with the red shadow represents the decode SDE with the other NN s_θ , which transports the prior π to q_θ . The first term in Eq. (21) means the prior loss between p_ϕ and π (see blue dotted squares); the second term indicates the drift matching between the reverse-encode and the decode SDEs (see red dotted squares).

The objective function in Eq. (21) also gives us the interpretation for the training of diffusion models as follows (see Fig. 1). The first term can be considered as the prior loss. Training the parameter ϕ in the encoder implies searching an SDE that transports the data distribution $\mu(x)$ to the prior $\pi(z)$ by minimizing the KL divergence. The second term can be interpreted as the drift matching. If we find the optimal parameter ϕ^* , we can generate new samples by solving the reverse-encode SDE (17) with ϕ^* . Nevertheless, the numerical generation of new samples is difficult since the drift term of the reverse-encode SDE includes the score function $\nabla \log \rho_\phi$. The second term exists for mimicking the drift term of the reverse-encode SDE with the other NN in the decoder, s_θ .

Owing to such an interpretation of the training scheme (21) consisting of the prior loss and drift matching, we can reduce the learning cost of the encoder. Furthermore, it is possible to mitigate the overfitting problem of SB-type models. We will see this in Sec. V with a numerical example.

IV. CONNECTION TO PREVIOUS WORK

We can rederive the previous works with new perspectives by employing the framework established in the above section. In this paper, we deal with two models and one sampling method.

A. Score-based models

In our framework, score-based models [3,13] can be understood as models that omit the training of the parameter ϕ in the encoder. To skip the training of ϕ , we need to manually find an encode SDE such that makes the prior loss in Eq. (21) smaller.

Consider an encode SDE [i.e., $u_\phi = f$ in Eq. (9)]:

$$dX_t = f(t, X_t) \cdot dt + g(t) \cdot dW_t, \quad (22)$$

where the drift $f(t, \cdot)$ is tuned as having the global stationary distribution p_{st} ; i.e., for any initial distribution, the probability

distribution ρ_t satisfies $\lim_{t \rightarrow \infty} \rho_t = p_{st}$. Then, by setting the prior $\pi = p_{st}$ and taking the time interval T in the diffusion model as the limit $T \rightarrow \infty$, we obtain $\lim_{T \rightarrow \infty} D_{KL}(\rho_T || \pi) = 0$. Therefore, if such f and g are adopted, the objective function in Eq. (21) is reduced to

$$\frac{1}{2} \int_0^\infty dt \frac{1}{g(t)^2} \mathbb{E}_{X_t \sim \rho(t, \cdot)} [\|f(t, X_t) - g(t)^2 \nabla \log \rho(t, X_t) - s_\theta(t, X_t)\|^2], \quad (23)$$

where the expectation is taken with respect to the encode SDE (22). Furthermore, by defining a new decode NN $s'_\theta := (s_\theta + f)/g^2$, Eq. (23) can be rewritten as

$$\frac{1}{2} \int_0^\infty dt g(t)^2 \mathbb{E}_{X_t \sim \rho(t, \cdot)} [\|s'_\theta(t, X_t) - \nabla \log \rho(t, X_t)\|^2]. \quad (24)$$

This objective function coincides with that of SBMs, known as *explicit* score matching [31]. In usual SBMs, the encode SDE (22) is modeled by a simple form (e.g., linear or degenerate drift), and the prior π is chosen as a normal distribution. Since this combination vanishes the prior loss in the limit $T \rightarrow \infty$, this usual setting allows us to use Eq. (23) for the training.

In practical training of SBMs, we face two technical problems: the computation of the score function $\nabla \log \rho$ and the infinite-time-horizon optimization of Eq. (24). The former is settled by using the technique called the *denoising* score matching [32] if the encode SDE (22) can be analytically solved. The latter is addressed by truncating the integral range in Eq. (24) to a finite time interval that can be feasible in a numerical computation. This means that the prior loss does not exactly vanish during practical training. Therefore, we can conclude that the training of SBMs is an approximation to reduce computational costs at the expense of accuracy of Eq. (21).

B. SB-FBSDE model

To realize the practical training without the above approximation, diffusion models based on the Schrödinger bridge have been recently developed [4–8]. SB-type models achieve high learning accuracy in a finite-time horizon by incorporating an NN into the noising (encoding) process, Eq. (22). The SB-FBSDE model [7] is one of the SB-type models and is rigorously established. Its objective function is derived by using the nonlinear Feynman-Kac theorem [17], which tells us the SDEs corresponding to the partial differential equations representing the optimal condition of the SB problem [16]. This model can also be understood within our framework as follows.

For the practical training, the objective function in Eq. (21) is not suitable because it contains the score function $\nabla \log \rho_\phi$, which is difficult to compute. By the technique in the *implicit* score matching [31], the objective function can be reformulated as

$$(\phi^*, \theta^*) = \arg \min_{\phi, \theta} \left\{ -\mathbb{E}_{Z \sim p_\phi} [\log \pi(Z)] + \frac{1}{2} \int_0^T \frac{dt}{g(t)^2} \mathbb{E}_{X_t \sim \rho_\phi(t, \cdot)} [\|u_\phi(t, X_t) - s_\theta(t, X_t)\|^2 - 2g(t)^2 \nabla \cdot s_\theta(t, X_t)] \right\}. \quad (25)$$

Its derivation is shown in Appendix E. Since this equation does not contain the score function $\nabla \log \rho_\phi$ explicitly, we can numerically solve it in the finite horizon T .

We notice that the difference between the objective functions in Eqs. (21) and (25) is just the Shannon entropy of the data distribution: $-\mathbb{E}_{X \sim \mu}[\log \mu]$, which is a constant value for a fixed dataset (see Appendix D). Therefore, the objective functions in Eqs. (21) and (25) exhibit equivalent behavior in the training process.

Similarly to the SBM case, if we replace the drift terms as $(u_\phi, s_\theta) = (f + g^2 u'_\phi, f - g^2 s'_\theta)$ by defining new encode and decode NNs, u'_ϕ and s'_θ , the reformulated objective function in Eq. (25) agrees with the optimization in the SB-FBSDE model (see also Theorem 4 in Ref. [7]):

$$(\phi^*, \theta^*) = \arg \max_{\phi, \theta} \left\{ \mathbb{E}_{Z \sim p_\phi}[\log \pi(Z)] - \int_0^T dt g(t)^2 \mathbb{E}_{X_t \sim \rho_\phi(t, \cdot)} \left[\frac{1}{2} \|u'_\phi(t, X_t)\|^2 + \frac{1}{2} \|s'_\theta(t, X_t)\|^2 \right. \right. \\ \left. \left. + \nabla \cdot \left(s'_\theta(t, X_t) - \frac{f(t, X_t)}{g(t)^2} \right) + u'_\phi(t, X_t) \cdot s'_\theta(t, X_t) \right] \right\}. \quad (26)$$

This implies that the learning process of the SB-FBSDE model is composed of the minimization of the prior loss and the drift matching. Conversely, this result also shows that our proposed training scheme, Eq. (21), implicitly solves the SB problem.

C. Probability-flow ODE

The probability-flow ODE [3,7,13] is a sampling method proposed for diffusion models, which enables us to generate new samples with ordinary differential equation (ODE) instead of SDE. This method is rederived in our framework as follows.

By using the FPE corresponding to the decode SDE (10), an ODE equivalent to the SDE (10) can be derived as

$$\frac{dX_t}{dt} = s_\theta(t, X_t) + \frac{1}{2} g(t)^2 \nabla \log \tilde{\rho}_\theta(t, X_t), \quad (27)$$

where $\tilde{\rho}_\theta$ is the marginal distribution of $\mathbb{Q}_\theta[x_{[0,T]}]$ at time t (see Appendix F). Under the optimal condition of Eq. (21), the score function is approximated by the difference of the drift terms: $g^2 \nabla \log \tilde{\rho}_\theta \simeq u_{\phi^*} - s_{\theta^*}$ (see the second term). Then, Eq. (27) can be rewritten as

$$\frac{dX_t}{dt} \simeq \frac{1}{2} [u_{\phi^*}(t, X_t) + s_{\theta^*}(t, X_t)]. \quad (28)$$

This is the probability-flow ODE for SB-type models. Unlike generation via the SDE (10), generation via the ODE requires two parameters (ϕ^*, θ^*) . If we manually set $u_\phi = f$ and shift the NN as $s_\theta = f - g^2 s'_\theta$ as in SBMs, Eq. (28) reduces to

$$\frac{dX_t}{dt} \simeq f(t, X_t) - \frac{1}{2} g(t)^2 s'_\theta(t, X_t), \quad (29)$$

which is the probability-flow ODE for SBMs.

As the ODE can be solved deterministically, it enables faster sampling than the SDE. However, Eq. (28) is an approximation as long as the objective function in Eq. (21) does not vanish. Therefore, the probability-flow ODE is more susceptible to the effects of the training loss, and the generative process is expected to be less stable.

V. NUMERICAL EXPERIMENTS

The proposed objective function in Eq. (21) enables us to separate the training of the encoder ϕ and the decoder θ . The former training of ϕ realizes good transportation from the empirical distribution μ to the prior distribution π by

using KL divergence (prior loss), while the latter training of θ mimics the transport dynamics with the SDE (10).

Since the prior distribution π is typically defined as a normal distribution, it is expected that transforming any data distribution into π is not too difficult in general. Therefore, we may stop the training of ϕ before the training of θ is completed. Moreover, by remaining the prior loss with this stopping, the overfitting problem may be mitigated. To verify these conjectures, we conduct several numerical experiments.

A. Numerical experiments in a 2-dim moon distribution

We performed the training of the encoder ϕ and the decoder θ by using the objective function in Eq. (25). Here, we adopted the 2-dim moon (see Appendix G) as the true distribution p_{true} . The prior π is set to the standard normal distribution, and the drifts u_ϕ and s_θ are parametrized with fully connected residual networks. The networks are trained with the AdamW optimizer with a learning rate of 10^{-4} and a batch size of 200. We solve the SDEs via the Euler-Maruyama method, the time horizon T is set to 1.0, and the time interval $[0, T]$ is discretized into 100 steps.

In Fig. 2, we show the objective function in Eq. (25) and the prior loss $D_{\text{KL}}(p_\phi \| \pi)$ in Eq. (21) as graphs of the number of training iterations. For calculating the prior loss, we use the kernel density estimation method and the Monte Carlo method with 200 samples. We can observe that the prior loss stabilizes more rapidly than the overall objective function. Hence, it may be allowed to interrupt the training of the encoder ϕ after the prior loss has stabilized.

In Fig. 3, we demonstrated the training with the above interruption technique for the encoder ϕ . To evaluate the performance of the training, we used a true loss $D_{\text{KL}}(p_{\text{true}} \| q_\theta)$, which is calculated from the 20 000 samples generated from p_{true} . The blue curve represents the joint training of the encoder ϕ and the decoder θ (default training), while the green dashed curve is the training of θ after interrupting the training of ϕ at iteration 1500. As shown in Fig. 3, the both curves exhibit a similar trend in the training iterations. This result shows that our interruption technique for the encoder ϕ is applicable and reduces the training cost.

The true losses decrease during the initial phase of the training but, after a certain number of iterations, begin to increase (see Fig. 3). That means the overfitting occurs. We can see that the green dashed curve is consistently lower than the blue curve after the interruption of the training of the encoder

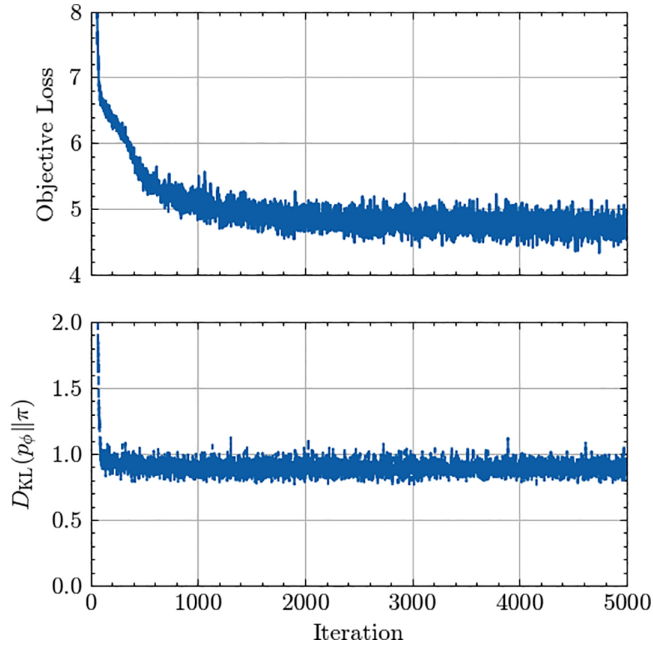


FIG. 2. The objective function in Eq. (25) (top) and the prior loss $D_{\text{KL}}(p_\phi \parallel \pi)$ in Eq. (21) (bottom) as graphs of the number of training iterations.

ϕ . This fact shows that the interruption of the training of the encoder ϕ mitigates the overfitting problem. In the following, we discuss why this mitigation occurs due to the interruption technique.

B. Overfitting in SB-type models

The overfitting is a common problem in generative models, including the diffusion models. The generative models $q_\theta(x)$ aim to reproduce the true distribution $p_{\text{true}}(x)$ that the dataset follows. In the training of generative models, the maximum likelihood estimation equation (1) is often used. If expressive

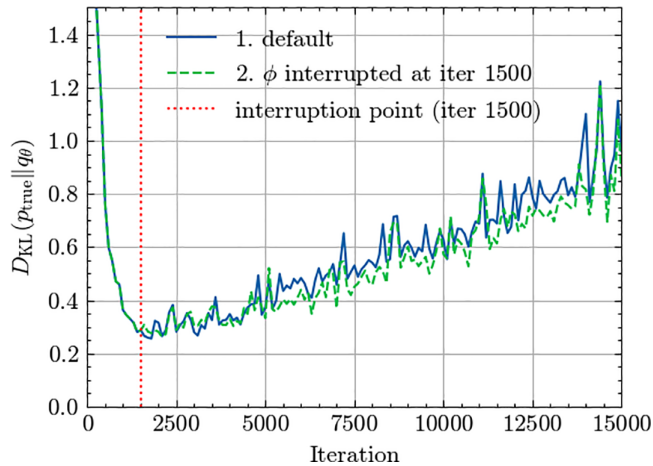


FIG. 3. The true loss $D_{\text{KL}}(p_{\text{true}} \parallel q_\theta)$ as a graph of the number of training iterations, which is computed at every 100 iterations. The blue curve represents the default training of the encoder ϕ and the decoder θ , while the green dashed curve is the training of the decoder θ after interrupting the training of the encoder ϕ at iteration 1500.

performance of the model is sufficiently high, $D_{\text{KL}}(\mu \parallel q_\theta)$ may attain zero, then the model completely reproduces the data distribution $\mu(x)$. However, when the number of training data n is finite, $\mu(x)$ is not equal to $p_{\text{true}}(x)$. Therefore, in the finite dataset case, the optimized model $q_{\theta^*}(x)$ does not coincide with $p_{\text{true}}(x)$ and is restricted to sampling only those data in the dataset $\{x_i\}_{i=1}^n$.

The simplest idea to prevent the overfitting is to stop the training of the model before it reaches the point where $D_{\text{KL}}(\mu \parallel q_\theta) = 0$. In our interruption technique, this idea can be rephrased as preventing prior loss from reaching zero. For the green dashed curve in Fig. 3, the prior loss remains nonzero by the interruption. Therefore, the green dashed curve is located below the blue curve for the default training.

For preventing the overfitting in SBMs, the time horizon T should not be too long, and it is important to set an appropriate length [18,19]. We have noted that to extend the diffusion time horizon to infinity corresponds to diminishing the prior loss. Hence, the restriction of the time horizon in SBMs is equivalent to the interruption of the training of the encoder ϕ in SB-type models.

VI. CONCLUSION

We have proposed an alternative method for deriving the objective function of SB-type diffusion models as the extension of VAEs. Our objective function (21) demonstrates that the diffusion model training scheme is structured by minimizing the prior loss and performing the drift matching. The key insight lies in extending the number of latent variables from one to infinity, leveraging the data processing inequality. This perspective enables us to interpret SB-type models within the framework of VAEs. Furthermore, we have shown that the training of the encoder can be interrupted owing to quickly stabilizing of the prior loss, which enables us to reduce the learning cost. Additionally, this interruption is expected to mitigate the overfitting problem.

Recently, the diffusion models have been applied to sampling methods in the statistical mechanics [33]. If we set the true distribution p_{true} to the Boltzmann distribution, the diffusion model can be interpreted as a sampling method for the Boltzmann distribution. It is anticipated that our proposed method can make a contribution even in such sampling schemes.

In this paper, we have modeled the encoder $\mathbb{P}_\phi[x_{[0,T]}|x_0]$ and the decoder $\mathbb{Q}_\theta[x_{[0,T]}|x_T]$ by SDEs (9) and (10), but this is just one example. Even if we extend them to other probabilistic rules (e.g., non-Markov processes), the proposed derivation for the objective function is expected to be applicable. A consequent training scheme of the objective function may be composed of two parts: One learns the encoding process that transports the data to the prior, corresponding to the prior loss, and the other trains the decoding process to mimic the reversal of the encoding process, which is the drift matching.

ACKNOWLEDGMENTS

This study was supported by the JSPS KAKENHI Grant No. 23H01432. Our study received financial support from the public/private R&D investment strategic expansion program

(PRISM) and programs for bridging the gap between R&D and ideal society (Society 5.0) and generating economic and social value (BRIDGE) from the Cabinet Office.

DATA AVAILABILITY

The data that support the findings of this article are not publicly available upon publication because it is not technically feasible and/or the cost of preparing, depositing, and hosting the data would be prohibitive within the terms of this research project. The data are available from the authors upon reasonable request.

APPENDIX A: DATA PROCESSING INEQUALITY

We derive the data processing inequality. Let $x^N := \{x_i\}_{i=1}^N \in \mathcal{X}^N$, and consider the two joint probability distributions $P_N(x^N)$ and $Q_N(x^N)$, and the KL divergence between them:

$$D_{\text{KL}}(P_N(x^N) \| Q_N(x^N)) = \mathbb{E}_{x^N \sim P_N} \left[\log \frac{P_N(x^N)}{Q_N(x^N)} \right]. \quad (\text{A1})$$

The joint distributions can be written as the product of conditional and marginal distributions:

$$P_N(x^N) = P_N(x_N | x^{N-1}) P_{N-1}(x^{N-1}), \quad (\text{A2})$$

$$Q_N(x^N) = Q_N(x_N | x^{N-1}) Q_{N-1}(x^{N-1}). \quad (\text{A3})$$

The substitution of Eqs. (A2) and (A3) into Eq. (A1) leads to

$$\begin{aligned} D_{\text{KL}}(P_N(x^N) \| Q_N(x^N)) &= D_{\text{KL}}(P_{N-1}(x^{N-1}) \| Q_{N-1}(x^{N-1})) \\ &\quad + \mathbb{E}_{x^{N-1} \sim P_{N-1}} [D_{\text{KL}}(P_N(x_N | x^{N-1}) \| Q_N(x_N | x^{N-1}))]. \end{aligned} \quad (\text{A4})$$

Since the second term of the right-hand side (RHS) is non-negative, we obtain the data processing inequality:

$$D_{\text{KL}}(P(x^N) \| Q(x^N)) \geq D_{\text{KL}}(P(x^{N-1}) \| Q(x^{N-1})). \quad (\text{A5})$$

If we set $N = 2$, this reduces to Eq. (4). Furthermore, by repeating the above procedure, we reach

$$D_{\text{KL}}(P(x^N) \| Q(x^N)) \geq D_{\text{KL}}(P(x_1) \| Q(x_1)), \quad (\text{A6})$$

which corresponds to Eq. (13) in the limit $N \rightarrow \infty$.

APPENDIX B: THE OBJECTIVE FUNCTION FOR VAEs

Equation (7) is derived as follows. By using the definitions of the encoder [Eq. ((5)) and the decoder [Eq. ((3)), we can calculate the KL divergence as

$$\begin{aligned} D_{\text{KL}}(P_\phi(x, z) \| Q_\theta(x, z)) &= \mathbb{E}_{x, z \sim P_\phi} \left[\log \frac{P_\phi(x, z)}{Q_\theta(x, z)} \right] \\ &= \mathbb{E}_{x \sim \mu, Z \sim P_\phi(\cdot | x)} \left[\log \frac{P_\phi(Z | x) \mu(x)}{Q_\theta(X | Z) \pi(Z)} \right] \\ &= \mathbb{E}_{x \sim \mu, Z \sim P_\phi(\cdot | x)} \left[\log \mu(x) \right. \\ &\quad \left. + \log \frac{P_\phi(Z | x)}{\pi(Z)} - \log Q_\theta(X | Z) \right]. \end{aligned} \quad (\text{B1})$$

Hence, the objective function can be decomposed as

$$\begin{aligned} D_{\text{KL}}(P_\phi(x, z) \| Q_\theta(x, z)) &= \mathbb{E}_{x \sim \mu} [\log \mu(x)] + \mathbb{E}_{x \sim \mu} [D_{\text{KL}}(P_\phi(z | x) \| \pi(z))] \\ &\quad - \mathbb{E}_{x \sim \mu, Z \sim P_\phi(\cdot | x)} [\log Q_\theta(X | Z)]. \end{aligned} \quad (\text{B2})$$

APPENDIX C: DERIVATION OF ONSAGER-MACHLUP ACTION AND TIME-REVERSE SDE

Before deriving Eqs. (17) and (18), we show how to calculate the conditional path probabilities from the Itô and reverse-Itô SDEs.

First, we consider the Itô SDE:

$$dX_t = f(t, X_t) \cdot dt + g(t) \cdot dW_t, \quad (\text{C1})$$

where W_t is the standard Wiener process. For an infinitesimal time interval $[t, t + \Delta t]$, Eq. (C1) can be discretized as

$$X_{t+\Delta t} - X_t = f(t, X_t) \Delta t + g(t) \Delta W_t. \quad (\text{C2})$$

Here, $\Delta W_t := W_{t+\Delta t} - W_t$, and it is sampled from the normal distribution:

$$\mathcal{N}(\Delta w_t; 0, \Delta t) = \frac{1}{\sqrt{2\pi \Delta t}} \exp \left(-\frac{\|\Delta w_t\|^2}{2\Delta t} \right). \quad (\text{C3})$$

Therefore, the transition probability density from $X_t = x_t$ to $X_{t+\Delta t} = x_{t+\Delta t}$ is computed as

$$\begin{aligned} T(x_{t+\Delta t} | x_t) &= \frac{1}{\sqrt{2\pi g(t)^2 \Delta t}} \exp \left[-\frac{\Delta t}{2g(t)^2} \left\| \frac{x_{t+\Delta t} - x_t}{\Delta t} - f(t, x_t) \right\|^2 \right]. \end{aligned} \quad (\text{C4})$$

Also, the conditional path probability $\mathbb{P}[x_{(0,T)} | x_0]$ is written as

$$\begin{aligned} \mathbb{P}[x_{(0,T)} | x_0] &= \prod_{t \in (0,T]} dx_t \\ &= \lim_{N \rightarrow \infty} \prod_{i=0}^{N-1} T(x_{i+1} | x_i) dx_{i+1} \\ &= \lim_{N \rightarrow \infty} \prod_{i=0}^{N-1} \frac{dx_{i+1}}{\sqrt{2\pi g(t_i)^2 \Delta t}} \\ &\quad \times \exp \left[-\Delta t \sum_{i=0}^{N-1} \frac{1}{2g(t_i)^2} \left\| \frac{x_{i+1} - x_i}{\Delta t} - f(t_i, x_i) \right\|^2 \right]. \end{aligned} \quad (\text{C5})$$

Here, $x_i = x_{t_i}$, $t_i = i\Delta t$, and $\Delta t = T/N$. Then, by formally writing $\dot{x}_t dt := x_{t+dt} - x_t$, $\mathcal{D}x_{(0,T)} := \prod_{t \in [0,T]} dx_t / \sqrt{2\pi g(t)^2 dt}$, we can express Eq. (C5) as the path integral formulation:

$$\mathbb{P}[x_{(0,T)} | x_0] = \prod_{t \in (0,T]} dx_t = \mathcal{D}x_{(0,T)} e^{-\int_0^T dt \mathcal{L}(t, x_t, \dot{x}_t)}, \quad (\text{C6})$$

$$\mathcal{L}(t, x_t, \dot{x}_t) = \frac{1}{2g(t)^2} \|\dot{x}_t - f(t, x_t)\|^2. \quad (\text{C7})$$

The function $\mathcal{L}(t, x_t, \dot{x}_t)$ is known as the Onsager-Machlup Lagrangian [28,29].

Second, we calculate the conditional path probability $\tilde{\mathbb{P}}[x_{[0,T)}|x_T]$ generated from the reverse-Itô SDE:

$$dX_t = f(t, X_t) \tilde{\cdot} dt + g(t) \tilde{\cdot} dW_t. \quad (\text{C8})$$

This can be discretized as

$$X_{t+\Delta t} - X_t = f(t + \Delta t, X_{t+\Delta t})\Delta t + g(t + \Delta t)\Delta W_t. \quad (\text{C9})$$

Corresponding to Eq. (C4), the transition probability density $\tilde{T}(x_t|x_{t+\Delta t})$ from $X_{t+\Delta t} = x_{t+\Delta t}$ to $X_t = x_t$ is computed as

$$\tilde{T}(x_t|x_{t+\Delta t}) = \frac{1}{\sqrt{2\pi g(t + \Delta t)^2 \Delta t}} \exp \left[-\frac{\Delta t}{2g(t + \Delta t)^2} \left\| \frac{x_{t+\Delta t} - x_t}{\Delta t} - f(t + \Delta t, x_{t+\Delta t}) \right\|^2 \right]. \quad (\text{C10})$$

Hence, the conditional path probability $\tilde{\mathbb{P}}[x_{[0,T)}|x_T]$ is written as

$$\begin{aligned} \tilde{\mathbb{P}}[x_{[0,T)}|x_T] \prod_{t \in [0,T)} dx_t &= \lim_{N \rightarrow \infty} \prod_{i=0}^{N-1} \tilde{T}(x_i|x_{i+1}) dx_i \\ &= \lim_{N \rightarrow \infty} \prod_{i=0}^{N-1} \frac{dx_i}{\sqrt{2\pi g(t_{i+1})^2 \Delta t}} \exp \left[-\sum_{i=0}^{N-1} \frac{\Delta t}{2g(t_{i+1})^2} \left\| \frac{x_{i+1} - x_i}{\Delta t} - f(t_{i+1}, x_{i+1}) \right\|^2 \right]. \end{aligned} \quad (\text{C11})$$

By using a Taylor expansion for f and g with respect to time t and ignoring the terms with irrelevant order of Δt , Eq. (C11) is further analyzed as follows. The infinitesimal transition probability $\tilde{T}(x_t|x_{t+\Delta t})$ is calculated as

$$\begin{aligned} \tilde{T}(x_t|x_{t+\Delta t}) &\approx \frac{1}{\sqrt{2\pi g(t)^2 \Delta t}} \exp \left[-\frac{\Delta t}{2g(t)^2} \left\| \frac{x_{t+\Delta t} - x_t}{\Delta t} - f(t, x_t) - \partial_t f(t, x_t) \Delta t \right. \right. \\ &\quad \left. \left. - \nabla \cdot f(t, x_t)(x_{t+\Delta t} - x_t)^2 - \frac{1}{2} g(t)^2 \nabla^2 f(t, x_t) \Delta t \right\|^2 \right] \\ &= \frac{1}{\sqrt{2\pi g(t)^2 \Delta t}} \exp \left[-\frac{\Delta t}{2g(t)^2} \left\| \frac{x_{t+\Delta t} - x_t}{\Delta t} - f(t, x_t) \right\|^2 + \nabla \cdot f(t, x_t) \Delta t \right]. \end{aligned} \quad (\text{C12})$$

Here, we used the Itô formula [27] for the expansion of f and g . One may be concerned that the Itô formula cannot be applied to this Taylor expansion since $x_{t+\Delta t}$ is not a stochastic variable. However, a realization $x_{[0,T]}$ that does not satisfy the SDE (C8) never occurs. Thus, we can formally use the Itô formula.

By plugging Eq. (C12) into Eq. (C11), the conditional path probability $\tilde{\mathbb{P}}[x_{[0,T)}|x_T]$ is also written as

$$\begin{aligned} \tilde{\mathbb{P}}[x_{[0,T)}|x_T] \prod_{t \in [0,T)} dx_t &= \lim_{N \rightarrow \infty} \prod_{i=0}^{N-1} \tilde{T}(x_i|x_{i+1}) \\ &= \lim_{N \rightarrow \infty} \prod_{i=0}^{N-1} \frac{dx_i}{\sqrt{2\pi g(t_i)^2 \Delta t}} \exp \left[-\Delta t \sum_{i=0}^{N-1} \left\{ \frac{1}{2g(t_i)^2} \left\| \frac{x_{i+1} - x_i}{\Delta t} - f(t_i, x_i) \right\|^2 - \nabla \cdot f(t_i, x_i) \right\} \right]. \end{aligned} \quad (\text{C13})$$

In the path integral formulation, we find

$$\tilde{\mathbb{P}}[x_{[0,T)}|x_T] \prod_{t \in [0,T)} dx_t = \mathcal{D}x_{[0,T)} e^{-\int_0^T dt \tilde{\mathcal{L}}(t, x_t, \dot{x}_t)}, \quad (\text{C14})$$

$$\tilde{\mathcal{L}}(t, x_t, \dot{x}_t) = \frac{1}{2g(t)^2} \|\dot{x}_t - f(t, x_t)\|^2 - \nabla \cdot f(t, x_t). \quad (\text{C15})$$

By using the above two path probabilities, Eqs. (C6) and (C14), we derive Eqs. (17) and (18) as follows. The numerator in the RHS of Eq. (15), $\mathbb{P}_\phi[x_{[0,T]}]$, can be represented by the following path integral formulation:

$$\mathbb{P}_\phi[x_{[0,T]}] \prod_{t \in [0,T]} dx_t = \mathcal{D}x_{[0,T]} \mu(x_0) e^{-\int_0^T dt \mathcal{L}_\phi(t, x_t, \dot{x}_t)}, \quad (\text{C16})$$

where $\mathcal{L}_\phi(t, x_t, \dot{x}_t)$ is the Onsager-Machlup Lagrangian:

$$\mathcal{L}_\phi(t, x_t, \dot{x}_t) = \frac{1}{2g(t)^2} \|\dot{x}_t - u_\phi(t, x_t)\|^2. \quad (\text{C17})$$

By substituting Eq. (C16) into Eq. (15), we have

$$\begin{aligned} \mathbb{P}_\phi[x_{[0,T)}|x_T] \prod_{t \in [0,T)} dx_t &= \mathcal{D}x_{[0,T)} \frac{\mu(x_0)}{p_\phi(x_T)} e^{-\int_0^T dt \mathcal{L}_\phi(t, x_t, \dot{x}_t)} \\ &= \mathcal{D}x_{[0,T)} e^{-\int_0^T dt \mathcal{L}'_\phi(t, x_t, \dot{x}_t)}, \end{aligned} \quad (\text{C18})$$

where $\mathcal{L}'_\phi(t, x_t, \dot{x}_t)$ is defined as

$$\mathcal{L}'_\phi(t, x_t, \dot{x}_t) = \mathcal{L}_\phi(t, x_t, \dot{x}_t) + \frac{d}{dt} \log \rho_\phi(t, x_t). \quad (\text{C19})$$

Furthermore, the second term can be expressed as

$$\begin{aligned} \frac{d}{dt} \log \rho_\phi(t, x_t) &= \frac{\partial}{\partial t} \log \rho_\phi(t, x_t) + \nabla \log \rho_\phi(t, x_t) \cdot \dot{x}_t \\ &\quad + \frac{1}{2} g(t)^2 \nabla^2 \log \rho_\phi(t, x_t), \end{aligned} \quad (\text{C20})$$

where we use the Itô formula. Taking into account that $\rho_\phi(t, \cdot)$ is given by the solution of the FPE (18), $\mathcal{L}'(t, x_t, \dot{x}_t)$ can be calculated as

$$\begin{aligned} \mathcal{L}'_\phi(t, x_t, \dot{x}_t) &= \frac{1}{2g(t)^2} \|\dot{x}_t - (u_\phi(t, x_t) - g(t)^2 \nabla \log \rho_\phi(t, x_t))\|^2 \\ &\quad - \nabla \cdot (u_\phi(t, x_t) - g(t)^2 \nabla \log \rho_\phi(t, x_t)). \end{aligned} \quad (\text{C21})$$

By comparing this form with Eq. (C14), we find that the conditional path probability $\mathbb{P}_\phi[x_{[0,T]}|x_T = z]$ is generated by the reverse-Itô SDE with drift term $u_\phi(t, x_t) - g(t)^2 \nabla \log \rho_\phi(t, x_t)$, that is, Eq. (17).

APPENDIX D: GIRSANOV THEOREM FOR REVERSE-ITÔ SDES AND THE OBJECTIVE FUNCTION FOR THE DIFFUSION MODEL

The left-hand side of Eq. (20) can be represented as

$$\begin{aligned} \mathbb{E}_{Z \sim \rho_\phi} [D_{\text{KL}}(\mathbb{P}_\phi[x_{[0,T]}|x_T = Z] \parallel \mathbb{Q}_\theta[x_{[0,T]}|x_T = Z])] &= \mathbb{E}_{X_{[0,T]} \sim \mathbb{P}_\phi} \left[\log \frac{\mathbb{P}_\phi[X_{[0,T]}|X_T]}{\mathbb{Q}_\theta[X_{[0,T]}|X_T]} \right]. \end{aligned} \quad (\text{D1})$$

By using the discretized representation of the conditional path probability [Eq. (C11)], we obtain

$$\begin{aligned} \log \frac{\mathbb{P}_\phi[x_{[0,T]}|x_T]}{\mathbb{Q}_\theta[x_{[0,T]}|x_T]} &= \lim_{N \rightarrow \infty} - \sum_{i=0}^{N-1} \frac{\Delta t}{2g(t_{i+1})^2} \left[\left\| \frac{x_{i+1} - x_i}{\Delta t} - \tilde{u}_\phi(t_{i+1}, x_{i+1}) \right\|^2 \right. \\ &\quad \left. - \left\| \frac{x_{i+1} - x_i}{\Delta t} - s_\theta(t_{i+1}, x_{i+1}) \right\|^2 \right], \end{aligned} \quad (\text{D2})$$

where we denote $\tilde{u}_\phi := u_\phi - g^2 \nabla \log \rho_\phi$ for notational simplicity. The expansion of the square terms gives

$$\begin{aligned} \left\| \frac{x_{i+1} - x_i}{\Delta t} - \tilde{u}_\phi(t_{i+1}, x_{i+1}) \right\|^2 - \left\| \frac{x_{i+1} - x_i}{\Delta t} - s_\theta(t_{i+1}, x_{i+1}) \right\|^2 &= -2 \frac{x_{i+1} - x_i}{\Delta t} \{ \tilde{u}_\phi(t_{i+1}, x_{i+1}) - s_\theta(t_{i+1}, x_{i+1}) \} \\ &\quad + \{ \|\tilde{u}_\phi(t_{i+1}, x_{i+1})\|^2 - \|s_\theta(t_{i+1}, x_{i+1})\|^2 \}. \end{aligned} \quad (\text{D3})$$

Since the realizations $x_{[0,T]}$ generated from the reverse SDE (17) only contribute the expectation in Eq. (D1), $x_{i+1} - x_i$ can be evaluated as

$$x_{i+1} - x_i = \tilde{u}_\phi(t_{i+1}, x_{i+1}) \Delta t + g(t_{i+1}) \Delta \bar{w}_i. \quad (\text{D4})$$

The substitution of the above equation into Eq. (D3) gives

$$\begin{aligned} \left\| \frac{x_{i+1} - x_i}{\Delta t} - \tilde{u}_\phi(t_{i+1}, x_{i+1}) \right\|^2 - \left\| \frac{x_{i+1} - x_i}{\Delta t} - s_\theta(t_{i+1}, x_{i+1}) \right\|^2 &= -2 \frac{\Delta \bar{w}_i}{\Delta t} g(t_{i+1}) \{ \tilde{u}_\phi(t_{i+1}, x_{i+1}) - s_\theta(t_{i+1}, x_{i+1}) \} \\ &\quad - \|\tilde{u}_\phi(t_{i+1}, x_{i+1}) - s_\theta(t_{i+1}, x_{i+1})\|^2. \end{aligned} \quad (\text{D5})$$

Hence, we obtain

$$\begin{aligned} \log \frac{\mathbb{P}_\phi[x_{[0,T]}|x_T]}{\mathbb{Q}_\theta[x_{[0,T]}|x_T]} &= \lim_{N \rightarrow \infty} \sum_{i=0}^{N-1} \left[\frac{\tilde{u}_\phi(t_{i+1}, x_{i+1}) - s_\theta(t_{i+1}, x_{i+1})}{g(t_{i+1})} \Delta \bar{w}_i \right. \\ &\quad \left. + \frac{\|\tilde{u}_\phi(t_{i+1}, x_{i+1}) - s_\theta(t_{i+1}, x_{i+1})\|^2}{2g(t_{i+1})^2} \Delta t \right] \\ &= \int_0^T \frac{\tilde{u}_\phi(t, x_t) - s_\theta(t, x_t)}{g(t)} \cdot d\bar{w}_t \\ &\quad + \frac{1}{2} \int_0^T \frac{\|\tilde{u}_\phi(t, x_t) - s_\theta(t, x_t)\|_2^2}{g(t)^2} \cdot dt. \end{aligned} \quad (\text{D6})$$

This is the Girsanov theorem [30] for the reverse-Itô SDE. If we take expectation over $\mathbb{P}_\phi[x_{[0,T]}]$ for the both sides of Eq. (D6), the first term vanishes and we get

$$\begin{aligned} \mathbb{E}_{X_{[0,T]} \sim \mathbb{P}_\phi} \left[\log \frac{\mathbb{P}_\phi[X_{[0,T]}|X_T]}{\mathbb{Q}_\theta[X_{[0,T]}|X_T]} \right] &= \mathbb{E}_{X_{[0,T]} \sim \mathbb{P}_\phi} \left[\frac{1}{2} \int_0^T \frac{\|\tilde{u}_\phi(t, X_t) - s_\theta(t, X_t)\|^2}{g(t)^2} \cdot dt \right] \\ &= \frac{1}{2} \int_0^T \frac{dt}{g(t)^2} \mathbb{E}_{X_t \sim \rho_\phi(t, \cdot)} [\|\tilde{u}_\phi(t, X_t) - s_\theta(t, X_t)\|^2]. \end{aligned} \quad (\text{D7})$$

Here, the expectation in the last line is taken over the distribution $\rho_\phi(t, x_t)$ at each time step t , while the expectation in the second line is over the path probability $\mathbb{P}_\phi[x_{[0,T]}]$. The substitution of Eq. (D7) into Eq. (D1) leads to Eq. (20).

APPENDIX E: REFORMATION OF THE OBJECTIVE FUNCTION WITH IMPLICIT SCORE MATCHING

The first term of the objective function in Eq. (21) can be decomposed as

$$D_{\text{KL}}(p_\phi(z) \parallel \pi(z)) = \mathbb{E}_{Z \sim p_\phi} [\log p_\phi(Z)] - \mathbb{E}_{Z \sim p_\phi} [\log \pi(Z)], \quad (\text{E1})$$

where $p_\phi(z)$ is the marginal distribution of $\mathbb{P}_\phi[x_{[0,T]}]$ at $t = T$. The first term can be rewritten as

$$\begin{aligned} \mathbb{E}_{Z \sim p_\phi} [\log p_\phi(Z)] &= \mathbb{E}_{X \sim \mu} [\log \mu(X)] + \int_0^T dt \frac{d}{dt} \mathbb{E}_{X \sim \rho_\phi(t, \cdot)} [\log \rho_\phi(t, X)]. \end{aligned} \quad (\text{E2})$$

By calculating the integrand in the second term in the last line, we have

$$\begin{aligned} & \frac{d}{dt} \mathbb{E}_{X \sim \rho_\phi(t, \cdot)} [\log \rho_\phi(t, X)] \\ &= \frac{d}{dt} \int dx \rho_\phi(t, x) \log \rho_\phi(t, x) \\ &= \int dx \frac{\partial \rho_\phi(t, x)}{\partial t} \log \rho_\phi(t, x) + \frac{d}{dt} \int dx \rho_\phi(t, x). \end{aligned} \quad (\text{E3})$$

The second term of the RHS vanishes because the probability distribution $\rho_\phi(t, x)$ is normalized for any t . Moreover, since ρ_ϕ is the solution of the encode FPE (18), the substitution of the FPE (18) into the first term of Eq. (E3) gives

$$\begin{aligned} & \frac{d}{dt} \mathbb{E}_{X \sim \rho_\phi(t, \cdot)} [\log \rho_\phi(t, X)] \\ &= - \int dx \nabla \cdot \left[\left(u_\phi - \frac{1}{2} g^2 \nabla \log \rho_\phi \right) \rho_\phi \right] \log \rho_\phi \\ &= \int dx \rho_\phi \left(u_\phi - \frac{1}{2} g^2 \nabla \log \rho_\phi \right) \cdot \nabla \log \rho_\phi \\ &= \frac{1}{2g^2} \mathbb{E}_{\rho_\phi} [2u_\phi \cdot (g^2 \nabla \log \rho_\phi) - \|g^2 \nabla \log \rho_\phi\|^2]. \end{aligned} \quad (\text{E4})$$

Here, we used the integration by parts and ignored the surface term to get the third line. Therefore, we obtain

$$\begin{aligned} D_{\text{KL}}(p_\phi \| \pi) &= \mathbb{E}_\mu [\log \mu] - \mathbb{E}_{p_\phi} [\log \pi] + \frac{1}{2} \int_0^T \frac{dt}{g^2} \mathbb{E}_{\rho_\phi} \\ &\quad \times [2u_\phi \cdot (g^2 \nabla \log \rho_\phi) - \|g^2 \nabla \log \rho_\phi\|^2]. \end{aligned} \quad (\text{E5})$$

The second term of the objective function in Eq. (21) can be expanded as

$$\begin{aligned} & \frac{1}{2} \int_0^T \frac{dt}{g^2} \mathbb{E}_{\rho_\phi} [\|u_\phi - g^2 \nabla \log \rho_\phi - s_\theta\|^2] \\ &= \frac{1}{2} \int_0^T \frac{dt}{g^2} \mathbb{E}_{\rho_\phi} [\|u_\phi - s_\theta\|^2 + \|g^2 \nabla \log \rho_\phi\|^2 \\ &\quad - 2(u_\phi - s_\theta) \cdot (g^2 \nabla \log \rho_\phi)]. \end{aligned} \quad (\text{E6})$$

By combining Eqs. (E5) and (E6), the objective function in Eq. (21) can be reformed as

$$\begin{aligned} & D_{\text{KL}}(\mu(x) \| q_\theta(x)) \\ &= \mathbb{E}_\mu [\log \mu] - \mathbb{E}_{p_\phi} [\log \pi] + \frac{1}{2} \int_0^T \frac{dt}{g^2} \mathbb{E}_{\rho_\phi} \\ &\quad \times [\|u_\phi - s_\theta\|^2 + 2s_\theta \cdot (g^2 \nabla \log \rho_\phi)]. \end{aligned} \quad (\text{E7})$$

Because the first term $\mathbb{E}_\mu [\log \mu]$ does not contain the parameters (ϕ, θ) , we can ignore it in the optimization problem. Furthermore, for any probability distribution $p(x)$ with the condition $\lim_{\|x\| \rightarrow \infty} p(x) = 0$ and test function $f(x)$, the fol-

lowing equality holds:

$$\mathbb{E}_{X \sim p} [\nabla \log p(X) \cdot f(X)] = -\mathbb{E}_{X \sim p} [\nabla \cdot f(X)]. \quad (\text{E8})$$

This leads to Eq. (25) as

$$\begin{aligned} & D_{\text{KL}}(p_\phi \| \pi) + \frac{1}{2} \int_0^T \frac{dt}{g^2} \mathbb{E}_{\rho_\phi} [\|\tilde{u}_\phi - s_\theta\|^2] \\ &= -\mathbb{E}_{p_\phi} [\log \pi] + \frac{1}{2} \int_0^T \frac{dt}{g^2} \mathbb{E}_{\rho_\phi} [\|u_\phi - s_\theta\|^2 - 2g^2 \nabla \cdot s_\theta]. \end{aligned} \quad (\text{E9})$$

The reformulation from the original objective function in Eq. (21) to Eq. (25) is called implicit score matching [31].

APPENDIX F: PROBABILITY-FLOW ODE

An FPE corresponding to the decode SDE (10) is given by

$$\begin{aligned} & \frac{\partial \tilde{\rho}(t, x)}{\partial t} \\ &= -\nabla \cdot [s_\theta(t, x) \tilde{\rho}(t, x)] - \frac{1}{2} g(t)^2 \nabla^2 \tilde{\rho}(t, x) \\ &= -\nabla \cdot \left[\left\{ s_\theta(t, x) + \frac{1}{2} g(t)^2 \nabla \log \tilde{\rho}(t, x) \right\} \tilde{\rho}(t, x) \right]. \end{aligned} \quad (\text{F1})$$

By focusing the last line, we notice that Eq. (F1) is also derived from the ODE with the vector field $s_\theta + 2^{-1} g^2 \nabla \log \tilde{\rho}$:

$$\frac{dX_t}{dt} = s_\theta(t, X_t) + \frac{1}{2} g(t)^2 \nabla \log \tilde{\rho}_\theta(t, X_t). \quad (\text{F2})$$

Here, $\tilde{\rho}_\theta(t, x)$ is the solution of the FPE (F1) with the terminal condition $\tilde{\rho}_\theta(T, \cdot) = \pi(\cdot)$, that is, Eq. (27).

If the optimization in Eq. (21) is sufficiently achieved, the path probability in the decoding process can be approximated by the one of the encoding process: $\mathbb{Q}_{\theta^*}[x_{[0,T]}] \simeq \mathbb{P}_{\phi^*}[x_{[0,T]}]$. Therefore, for the marginal distribution, $\tilde{\rho}_{\theta^*} \simeq \rho_{\phi^*}$ holds. Furthermore, owing to the drift matching in Eq. (21), we obtain $g^2 \nabla \log \rho_{\phi^*} \simeq s_{\theta^*} - u_{\phi^*}$. Thus, under the optimal condition in Eq. (21), the above ODE (F2) can be rewritten as

$$\begin{aligned} & \frac{dX_t}{dt} \simeq u_{\phi^*}(t, X_t) + \frac{1}{2} \nabla \log \rho_{\phi^*}(t, X_t) \\ &\simeq u_{\phi^*}(t, X_t) + \frac{1}{2} [u_{\phi^*}(t, X_t) - s_{\theta^*}(t, X_t)] \\ &= \frac{1}{2} [u_{\phi^*}(t, X_t) + s_{\theta^*}(t, X_t)], \end{aligned} \quad (\text{F3})$$

which is called the probability-flow ODE [7].

APPENDIX G: MOON DISTRIBUTION

The moon distribution is a well-known toy dataset used in machine learning, particularly for testing clustering and

ALGORITHM 1. Generate a batch of samples on the moon distribution.

```

Function Sample( $n$ ):
  if  $n$  is None then
     $n \leftarrow \text{batch\_size}$ 
   $m \leftarrow \lfloor n/2 \rfloor$ 
   $x \leftarrow \text{linspace}(0, \pi, m)$  // Divide  $[0, \pi]$  into  $m$ 
    points
   $u \leftarrow [(\cos(x_i) + 0.5, -\sin(x_i) + 0.2) \mid i =$ 
     $1, \dots, m] \times 10$ 
   $u \leftarrow u + 0.5 \mathcal{N}(0, 1)$  // Add Gaussian noise
   $v \leftarrow [(\cos(x_i) - 0.5, \sin(x_i) - 0.2) \mid i = 1, \dots, m] \times 10$ 
   $v \leftarrow v + 0.5 \mathcal{N}(0, 1)$ 
  return Tensor(concatenate( $u, v$ ))

```

classification algorithms. It consists of two interleaving half circles, which can be challenging for linear classifiers to separate.

The algorithm to generate samples from the moon distribution is in Algorithm 1. The samples are generated by creating two half circles with some Gaussian noise added to them. The generated samples are shown in Fig. 4.

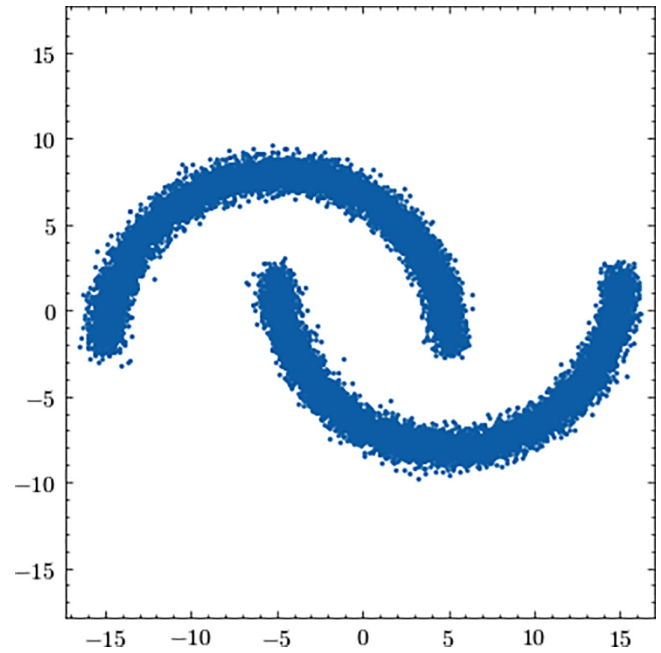


FIG. 4. Scatter of 20 000 samples on the moon distribution generated by Algorithm 1.

-
- [1] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, Deep unsupervised learning using nonequilibrium thermodynamics, in *Proceedings of the International Conference on Machine Learning* (PMLR, 2015), pp. 2256–2265.
 - [2] J. Ho, A. Jain, and P. Abbeel, Denoising diffusion probabilistic models, *Adv. Neural Inf. Proc. Syst.* **33**, 6840 (2020).
 - [3] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, *Score-Based Generative Modeling Through Stochastic Differential Equations* (ICLR, 2021).
 - [4] G. Wang, Y. Jiao, Q. Xu, Y. Wang, and C. Yang, Deep generative learning via Schrödinger bridge, in *Proceedings of the 38th International Conference on Machine Learning* (PMLR, 2021), Vol. 139, pp. 10794–10804.
 - [5] F. Vargas, P. Thodoroff, A. Lamacraft, and N. Lawrence, Solving Schrödinger bridges via maximum likelihood, *Entropy* **23**, 1134 (2021).
 - [6] V. De Bortoli, J. Thornton, J. Heng, and A. Doucet, Diffusion Schrödinger bridge with applications to score-based generative modeling, *Adv. Neural Inf. Proc. Syst.* **34**, 17695 (2021).
 - [7] T. Chen, G.-H. Liu, and E. A. Theodorou, Likelihood training of Schrödinger bridge using forward-backward SDEs theory, in *International Conference on Learning Representations* (2022).
 - [8] A. Tong, K. FATRAS, N. Malkin, G. Huguët, Y. Zhang, J. Rector-Brooks, G. Wolf, and Y. Bengio, Improving and generalizing flow-based generative models with minibatch optimal transport, *Trans. Mach. Learn. Res.* (2024).
 - [9] P. Dhariwal and A. Nichol, Diffusion models beat GANs on image synthesis, *Adv. Neural Inf. Proc. Syst.* **34**, 8780 (2021).
 - [10] G. Batzolis, J. Stanczuk, C.-B. Schönlieb, and C. Etmann, Conditional image generation with score-based diffusion models, [arXiv:2111.13606](https://arxiv.org/abs/2111.13606).
 - [11] N. Chen, Y. Zhang, H. Zen, R. J. Weiss, M. Norouzi, and W. Chan, WaveGrad: Estimating gradients for waveform generation, [arXiv:2009.00713](https://arxiv.org/abs/2009.00713).
 - [12] V. Popov, I. Vovk, V. Gogoryan, T. Sadekova, and M. Kudinov, Grad-TTS: A diffusion probabilistic model for text-to-speech, in *International Conference on Machine Learning* (PMLR, 2021), pp. 8599–8608.
 - [13] Y. Song, C. Durkan, I. Murray, and S. Ermon, Maximum likelihood training of score-based diffusion models, *Adv. Neural Inf. Proc. Syst.* **34**, 1415 (2021).
 - [14] H. Cao, C. Tan, Z. Gao, Y. Xu, G. Chen, P.-A. Heng, and S. Z. Li, A survey on generative diffusion models, *IEEE Trans. Knowl. Data Eng.* **36**, 2814 (2024).
 - [15] E. Schrödinger, Sur la théorie relativiste de l'électron et l'interprétation de la mécanique quantique, *Ann. Inst. Henri Poincaré* **2**, 269 (1932).
 - [16] K. F. Caluya and A. Halder, Wasserstein proximal algorithms for the Schrödinger bridge problem: Density control with nonlinear drift, *IEEE Trans. Autom. Control* **67**, 1163 (2021).
 - [17] I. Exarchos and E. A. Theodorou, Stochastic optimal control via forward and backward stochastic differential equations and importance sampling, *Automatica* **87**, 159 (2018).
 - [18] P. Li, Z. Li, H. Zhang, and J. Bian, On the generalization properties of diffusion models, *Adv. Neural Inf. Proc. Syst.* **36**, 2097 (2023).
 - [19] Z. Kadhodaie, F. Guth, E. P. Simoncelli, and S. Mallat, Generalization in diffusion models arises from geometry-adaptive harmonic representation, in *The Twelfth International Conference on Learning Representations* (2024).

- [20] D. P. Kingma and M. Welling, Auto-encoding variational Bayes, in *Advances in Neural Information Processing Systems* (2014).
- [21] D. P. Kingma and M. Welling, An Introduction to Variational Autoencoders, *Foundations and Trends in Machine Learning* **12**, 307 (2019).
- [22] C. Luo, Understanding diffusion models: A unified perspective, [arXiv:2208.11970](https://arxiv.org/abs/2208.11970).
- [23] T. M. Mitchell, *Machine Learning* (McGraw-Hill, 1997).
- [24] We often choose that $\dim \mathcal{Z} \leq \dim \mathcal{X}$ and the prior is a normal distribution.
- [25] This inequality usually describes the relationships among mutual informations, but in this paper, we apply it for the KL divergences.
- [26] In diffusion models, $\dim \mathcal{Z} = \dim \mathcal{X}$.
- [27] C. Gardiner, *Stochastic Methods: A Handbook for the Natural and Social Sciences*, Springer Series in Synergetics, 4th ed. (Springer, Berlin, 2009), Vol. 13.
- [28] H. H. Risken, *The Fokker-Planck Equation: Methods of Solution and Applications*, Springer Series in Synergetics (Springer, 1996), 2nd ed., Vol. 18.
- [29] Y. Hirono, A. Tanaka, and K. Fukushima, Understanding diffusion models by Feynman's path integral, in *Proceedings of the 41st International Conference on Machine Learning* (PMLR, 2024), Vol. 235, pp. 18324–18351.
- [30] B. Oksendal, *Stochastic Differential Equations: An Introduction with Applications* (Springer Science & Business Media, 2013).
- [31] A. Hyvärinen and P. Dayan, Estimation of non-normalized statistical models by score matching, *J. Mach. Learn. Res.* **6**, 695 (2005).
- [32] P. Vincent, A connection between score matching and denoising autoencoders, *Neural Comput.* **23**, 1661 (2011).
- [33] Y. Wang, L. Guo, H. Wu, and T. Zhou, Energy based diffusion generator for efficient sampling of Boltzmann distributions, [arXiv:2401.02080](https://arxiv.org/abs/2401.02080).