The Physics of Diffusion Generative Models

Yung-Kyun Noh[1,3] and Daniel D. Lee[2,3]

[1]Dept. of Computer Science, Hanyang University, 222 Wangsimni-ro, Seongdong-gu, Seoul 04763, Rep. of Korea.

[2]Dept. of Electrical and Computer Engineering, Cornell Tech, 2 W Loop Rd, New York, NY 10044, USA.

[3]Center for AI and Natural Sciences, Korea Institute for Advanced Study, 85 Hoegi-ro, Dongdaemun-gu, Seoul 02455, Rep. of Korea.

Contributing authors: nohyung@hanyang.ac.kr; ddl46@cornell.edu;

Abstract

Diffusion generative models have been intensively employed for data generation, yielding ground-breaking results. This paper elucidates the physics underlying the algorithms of these models and aims to promote a thorough understanding of the diffusive mechanism for information processing.

Keywords: Diffusion models, Generative models, Physics-based models, Fluid dynamics

1 Introduction

The core concept of the diffusion generative models rests on the understanding that a random walk produces a diffusive flow. The flow is characterized *not* by fluctuations of diffusive substance but by coherent movements that broadly move molecules from regions of high density to those of low density. In diffusion generative models, data are considered as diffusive molecules, and their flow patterns are memorized by neural networks. The neural networks are then subsequently used to guide the reverse flow that reproduces the original data-generating probability density out of random data.

Due to the difference in density, random walks produce a flow as in Fig. 1. Although the individual movement is random without coherence, data in overall flows from left to right. The amount of flow at each point is expressed as vector field J, and the average movement is expressed as velocity field $\mathbf{v}$. Reverse process for reproducing

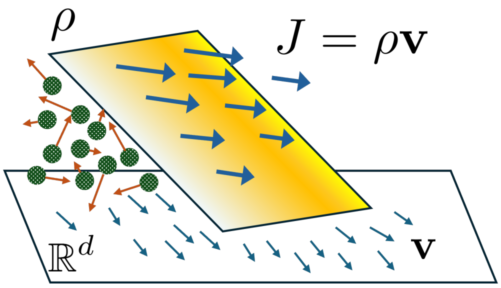

Fig. 1: Random walk of data and diffusive flow velocity $\mathbf{v}$. While $\mathbf{v}$ represents the overall velocity of flow, the actual amount of data movement is $J = \rho\mathbf{v}$ with current data density ρ.

the original density is performed by the change of $\mathbf{v}$, which is the advection control of the velocity field. For example, if a diffusion process makes a $\mathbf{v}(\mathbf{x}, t)$ at positions $\mathbf{x} \in \mathbb{R}^d$ at current time t, an advection control $-\mathbf{v}(\mathbf{x}, t)$ on top of the diffusion process makes the density not to make any diffusive flow. If a diffusion made a time-dependent flow $\mathbf{v}(\mathbf{x}, t)$ for $0 \leq t \leq T$, then a continued diffusion with an advective control $\mathbf{v}(\mathbf{x}, t') = -2\mathbf{v}(\mathbf{x}, T - t')$ for $T \leq t' \leq 2T$ will reconstruct the original density at $t' = 2T$.

In this paper, a physics system that performs this diffusion and reverse advective control is illustrated.

2 Overview of Diffusion and Advection

A flux J is a vector field that represents both the amount of flow and its direction. The flux caused by diffusion is proportional to the slope of density ρ with proportional constant D:

$$J = -D\nabla\rho \ . \tag{1}$$

The constant D depends on the diffusion properties such as the variance of diffusion.

The flux, in turn, induces a corresponding change in the density. Consider a flux illustrated in Fig. 2, with components for each dimension, $J = [J_1, \ldots, J_d]^\top$. For the i-th component, J_i, the rate of mass accumulation within a differential element dx_i along the x_i-coordinate is determined by the change in J_i. Within a small volume of thickness dx_i along the x_i-axis, the change of J_i quantifies the net difference between the inflow and outflow. This difference between inflow and outflow, accounted for by the conservation of mass, remains within dx_i.

We let the amount of flow be the velocity weighted by mass, $J = \rho\mathbf{v}$. Then the rate of change of J_i along the x_i-coordinate characterizes the rate of density change. Summing these rates across all directions constitutes yields the total rate of density

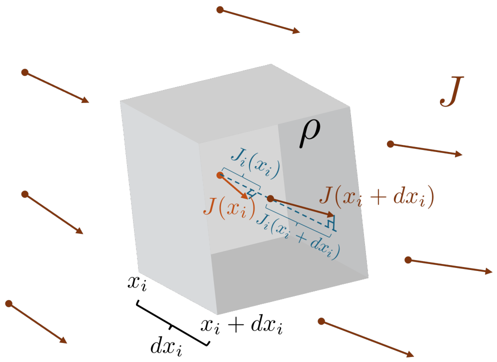

Fig. 2: Change of flux field with an infinitesimal change of a coordinate value. The infinitesimal spatial change of field $\frac{dJ_i}{dx_i}$ in each direction causes the change of mass density $\frac{d\rho}{dt}$.

change:

$$\frac{d\rho}{dt} = -\sum_{i=1}^{d} \frac{dJ_i}{dx_i} = -\nabla \cdot J. \tag{2}$$

Upon substituting J in Eq. (2) with Eq. (1), we derive the equation for density change due to diffusion:

$$\frac{d\rho}{dt} = -\nabla \cdot (-D\nabla\rho) = D\nabla^2\rho \ . \tag{3}$$

The change in density, solely caused by diffusion, depends on the sum of the second derivatives across all directions.

With the flow amount expressed as $J = \rho\mathbf{v}$, and referring to Eq. (1), the velocity field resulting from the diffusion process is

$$\mathbf{v} = \frac{J}{\rho} = -D\frac{\nabla\rho}{\rho} = -D\nabla\log\rho, \tag{4}$$

where $\nabla\log\rho$ is known as the score vector field in machine learning. Here, the gradient is with respect to the data position $\mathbf{x} \in \mathbb{R}^d$. Control over data flow can be exerted by introducing an additional velocity field. If a control $\mathbf{u}_c$ is applied to the diffusion process, it modifies the velocity $\mathbf{v}$ to $\mathbf{u}_c + \mathbf{v}$. The net flux modified by control linearly combines the contributions from both control and diffusion:

$$J_c = \rho\mathbf{u}_c - D\nabla\rho. \tag{5}$$

Here, the first term $\rho\mathbf{u}_c$ represents the flux from control, and the second term $-D\nabla\rho$ represents the flux due to pure diffusion.

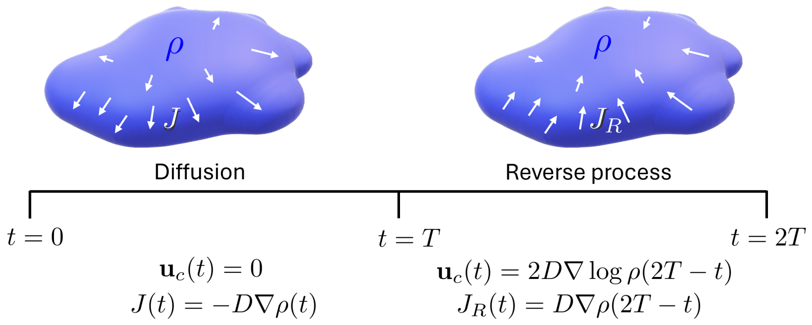

Fig. 3: Diffusion and its reverse process. Starting from $t = 0$, sample points from the original density function diffuse until $t = T$, and the diffusive velocity $\mathbf{v}(t) = -D\nabla \log \rho(t)$ observed from data is memorized. For the time t' greater than T, we begin to apply the control $\mathbf{u}_c(t') = -2\mathbf{v}(t = 2T - t')$ using the memorized $\mathbf{v}(t)$ during $0 \leq t \leq T$. At $t' = 2T$, the diffused density returns to the original density $\rho(2T) = \rho(0)$ after the reverse process from T to $2T$ under this control.

For example, if we want to eliminate the net flow, we can apply a control field $\mathbf{u}_c = D\nabla \log \rho$, which is equal to the opposite of the diffusive velocity field $\mathbf{v}$, on top of the diffusion. The resulting net flow is

$$\begin{aligned} J_c &= \rho D\nabla \log \rho - D\nabla\rho \qquad (6) \\ &= \rho D\frac{\nabla\rho}{\rho} - D\nabla\rho = 0. \qquad (7) \end{aligned}$$

Consequently, the system experiences no net flow, thus preserving the current density ρ. Note that although the net flow vanishes, data points continue to exhibit diverse individual movements.

Finally, a time-reverse process can be achieved using $\mathbf{u}_c = 2D\nabla \log \rho$, which is twice the aforementioned density-preserving control, $D\nabla \log \rho$. This adjustment results in the following reversed flux:

$$\begin{aligned} J_R &= 2\rho D\nabla \log \rho - D\nabla\rho \qquad (8) \\ &= D\nabla\rho, \qquad (9) \end{aligned}$$

which is the opposite flux to the forward diffusive flux $J = -D\nabla\rho$.

A diffusion generative model can be understood through the physics that combines the diffusion process with a subsequent reverse process to reproduce the original density. We allow the samples to diffuse from $t = 0$ to $t = T$, as illustrated in Fig. 3, during which time we memorize the velocity field $\mathbf{v}(t)$. For the time t' greater than T, we incorporate a control $\mathbf{u}_c(t') = -2\mathbf{v}(t = 2T - t')$ using the memorized $\mathbf{v}(t)$ at the reflected time $t = 2T - t'$. Then, the resulting flux $J(t')$ due to diffusion is reversed after $t' = T$.

3 Review: Diffusion Generative Models

Diffusion generative models follow the forward and reverse processes described in the previous section. The models corrupt data by adding Gaussian noise. The diffusive velocity is memorized, and this memorized velocity field is later used to control the reversed flow, reconstructing the original density. In this section, we review the forward and reverse processes used in diffusion generative models.

3.1 Forward process

The forward process uses the following recursive equation for the update of $\mathbf{x}_t$:

$$\mathbf{x}_t = \sqrt{1 - \beta_t}\mathbf{x}_{t-1} + \sqrt{\sigma^2 \beta_t}\, \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, I). \tag{10}$$

Here, $\mathbf{x}_t, \mathbf{x}_{t+1}$, and ϵ_t belong to $\mathbb{R}^d$, β is in the interval $(0, 1)$, and I represents the $d \times d$ identity matrix. We define $q(\mathbf{x}_0)$ as the data-generating density, and $q(\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_T)$ as the joint density function for all forward-generated $\mathbf{x}_1, \ldots, \mathbf{x}_T$ along with the original data $\mathbf{x}_0$. According to the forward process, $q(\mathbf{x}_1, \ldots, \mathbf{x}_T|\mathbf{x}_0)$ is jointly Gaussian for a given sample $\mathbf{x}_0$. Because $\mathbf{x}_1, \ldots, \mathbf{x}_T$ are jointly Gaussian for a given $\mathbf{x}_0$, its marginalized density $q(\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_k}|\mathbf{x}_0)$ for any subset $\{i_1, \ldots, i_k\} \subseteq \{1, \ldots, T\}$, and conditional density $q(\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_k}|\mathbf{x}_0, \mathbf{x}_{j_1}, \ldots, \mathbf{x}_{j_l})$ for any disjoint subsets $\{i_1, \ldots, i_k\}, \{j_1, \ldots, j_l\} \subseteq \{1, \ldots, T\}$ are also Gaussians irrespective of the subset sizes $0 \leq k, l \leq T$. Note that $q(\mathbf{x}_0)$, the data-generating function, is not necessarily assumed to be Gaussian. Finally, because the forward process is a Markov process, the following condition is also satisfied: $q(\mathbf{x}_i, \mathbf{x}_j|\mathbf{x}_l) = q(\mathbf{x}_i|\mathbf{x}_l)q(\mathbf{x}_j|\mathbf{x}_l)$ if $0 \leq j < l < i \leq T$. Marginalized density $q(\mathbf{x}_t)$ corresponds to the mass density $\rho(t)$ in the previous section.

In summary, the following three conditions can be employed to develop analytical methods using diffusion processes.

- $q(\mathbf{x}_1, \ldots, \mathbf{x}_T|\mathbf{x}_0)$ is jointly Gaussian.
- $q(\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_k}|\mathbf{x}_0, \mathbf{x}_{j_1}, \ldots, \mathbf{x}_{j_l})$ is jointly Gaussian for $\{i_1, \ldots, i_k\}, \{j_1, \ldots, j_l\} \subseteq \{1, \ldots, T\}$ and $\{i_1, \ldots, i_k\} \cap \{j_1, \ldots, j_l\} = \emptyset$ with $0 \leq k, l \leq T$.
- $q(\mathbf{x}_i, \mathbf{x}_j|\mathbf{x}_l) = q(\mathbf{x}_i|\mathbf{x}_l)q(\mathbf{x}_j|\mathbf{x}_l)$ for $0 \leq i < l < j \leq T$.

The β_t parameter is used to balance between the sample and noise. Adding consecutive noise causes the variance to blow up; however, the introduction of β_t allows the asymptotic density become a Gaussian with finite variance.

The variance parameter σ^2 determines the variance of the asymptotic density. Instead of using σ^2 in the update equation, we can have an equivalent result by using $\epsilon_t \sim \mathcal{N}(0, \sigma^2 I)$ replacing the noise with variance 1.

3.1.1 Diffused density with large T

With the forward update outlined in Eq. (10), the original density $q(\mathbf{x}_0)$ gradually transforms into a Gaussian. In terms of the covariance of the data, it consistently becomes $\sigma^2 I$ regardless of the initial covariance of $q(\mathbf{x}_0)$. Let the covariance of $q(\mathbf{x}_0)$ is Σ_0 regardless of its shape, and let the covariance at step t be Σ_t. The covariance Σ_t is a linear combination of the Σ_0 and the covariance of noise, $\sigma^2 I$. At each step, the weight of Σ_0 is decomposed into the weights of Σ_0 and $\sigma^2 I$ for the subsequent step, at rates $1 - \beta_t$ and β_t, respectively, as follows:

$$\begin{aligned}
\Sigma_1 &= (1-\beta_1)\Sigma_0 + \beta_1 \sigma^2 I & (11)\\
\Sigma_2 &= \prod_{i=1}^{2}(1-\beta_i)\Sigma_0 \; + \; \underbrace{((1-\beta_1)\beta_2 + \beta_1)}_{=1-\prod_{i=1}^{2}(1-\beta_i)}\sigma^2 I & (12)\\
&\vdots \\
\Sigma_t &= \underbrace{\prod_{i=1}^{t}(1-\beta_i)}_{\Sigma_0\text{-coefficient}}\Sigma_0 + \underbrace{\left(1 - \prod_{i=1}^{t}(1-\beta_i)\right)}_{\sigma^2 I\text{-coefficient}}\sigma^2 I. & (13)
\end{aligned}$$

The allocation of weights during forward process is illustrated in Fig. 4. At $t = 1$, the coefficient is β_1 for Σ_0 and $1 - \beta_1$ for $\sigma^2 I$, Next, β_2 portion of the $\sigma^2 I$ coefficient, $\beta_2(1-\beta_1)$ turned into the coefficient of Σ_0, and the remaining $1-\beta_2$ portion, $(1-\beta_2)(1-\beta_1)$ became the coefficient of $\sigma^2 I$. At step t, $\prod_{i=1}^{t}(1-\beta_i)$ is the coefficient of $\sigma^2 I$, and we can notice that all the accumulated weights $\beta_1+\beta_2(1-\beta_1)+\cdots+\beta_t(1-\beta_{t-1})\cdots(1-\beta_1)$ constitutes $1 - \prod_{i=1}^{t}(1-\beta_t)$.

The coefficients in the update equation Eq. (10) can be considered as components of a hypersphere. In Fig. 5(a), the vertical axis represents the coefficient for $\mathbf{x}_0$, and the horizontal axis represents the coefficient for ϵ_1. The balance between $\sqrt{1-\beta_1}$ vs. $\sqrt{\beta_1}$ in constructing $\mathbf{x}_1$ is depicted as a point on the surface of the circle in the first orthant. Considering $\mathbf{x}_2$, the coefficients for $\mathbf{x}_0$, ϵ_1, and ϵ_2 are $\prod_{i=1}^{2}\sqrt{1-\beta_i}$, $\sqrt{\beta_1}\sqrt{1-\beta_2}$, and $\sqrt{\beta_2}$, respectively. These are also depicted as a point on the surface of the 3-dimensional sphere in Fig. 5(b). If we combine the noise $\epsilon_1, \ldots, \epsilon_t$ into a single Gaussian noise $\epsilon_{\text{tot}} \sim \mathcal{N}(0, I)$, the coefficients for $\mathbf{x}_t$ is depicted in Fig. 5(c), and the

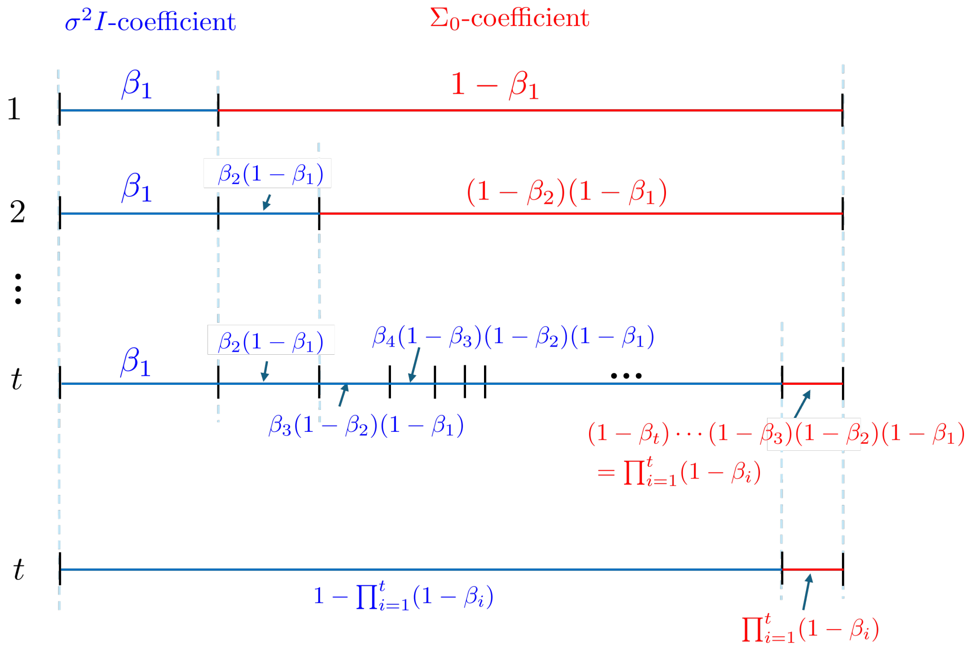
$\sigma^2 I$-coefficient
Σ_0-coefficient
1
β_1
$1-\beta_1$
2
β_1
$\beta_2(1-\beta_1)$
$(1-\beta_2)(1-\beta_1)$
⋮
t
β_1
$\beta_2(1-\beta_1)$
$\beta_4(1-\beta_3)(1-\beta_2)(1-\beta_1)$
$\cdots$
$\beta_3(1-\beta_2)(1-\beta_1)$
$(1-\beta_t)\cdots(1-\beta_3)(1-\beta_2)(1-\beta_1)$
$=\prod_{i=1}^{t}(1-\beta_i)$
t
$1-\prod_{i=1}^{t}(1-\beta_i)$
$\prod_{i=1}^{t}(1-\beta_i)$

Fig. 4: Diagram of covariance coefficients. The update in Eq. (10) redistribute the coefficient of Σ_0, transferring β_t rate of it to the coefficient of $\sigma^2 I$. The remaining coefficient for $\sigma^2 I$ is $\prod_{i=1}^{t}(1-\beta_t)$, while the coefficient for Σ_0 is $1-\prod_{i=1}^{t}(1-\beta_t)$.

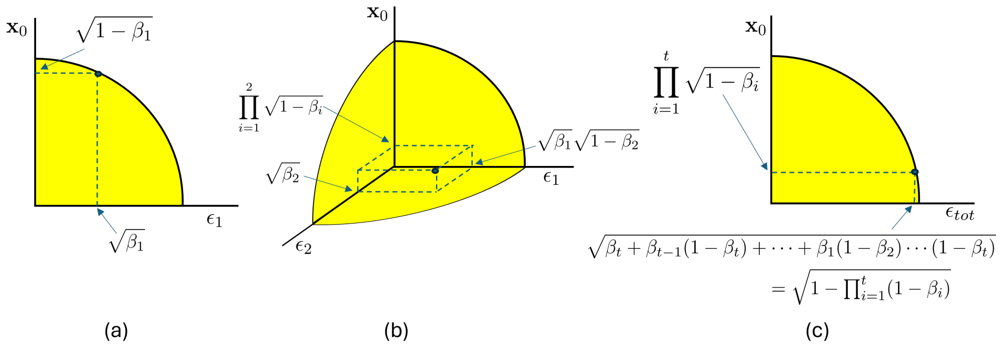
$\mathbf{x}_0$
$\sqrt{1-\beta_1}$
ϵ_1
$\sqrt{\beta_1}$
(a)
$\mathbf{x}_0$
$\prod_{i=1}^{2}\sqrt{1-\beta_i}$
$\sqrt{\beta_1}\sqrt{1-\beta_2}$
$\sqrt{\beta_2}$
ϵ_1
ϵ_2
(b)
$\mathbf{x}_0$
$\prod_{i=1}^{t}\sqrt{1-\beta_i}$
ϵ_{tot}
$\sqrt{\beta_t+\beta_{t-1}(1-\beta_t)+\cdots+\beta_1(1-\beta_2)\cdots(1-\beta_t)}$
$=\sqrt{1-\prod_{i=1}^{t}(1-\beta_i)}$
(c)

Fig. 5: Coefficient representation on the surface of a sphere. (a) The Coefficient representation for $\mathbf{x}_1$ consists of $\sqrt{1-\beta_1}$ for $\mathbf{x}_0$ and $\sqrt{\beta_1}$ for ϵ_1. (b) The coefficient representation for $\mathbf{x}_2$ is expressed on the surface of a three-dimensional sphere. (c) The coefficient representation for $\mathbf{x}_t$ is depicted on the surface of a circle when $\epsilon_1,\ldots,\epsilon_t$ are combined into ϵ_{tot}.

update equation can be represented as[1]

$$\mathbf{x}_t = \prod_{i=1}^{t} \sqrt{1-\beta_i}\ \mathbf{x}_0 + \sum_{i=1}^{t} \frac{\sqrt{\sigma^2\beta_i}}{\sqrt{1-\beta_i}} \prod_{j=i}^{t} \sqrt{1-\beta_j}\ \epsilon_i \tag{15}$$

$$= \prod_{i=1}^{t} \sqrt{1-\beta_i}\ \mathbf{x}_0 + \sqrt{\sigma^2}\sqrt{1-\prod_{i=1}^{t}(1-\beta_i)}\ \epsilon_{\text{tot}}, \tag{16}$$

with a substitution

$$\epsilon_{\text{tot}} = \frac{1}{\sqrt{1-\prod_{i=1}^{t}(1-\beta_i)}} \sum_{i=1}^{t} \frac{\sqrt{\beta_i}}{\sqrt{1-\beta_i}} \prod_{j=i}^{t} \sqrt{1-\beta_j}\ \epsilon_i. \tag{17}$$

Note that a single noise ϵ_{tot} is used instead of t number of Gaussian noises $\epsilon_1, \dots, \epsilon_t$ with $\epsilon_{\text{tot}} \sim \mathcal{N}(0, I)$ because

$$\sum_{i=1}^{t} \frac{\beta_i}{1-\beta_i} \prod_{j=i}^{t}(1-\beta_j) + \prod_{i=1}^{t}(1-\beta_i) = 1. \tag{18}$$

As we increase t, the coefficient for $\mathbf{x}_0$ in Eq. (16) and the coefficient for Σ_0 in Eq. (13) exponentially decreases. For large enough T, the $\mathbf{x}_0$ contribution to $\mathbf{x}_T$ vanishes, and the resulting marginal density $q(\mathbf{x}_T) = \mathcal{N}(0, \sigma^2 I)$ retains only pure noise for any $q(\mathbf{x}_0)$.

3.1.2 Formulations for forward processes

The joint density for the forward-generated Gaussian $q(\mathbf{x}_1, \dots, \mathbf{x}_T|\mathbf{x}_0)$ can be represented using a mean vector and a covariance matrix for $\mathbf{x}_1, \dots, \mathbf{x}_T$. Let the t-th element of the mean vector be $\mu_{t|0}$ and the t, t'-th element of the covariance matrix be $\Sigma_{t,t'|0}$. According to Eq. (16), the mean for the variable $\mathbf{x}_t$ can be calculated as

$$\mu_{t|0} = \mathbb{E}[\mathbf{x}_t|\mathbf{x}_0] = \prod_{i=1}^{t} \sqrt{1-\beta_i}\ \mathbf{x}_0, \tag{19}$$

the variance calculation using $\epsilon_{\text{tot}} \sim \mathcal{N}(0, I)$ as

$$\sigma^2_{t|0} = \mathbb{E}[\mathbf{x}_t^2|\mathbf{x}_0] - \mathbb{E}[\mathbf{x}_t|\mathbf{x}_0]^2 = \sigma^2\left(1-\prod_{i=1}^{t}(1-\beta_i)\right) I, \tag{20}$$

[1] Eq. (15) can be simplified using the following:

$$\mathbf{x}_t = \prod_{i=1}^{t} \sqrt{1-\beta_i}\ \mathbf{x}_0 + \sum_{i=1}^{t} \sqrt{\sigma^2\beta_i} \prod_{j=i+1}^{t} \sqrt{1-\beta_j}\ \epsilon_i, \tag{14}$$

if we use the abused definition $\prod_{j=t+1}^{t} \sqrt{1-\beta_j} = 1$.

and the covariance between $\mathbf{x}_t$ and $\mathbf{x}_{t-l}$ for $0 < t < T$ and $0 < l < t$ as

$$\sigma_{t,t-l|0} = \sigma^2 \left(\prod_{i=t-l+1}^{t} \sqrt{1-\beta_i} \right) \left(1 - \prod_{i=1}^{t-l} (1-\beta_i) \right) I. \tag{21}$$

For two random variables $\mathbf{x}_t$ and $\mathbf{x}_{t-l}$, the density function can be written as

$$q(\mathbf{x}_t, \mathbf{x}_{t-l}|\mathbf{x}_0) = \tag{22}$$

$$\mathcal{N}\left(\begin{pmatrix} \sqrt{\bar{\alpha}_t}\mathbf{x}_0 \\ \sqrt{\bar{\alpha}_{t-l}}\mathbf{x}_0 \end{pmatrix}, \begin{pmatrix} \sigma^2(1-\bar{\alpha}_t)I & \sigma^2\sqrt{\frac{\bar{\alpha}_t}{\bar{\alpha}_{t-l}}}(1-\bar{\alpha}_{t-l})I \\ \sigma^2\sqrt{\frac{\bar{\alpha}_t}{\bar{\alpha}_{t-l}}}(1-\bar{\alpha}_{t-l})I & \sigma^2(1-\bar{\alpha}_{t-l})I \end{pmatrix} \right), \tag{23}$$

where $\bar{\alpha}_t = \prod_{i=1}^{t}(1-\beta_i)$ is used as a substitution. This equation can be sraightforwardly extended to the joint density for any subset of variables given $\mathbf{x}_0$.

3.2 Reverse process

The forward process in Eq. (10) ensures that $q(\mathbf{x}_T)$ approaches a Gaussian $\mathcal{N}(0, \sigma^2 I)$ as T becomes large for any initial density $q(\mathbf{x}_0)$. This formulation can be understood as consecutive convolutions of density using a Gaussian kernel. Then the reverse process is a deconvolution of the density through the movement of mass. The movement can be achieved in various ways. Any movement that successfully retraces the marginal density $q(\mathbf{x}_t)$ at each time point t experienced during diffusion is relevant.

3.2.1 Reverse process with expected mean

The method presented in [1] involves calculating $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ from the diffusive system. When we are given $\mathbf{x}_t$ in the reverse process, the density for the previous sample point $\mathbf{x}_{t-1}$ is a Gaussian mixture, which is not Gaussian, represented as follows:

$$\underbrace{q(\mathbf{x}_{t-1}|\mathbf{x}_t)}_{\text{Gaussian mixture}} = \int \underbrace{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)}_{\text{Gaussian.}} q(\mathbf{x}_0|\mathbf{x}_t) d\mathbf{x}_0. \tag{24}$$

Although $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is non-Gaussian, $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ is Gaussian because it is inferred from the Gaussian $q(\mathbf{x}_1, \ldots, \mathbf{x}_T|\mathbf{x}_0)$ that has been obtained during diffusion processes.

From Eq. (22), we can obtain the conditional density[2] $q(\mathbf{x}_{t-l}|\mathbf{x}_t, \mathbf{x}_0)$ for arbitrary l less than t:

$$\begin{aligned} q(\mathbf{x}_{t-l}|\mathbf{x}_t, \mathbf{x}_0) &= \mathcal{N}\left(\mu_{t-l|t,0},\ \Sigma_{t-l|t,0}\right) && (26) \\ \mu_{t-l|t,0} &= \frac{1}{1-\bar{\alpha}_t}\left[\sqrt{\bar{\alpha}_{t-l}}\left(1-\frac{\bar{\alpha}_t}{\bar{\alpha}_{t-l}}\right)\mathbf{x}_0 + \sqrt{\frac{\bar{\alpha}_t}{\bar{\alpha}_{t-l}}}\left(1-\bar{\alpha}_{t-l}\right)\mathbf{x}_t\right] && (27) \\ \Sigma_{t-l|t,0} &= \frac{\left(1-\frac{\bar{\alpha}_t}{\bar{\alpha}_{t-l}}\right)\left(1-\bar{\alpha}_{t-l}\right)\sigma^2}{1-\bar{\alpha}_t} I, && (28) \end{aligned}$$

with $\bar{\alpha}_t = \prod_{i=1}^{t}(1-\beta_i)$. The Gaussian equation for $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ with $l = 1$ is

$$\begin{aligned} q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) &= \mathcal{N}\left(\mu_{t-1|t,0},\ \Sigma_{t-1|t,0}\right) && (29) \\ \mu_{t-1|t,0} &= \frac{1}{1-\bar{\alpha}_t}\left[\beta_t\sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0 + \sqrt{1-\beta_t}\left(1-\bar{\alpha}_{t-1}\right)\mathbf{x}_t\right] && (30) \\ \Sigma_{t-1|t,0} &= \frac{\beta_t\left(1-\bar{\alpha}_{t-1}\right)\sigma^2}{1-\bar{\alpha}_t} I. && (31) \end{aligned}$$

Note that the equation does not depend on the original density $q(\mathbf{x}_0)$.

The Gaussian mixture obtained in Eq. (24), using the Gaussian described in Eq. (29), represents one possible solution that can be used for reverse processes. In the reverse process at time $t' > T$, sampling using $q(\mathbf{x}_{t'+1}|\mathbf{x}_{t'}) = q(\mathbf{x}_{2T-t'-1} = \mathbf{x}_{t'+1}|\mathbf{x}_{2T-t'} = \mathbf{x}_{t'})$ retraces the marginalized density $q(\mathbf{x}_{t'}) = q(\mathbf{x}_{2T-t'})$, which appeared throughout the history of diffusion. Here, both $t = 2T - t' - 1$ and $t = 2T - t'$ are less than T, and $q(\mathbf{x}_{2T-t'-1} = \mathbf{x}_{t'+1}|\mathbf{x}_{2T-t'} = \mathbf{x}_{t'})$ means the value of memorized density $q(\mathbf{x}_{2T-t'-1}|\mathbf{x}_{2T-t'})$ when $\mathbf{x}_{2T-t'-1} = \mathbf{x}_{t'+1}$ and $\mathbf{x}_{2T-t'} = \mathbf{x}_{t'}$.

However, to obtain $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ from Eq. (29), we still need $q(\mathbf{x}_0|\mathbf{x}_t)$ for integration. Unfortunately, we don't have any analytical knowledge about $q(\mathbf{x}_0|\mathbf{x}_t)$. Instead, a neural network $f_\theta(\mathbf{x}_t, t) \in \mathbb{R}^d$ can be employed to learn $\mathbb{E}_{\mathbf{x}_{t-1}}[q(\mathbf{x}_{t-1}|\mathbf{x}_t)]$ with parameter θ. If the neural network $f_\theta(\mathbf{x}_t, t)$ is intended to provide the appropriate mean $\mu_{\mathbf{x}_{t-1}|\mathbf{x}_t} = \mathbb{E}_{\mathbf{x}_{t-1}}[q(\mathbf{x}_{t-1}|\mathbf{x}_t)]$ by minimizing

$$\begin{aligned} L_t &= \mathbb{E}_{\mathbf{x}_t}\left[\left(f_\theta(\mathbf{x}_t, t) - \mu_{\mathbf{x}_{t-1}|\mathbf{x}_t}\right)^2\right] && (32) \\ &= \mathbb{E}_{\mathbf{x}_t}\left[\left(f_\theta(\mathbf{x}_t, t) - \mathbb{E}_{\mathbf{x}_{t-1},\mathbf{x}_0}\left[q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)|\mathbf{x}_t\right]\right)^2\right] && (33) \end{aligned}$$

[2] We use analytical solution for conditional density of Gaussian. For a joint Gaussian

$$p(\mathbf{x}_a, \mathbf{x}_b) =$$
$$\frac{1}{\sqrt{2\pi}^d\left|\begin{pmatrix}\Sigma_a & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_b\end{pmatrix}\right|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\left(\begin{pmatrix}\mathbf{x}_a \\ \mathbf{x}_b\end{pmatrix} - \begin{pmatrix}\mu_a \\ \mu_b\end{pmatrix}\right)^\top \begin{pmatrix}\Sigma_a & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_b\end{pmatrix}^{-1}\left(\begin{pmatrix}\mathbf{x}_a \\ \mathbf{x}_b\end{pmatrix} - \begin{pmatrix}\mu_a \\ \mu_b\end{pmatrix}\right)\right),$$

with $\mathbf{x}_a \in \mathbb{R}^{d_a}$ and $\mathbf{x}_b \in \mathbb{R}^{d_b}$ for $d = d_a + d_b$, conditional density $p(\mathbf{x}_a|\mathbf{x}_b)$ is

$$p(\mathbf{x}_a|\mathbf{x}_b) = \frac{1}{\sqrt{2\pi}^{d_a}|\Sigma_{a|b}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{x}_a - \mu_{a|b})^\top \Sigma_{a|b}^{-1}(\mathbf{x}_a - \mu_{a|b})\right), \qquad (25)$$

with parameters $\mu_{a|b} = \mu_a + \Sigma_{ab}\Sigma_b^{-1}(\mathbf{x}_b - \mu_b)$ and $\Sigma_{a|b} = \Sigma_a - \Sigma_{ab}\Sigma_b^{-1}\Sigma_{ba}$.

$$= \mathbb{E}_{\mathbf{x}_t, \mathbf{x}_{t-1}, \mathbf{x}_0} \left[\left(f_\theta(\mathbf{x}_t, t) - \mathbf{x}_{t-1} \right)^2 \right], \tag{34}$$

where $\mathbb{E}_{\mathbf{x}_0} \left[q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) | \mathbf{x}_t \right] = \int q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) q(\mathbf{x}_0 | \mathbf{x}_t) d\mathbf{x}_0$ is used from Eq. (32) to Eq. (33). We have samples from forward diffusion process, and we can learn a neural network $f_\theta(\mathbf{x}_t)$ to produce the mean of $\mathbf{x}_{t-1}$ that the sample went through during forward diffusion process by minimizing

$$\widehat{L}_t = \frac{1}{N} \sum_{i=1}^{N} \left(f_\theta(\mathbf{x}_{t,i}) - \mathbf{x}_{t-1,i} \right)^2, \tag{35}$$

where $\mathbf{x}_{t-1,i}$ and $\mathbf{x}_{t,i}$ are the points diffused from the i-th sample $\mathbf{x}_{0,i}$.

Note that fitting each individual $\mathbf{x}_{t-1}$ is not of our interest; rather, we want to predict the average of previous points from which $\mathbf{x}_t$ is diffused. Eqs. (34) and (35) can be further modified using Eq. (29):

$$\begin{aligned}
L_t &= \mathbb{E}_{\mathbf{x}_t, \mathbf{x}_{t-1}, \mathbf{x}_0} \left[\left(f_\theta(\mathbf{x}_t, t) - \mathbf{x}_{t-1} \right)^2 \right] && (36) \\
&= \mathbb{E}_{\mathbf{x}_t, \mathbf{x}_0} \left[\left(f_\theta(\mathbf{x}_t, t) - \mathbb{E}\left[\mathbf{x}_{t-1} | \mathbf{x}_0, \mathbf{x}_t \right] \right)^2 \right] && (37) \\
&= \mathbb{E}_{\mathbf{x}_t, \mathbf{x}_0} \left[\left(f_\theta(\mathbf{x}_t, t) - \frac{1}{1 - \bar{\alpha}_t} \left[\beta_t \sqrt{\bar{\alpha}_{t-1}} \mathbf{x}_0 + \sqrt{1 - \beta_t} \left(1 - \bar{\alpha}_{t-1} \right) \mathbf{x}_t \right] \right)^2 \right]. && (38)
\end{aligned}$$

The expectation with respect to $\mathbf{x}_{t-1}$ has been eliminated. We consider the change of equation using Eq. (16). From the equation $\sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon_{\text{tot}}$, we consider a substitution of $\mathbf{x}_0$ with

$$\mathbf{x}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}} \left(\mathbf{x}_t - \sqrt{1 - \bar{\alpha}_t} \epsilon_{\text{tot}} \right), \tag{39}$$

to produce the objective function without $\mathbf{x}_0$,

$$\begin{aligned}
L_t &= \mathbb{E}_{\mathbf{x}_t, \epsilon_{\text{tot}}} \left[\left(f_\theta(\mathbf{x}_t, t) - \frac{1}{1 - \bar{\alpha}_t} \left[\frac{\beta_t \sqrt{\bar{\alpha}_{t-1}}}{\sqrt{\bar{\alpha}_t}} \left(\mathbf{x}_t - \sqrt{1 - \bar{\alpha}_t} \epsilon_{\text{tot}} \right) \right. \right. \right. \\
&\qquad \left. \left. \left. + \sqrt{1 - \beta_t} \left(1 - \bar{\alpha}_{t-1} \right) \mathbf{x}_t \right] \right)^2 \right] && (40) \\
&= \mathbb{E}_{\mathbf{x}_t, \epsilon_{\text{tot}}} \left[\left(f_\theta(\mathbf{x}_t, t) - \frac{1}{\sqrt{1 - \beta_t}} \left[\mathbf{x}_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_{\text{tot}} \right] \right)^2 \right]. && (41)
\end{aligned}$$

We consider the change of parameters using a new neural network $\epsilon_\theta(\mathbf{x}_t, t) \in \mathbb{R}^d$, and let the output of $f_\theta(\mathbf{x}_t, t)$ be

$$f_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{1 - \beta_t}} \left[\mathbf{x}_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(\mathbf{x}_t, t) \right]. \tag{42}$$

After we plug in this substitution, the objective function can be reformulated for a neural network $\epsilon_\theta(\mathbf{x}_t, t)$:

$$L_t = \mathbb{E}_{\mathbf{x}_t,\epsilon_{\text{tot}}} \left[\frac{1}{1-\bar{\alpha}_t} \frac{\beta_t^2}{1-\beta_t} \left(\epsilon_{\text{tot}} - \epsilon_\theta(\mathbf{x}_t, t)\right)^2 \right]. \tag{43}$$

We can consider an objective function for $\epsilon_\theta(\mathbf{x}_t, t)$:

$$L'_t = \mathbb{E}_{\mathbf{x}_t,\epsilon_{\text{tot}}} \left[\left(\epsilon_{\text{tot}} - \epsilon_\theta(\mathbf{x}_t, t)\right)^2 \right], \tag{44}$$

and the following empirical objective function:

$$\widehat{L'} = \frac{1}{TN} \sum_{t=1}^{T} \sum_{i=1}^{N} \left(\epsilon_{\text{tot},i}(t) - \epsilon_\theta(\mathbf{x}_{t,i}, t)\right)^2. \tag{45}$$

Here, $\epsilon_{\text{tot},i}(t)$ and $\mathbf{x}_{t,i}$ are the total noise added to the i-th sample of $\mathbf{x}_0$ and the resulting $\mathbf{x}_t$, respectively. The neural network $\epsilon_\theta(\mathbf{x}_t, t)$ is trained to predict the total noise added to $\mathbf{x}_0$ for given $\mathbf{x}_t$ and t. The objective function Eq. (45) can be iteratively updated using new noise samples at each iteration.

Using trained $\epsilon_\theta(\mathbf{x}_t, t)$, we move the current data $\mathbf{x}_t$ toward the opposite direction of the $\epsilon_\theta(\mathbf{x}_t, t) \in \mathbb{R}^d$. Considering the mean representation of $\mathbf{x}_{t-1}$ for given $\mathbf{x}_t$ and ϵ_{tot},

$$\mu_{\mathbf{x}_{t-1}|\mathbf{x}_t,\epsilon_{\text{tot}}} = \frac{1}{\sqrt{1-\beta_t}} \left[\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_{\text{tot}} \right], \tag{46}$$

the following update equation is used in [1] to perform the reverse process for $t' > T$:

$$\mathbf{x}_{t'} = \frac{1}{\sqrt{1-\beta_{2T-t'+1}}} \left[\mathbf{x}_{t'-1} - \frac{\beta_{2T-t'+1}}{\sqrt{1-\bar{\alpha}_{2T-t'+1}}} \epsilon_\theta(\mathbf{x}_{t'-1}, 2T - t' + 1) \right] + \sqrt{\sigma^2}\, \epsilon_{t'}, \tag{47}$$

with stochastic noise $\epsilon_{t'} \sim \mathcal{N}(0, I)$. The coefficient $\sqrt{\sigma^2}$ for the stochastic noise is not derived from rigorous discussion; however it signifies that diffusion continues during the reverse process. The next section provides a discussion on the choice of the coefficient for stochastic noise in the reverse process.

3.2.2 Reverse process with stochastic noise

The coefficient of the stochastic noise in the reverse process is difficult to determine. We consider a one step forward and reverse processes. The update equation for forward process is

$$x_1 = \sqrt{1-\beta} x_0 + \sqrt{\sigma^2 \beta}\, \epsilon_1, \quad \epsilon_1 \in \mathcal{N}(0, 1), \tag{48}$$

for one-dimensional x_0 and x_1. Once the original density $q(\mathbf{x}_0)$ is a Gaussian with mean 0 and variance σ_0^2, the joint density for x_0 and x_1 is as follows:

$$q(x_0, x_1) = \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_0^2 & \sqrt{1-\beta}\sigma_0^2 \\ \sqrt{1-\beta}\sigma_0^2 & (1-\beta)\sigma_0^2 + \beta\sigma^2 \end{pmatrix}\right). \tag{49}$$

The conditional density $q(x_1|x_0)$ obtained from the joint density explains the forward process,

$$q(x_1|x_0) = \mathcal{N}(\sqrt{1-\beta}x_0, \beta\sigma^2). \tag{50}$$

Here, the mean and the variance correspond to the first and second terms, respectively, in the update equation, Eq. (48). Conversely, if we obtain $q(x_0|x_1)$ from the joint density,

$$q(x_0|x_1) = \mathcal{N}\left(\frac{\sqrt{1-\beta}\sigma_0^2}{(1-\beta)\sigma_0^2 + \beta\sigma^2}x_1,\ \frac{\beta\sigma_0^2\sigma^2}{(1-\beta)\sigma_0^2 + \beta\sigma^2}\right), \tag{51}$$

we can use the equation for constructing the reverse process. From the conditional density, we introduce x_2 and consider the update rule:

$$x_2 = \frac{\sqrt{1-\beta}\sigma_0^2}{(1-\beta)\sigma_0^2 + \beta\sigma^2}x_1 + \sqrt{\frac{\beta\sigma_0^2\sigma^2}{(1-\beta)\sigma_0^2 + \beta\sigma^2}}\ \epsilon_2, \quad \epsilon_2 \sim \mathcal{N}(0,1), \tag{52}$$

for the reverse process. The reverse process will reconstruct the marginal density $q(x_0)$.

If we consider the coefficient of the stochastic noise, $\sqrt{\frac{\beta\sigma_0^2\sigma^2}{(1-\beta)\sigma_0^2+\beta\sigma^2}}$, the coefficient not only depends on the diffusion noise coefficient, $\sqrt{\sigma^2\beta}$, but also depends on the variance of the original density of data, σ_0^2. The following reformulation of the diffusion coefficient,

$$\sqrt{\frac{\beta\sigma_0^2\sigma^2}{(1-\beta)\sigma_0^2 + \beta\sigma^2}} = \sqrt{\sigma^2\beta}\sqrt{\frac{1}{1+\beta\left(\frac{\sigma^2}{\sigma_0^2} - 1\right)}}, \tag{53}$$

deomonstrates that the coefficient is greater than the diffusion coefficient, $\sqrt{\sigma^2\beta}$, when the diffusion noise σ^2 exceeds the total variance of the underlying density, σ_0^2, and is less than the diffusion coefficient otherwise. Note that $q(x_2|x_1) = q(x_0 = x_2|x_1)$ is a non-unique solution for the reverse process; however any stochastic process is valid as long as the marginal densities are equivalent: $q(x_2) = q(x_0 = x_2)$.

3.2.3 Comparison to Deterministic Flow-based Models

Several algorithms aim to identify a vector field that transports data generated from a single Gaussian to samples likely derived from the data density. This vector field is a function that does not permit one-to-many mappings. Diffusion models focus on learning the movement of the mainstream emanating from the diffusion process. While diffusion is characterized by stochastic and mixing procedures, the mainstream movement out of the diffusion is represented as a vector field.

Normalizing flow as explained in [2] also involves obtaining vector fields that transform samples from Gaussian to the data. However, it eschews stochastic or mixing procedures. It adopts a series of non-stochastic neural networks that perform nonlinear coordinate transformations from data density to a single Gaussian. Data samples are generated by the inverse transformation of the Gaussian samples. In this model, the coordinate transformation is monotonic, deterministic, and invertible. Because the transformation is monotonic, there is no mixing of sample points when neural networks are used to learn the transformation.

Due to these properties of normalizing flow, the resulting vector fields differ from those of diffusion models. The monotonic and deterministic nature dictates that empty spaces should map to empty spaces. Therefore, if the data density is topologically distinct from Gaussian, normalizing flow encounters difficulties in identifying a simple flow that fills the empty spaces typically expected in diffusion models.

The difference can also be illustrated using a Gaussian mapping. Suppose the density function for data $\mathbf{x}_0$ is Gaussian with variance of σ_0^2. Then, the joint density of $\mathbf{x}_0$ and $\mathbf{x}_T$ in the diffusion model should have the covariance matrix $\begin{pmatrix} \sigma_0^2 & 0 \\ 0 & \sigma^2 \end{pmatrix}$. When using normalizing flow to map $\mathbf{x}_0$ to a point in a Gaussian with variance σ^2, and assuming that $\mathbf{x}_0$ and the mapped Gaussian points are jointly Gaussian, the only possible covariance is $\lim_{\delta \to 0^+} \begin{pmatrix} \sigma_0^2 & \sigma_0\sigma - \delta \\ \sigma_0\sigma - \delta & \sigma^2 \end{pmatrix}$, a rank-one matrix indicative of a deterministic mapping. In a normalizing flow, a sample mapped to a Gaussian density should exactly recover the original sample via inverse transformation. Conversely, a sample that has diffused to a Gaussian may become a completely different point after the reverse process.

4 Continuous Diffusion

We return to our original discussion on the physics of diffusion processes. With an infinitesimal time step, Δt, in the diffusion process, the update equation that transforms the data density $\rho(0)$ to a Gaussian with covariance $\sigma^2 I$ can be provided as

$$\mathbf{x}_{t+\Delta t} = \sqrt{1 - \beta_t \Delta t}\mathbf{x}_t + \sqrt{\sigma^2 \beta_t \Delta t}\ \epsilon_t, \tag{54}$$

with $\epsilon_t \sim \mathcal{N}(0, I)$. Using the update rule, the mean $\mu_{t+\Delta t}$ and the covariance $\Sigma_{t+\Delta t}$ for $\mathbf{x}_{t+\Delta t}$ can be expressed in terms of the mean μ_t and the covariance Σ_t for $\mathbf{x}_t$:

$$\begin{aligned} \mu_{t+\Delta t} &= \mathbb{E}[\mathbf{x}_{t+\Delta t}] \ = \ \sqrt{1 - \beta_t \Delta t}\mathbb{E}[\mathbf{x}_t] && (55) \\ \Sigma_{t+\Delta t} &= \mathbb{E}[\mathbf{x}_{t+\Delta t}\mathbf{x}_{t+\Delta t}^\top] - \mathbb{E}[\mathbf{x}_{t+\Delta t}]\mathbb{E}[\mathbf{x}_{t+\Delta t}]^\top && (56) \\ &= (1 - \beta_t \Delta t)\mathbb{E}[\mathbf{x}_t \mathbf{x}_t^\top] + \beta_t \Delta t \sigma^2 I - (1 - \beta_t \Delta t)\mathbb{E}[\mathbf{x}_t]\mathbb{E}[\mathbf{x}_t]^\top && (57) \\ &= (1 - \beta_t \Delta t)\Sigma_t + \beta_t \Delta t \sigma^2 I. && (58) \end{aligned}$$

If we calculate the stationary covariance Σ, where $\Sigma = \Sigma_{t+\Delta t} = \Sigma_t$, using the equation

$$\Sigma = (1 - \beta_t \Delta t)\Sigma + \beta_t \Delta t \sigma^2 I, \tag{59}$$

then the stationary covariance resulting from this update is

$$\Sigma = \sigma^2 I, \tag{60}$$

which is achieved after a sufficient number of update iterations.

4.1 Differential equation for the update

Our interest is the velocity field produced by the infinitesimal update. From Eq. (54), the velocity equation we obtain is[3]

$$\begin{aligned}
\mathbf{v}_t &= \dot{\mathbf{x}}|_t = \lim_{\Delta t \to 0} \frac{\mathbf{x}_{t+\Delta t} - \mathbf{x}_t}{\Delta t} && (61)\\
&= \lim_{\Delta t \to 0} \frac{\sqrt{1 - \beta_t \Delta t} - 1}{\Delta t} \mathbf{x}_t + \frac{\sqrt{\sigma^2 \beta_t \Delta t}}{\Delta t} \epsilon_t. && (62)
\end{aligned}$$

The coefficient for the first term is calculated as follows:

$$\lim_{\Delta t \to 0} \frac{\sqrt{1 - \beta_t \Delta t} - 1}{\Delta t} = \lim_{\Delta t \to 0} \frac{-\frac{\beta_t}{2} \Delta t}{\Delta t} = -\frac{\beta_t}{2}, \tag{63}$$

which results in a drift of the flow towards the origin by the amount $\mathbf{v} = -\frac{\beta_t}{2}\mathbf{x}$. The resulting change in density due to the first term can be calculated as follows:

$$\begin{aligned}
\frac{d\rho}{dt} &= -\nabla \cdot J = -\nabla \cdot (\rho \mathbf{v}) && (64)\\
&= \frac{\beta_t \rho}{2} \nabla \cdot \mathbf{x} && (65)\\
&= \frac{\beta_t \rho d}{2}. && (66)
\end{aligned}$$

Here, d is the dimensionality of the data. We can recognize that the density increases at all points proportionally to the density at the same point and to the dimensionality.

The contribution of the second term to the flow arises from the movement of the collective particles. The procedure, formulated as $\mathbf{x}_{t+\Delta t} = a\mathbf{x}_t + \sqrt{\sigma^2 \beta_t \Delta t}\epsilon_t$ explains the rate of conditional covariance increase, given by $\frac{\mathbb{E}[(\mathbf{x}_{t+\Delta t} - a\mathbf{x}_t)^2 | \mathbf{x}_t]}{\Delta t} = \sigma^2 \beta_t I$, for a constant a. The procedure in Eq. (62) produces the flow that convolutes the density with a Gaussian having the covariance $\sigma^2 \beta_t \Delta t I$. Let the $\rho(\mathbf{x}; t + \Delta t)$ be the convoluted

[3] The vector field $\mathbf{v}_t$ at time t is a function of spatial points. Accordingly, $\dot{\mathbf{x}}|_t$, $\mathbf{x}_{t+\Delta t}$, and $\mathbf{x}_t$ are also functions of spatial points.

density of $\rho(\mathbf{x}; t)$ with a Gaussian kernel,

$$k(\mathbf{x}) = \frac{1}{\sqrt{2\pi\sigma^2\beta_t\Delta t}^d} \exp\left(-\frac{\mathbf{x}^2}{2\sigma^2\beta_t\Delta t}\right). \tag{67}$$

The convoluted density can be formulated as

$$\rho(\mathbf{x}; t+\Delta t) = \int \rho(\mathbf{x}'; t) k(\mathbf{x}' - \mathbf{x}) d\mathbf{x}', \tag{68}$$

where $\mathbf{x}'$ serves as a dummy variable. We consider a substitution $\mathbf{z} = \frac{\mathbf{x}'-\mathbf{x}}{\sqrt{\sigma^2\beta_t\Delta t}}$ with fixed $\mathbf{x}$ and introduce a new kernel for $\mathbf{z}$:

$$k(\mathbf{z}) = \frac{1}{\sqrt{2\pi}^d} \exp\left(-\frac{\mathbf{z}^2}{2}\right). \tag{69}$$

Along with $\mathbf{x}' = \sqrt{\sigma^2\beta_t\Delta t}\ \mathbf{z} + \mathbf{x}$, $d\mathbf{x}' = \sqrt{\sigma^2\beta_t\Delta t}^d d\mathbf{z}$, and $k(\mathbf{z}) = \sqrt{\sigma^2\beta_t\Delta t}^d k(\mathbf{x}' - \mathbf{x})$[4], the convolution can be written as

$$\int \rho(\mathbf{x}'; t) k(\mathbf{x}' - \mathbf{x}) d\mathbf{x}' \tag{72}$$

$$= \int \rho\Big(\mathbf{x} + \sqrt{\sigma^2\beta_t\Delta t}\,\mathbf{z}; t\Big) \frac{k(\mathbf{z})}{\cancel{\sqrt{\sigma^2\beta_t\Delta t}^d}} \cancel{\sqrt{\sigma^2\beta_t\Delta t}^d} d\mathbf{z} \tag{73}$$

$$= \int \rho\Big(\mathbf{x} + \sqrt{\sigma^2\beta_t\Delta t}\,\mathbf{z}; t\Big)\, k(\mathbf{z})\ d\mathbf{z}. \tag{74}$$

We use Taylor expansion for a fixed $\mathbf{x}$ and small $\mathbf{z}$,

$$\begin{aligned} &\rho\Big(\mathbf{x} + \sqrt{\sigma^2\beta_t\Delta t}\,\mathbf{z}; t\Big) \\ \approx\ & \rho(\mathbf{x}; t)\ +\ \sqrt{\sigma^2\beta_t\Delta t}\ \nabla^\top\rho\ \mathbf{z}\ +\ \frac{\sigma^2\beta_t\Delta t}{2}\mathbf{z}^\top\nabla\nabla\rho\ \mathbf{z}\ . \end{aligned} \tag{75}$$

Because $\int k(\mathbf{z})d\mathbf{z} = 1$, $\int \mathbf{z}k(\mathbf{z})d\mathbf{z} = 0$, and $\int k(\mathbf{z}) tr[\mathbf{z}\mathbf{z}^\top] d\mathbf{z} = 1$, the convolution can be rewritten as

$$\begin{aligned} &\text{Eq. (74)} \\ =\ & \rho(\mathbf{x}; t)\int k(\mathbf{z})d\mathbf{z}\ +\ \sqrt{\sigma^2\beta_t\Delta t}\ \nabla^\top\rho\cancel{\int \mathbf{z}k(\mathbf{z})d\mathbf{z}}\ +\ \frac{\sigma^2\beta_t\Delta t}{2} tr\left[\nabla\nabla\rho\int \mathbf{z}\mathbf{z}^\top k(\mathbf{z})d\mathbf{z}\right] \end{aligned}$$

[4] $k(\mathbf{z}) = \sqrt{\sigma^2\beta_t\Delta t}^d k(\mathbf{x}' - \mathbf{x})$ can be derived from

$$k(\mathbf{z}) = \frac{1}{\sqrt{2\pi}^d}\exp\left(-\frac{\mathbf{z}^2}{2}\right) = \frac{1}{\sqrt{2\pi}^d}\exp\left(-\frac{(\mathbf{x}' - \mathbf{x})^2}{2\sigma^2\beta_t\Delta t}\right) \tag{70}$$

$$= \sqrt{\sigma^2\beta_t\Delta t}^d k(\mathbf{x}' - \mathbf{x}). \tag{71}$$

$$= \rho(\mathbf{x};t) \ + \ \frac{\sigma^2 \beta_t \Delta t}{2} \nabla^2 \rho \ . \tag{76}$$

Note that we used $\nabla^2 \rho = tr[\nabla\nabla\rho] = \sum_{i=1}^{d} \frac{\partial^2 \rho}{\partial x_i^2}$. Therefore from

$$\rho(\mathbf{x}; t + \Delta t) = \rho(\mathbf{x};t) + \frac{\sigma^2 \beta_t \Delta t}{2} \nabla^2 \rho, \tag{77}$$

we obtain the change rate of ρ due to pure diffusion:

$$\frac{d\rho}{dt} = \lim_{\Delta t \to 0} \frac{\rho(\mathbf{x}; t + \Delta t) - \rho(\mathbf{x};t)}{\Delta t} = \frac{\sigma^2 \beta_t}{2} \nabla^2 \rho. \tag{78}$$

From the derivation, we obtain the diffusion parameter D in Eq. (3) is

$$D = \frac{\sigma^2 \beta_t}{2}, \tag{79}$$

and the velocity field that produces this rate of change is

$$\mathbf{v} = -D \nabla \log \rho = -\frac{\sigma^2 \beta_t}{2} \nabla \log \rho, \tag{80}$$

from Eq. (4).

By combining the first and second terms in Eq. (66) and Eq. (78), respectively, the density change by the update in Eq. (54) with small Δt is

$$\frac{d\rho(\mathbf{x};t)}{dt} = \beta_t \left(\frac{d}{2} - \frac{\sigma^2}{2} \nabla^2 \right) \rho(\mathbf{x};t), \tag{81}$$

and the velocity field due to the update is

$$\mathbf{v}_t = -\frac{\beta_t}{2} \left(\mathbf{x} + \sigma^2 \nabla \log \rho(\mathbf{x};t) \right) . \tag{82}$$

4.2 β-independent diffusive velocity fields

If current $\rho(\mathbf{x};t)$ is $\mathcal{N}(0, \sigma^2 I)$, which is the terminal density of the update in Eq. (54), the score vector field is

$$\nabla \log \rho(\mathbf{x};t) \ = \nabla \log \left(\frac{1}{\sqrt{2\pi}^d} \exp \left(-\frac{\mathbf{x}^2}{2\sigma^2} \right) \right) \tag{83}$$

$$= -\frac{\mathbf{x}}{\sigma^2}. \tag{84}$$

Then the resulting velocity field by the update does not change the density because

$$\mathbf{v}_t = -\frac{\beta_t}{2}\left(\mathbf{x} + \sigma^2 \nabla \log \rho(\mathbf{x}; t)\right) = -\frac{\beta_t}{2}\left(\mathbf{x} + \sigma^2\left(-\frac{\mathbf{x}}{\sigma^2}\right)\right) = 0. \tag{85}$$

In addition to the terminal density, there are other nonintuitive diffusion fields where the velocity field is independent of the update parameter. A kernel function $k(\mathbf{x}; t) = \mathcal{N}(0, \beta t)$ after time t represents one such density under a diffusive update $\mathbf{x}_{t+\Delta t} = \mathbf{x}_t + \sqrt{\beta \Delta t}\epsilon_t$ with time-independent parameter β. If we consider $k(\mathbf{x}; t)$, its convolution using $k(\mathbf{x}; \Delta t)$ yields a new kernel, represented as $k(\mathbf{x}; t + \Delta t) = \int k(\mathbf{x}'; t)k(\mathbf{x}' - \mathbf{x}; \Delta t)d\mathbf{x}'$. Therefore, $k(\mathbf{x}; t)$ is a solution for the density that we obtain from the diffusive update. In order to obtain diffusive velocity field, we consider the following diffusion equation:

$$\nabla \cdot (\mathbf{v}\ k(\mathbf{x}; t)) = -\frac{dk(\mathbf{x}; t)}{dt}, \tag{86}$$

which yields the following derivations:

$$\begin{aligned} k(\mathbf{x}; t)\nabla \cdot \mathbf{v} + \mathbf{v} \cdot \nabla k(\mathbf{x}; t) &= -\frac{dk(\mathbf{x}; t)}{dt} &(87)\\ k(\mathbf{x}; t)\left[\nabla \cdot \mathbf{v} - \frac{\mathbf{x}}{\beta t} \cdot \mathbf{v}\right] &= -\frac{1}{2t}\left(\frac{\mathbf{x}^2}{\beta t} - d\right) k(\mathbf{x}; t) &(88)\\ \nabla \cdot \mathbf{v} - \frac{\mathbf{x}}{\beta t} \cdot \mathbf{v} &= -\frac{1}{2t}\left(\frac{\mathbf{x}^2}{\beta t} - d\right). &(89) \end{aligned}$$

If we let $\mathbf{v} = f(t)\mathbf{x}$, then

$$f(t)d - \frac{f(t)}{\beta t}\mathbf{x}^2 = -\frac{1}{2t}\frac{\mathbf{x}^2}{\beta t} + \frac{d}{2t}, \tag{90}$$

and we can obtain $f(t) = \frac{1}{2t}$. Therefore, the diffusive velocity field is

$$\mathbf{v}_t = \frac{1}{2t}\mathbf{x}, \tag{91}$$

which does not depend on β. Although the velocity field is independent of the choice β, the flux and the density $k(\mathbf{x}; t)$ depends on β. When t approaches zero, $k(\mathbf{x}; t)$ becomes a δ-function. and the velocity field for diffusion approaches infinity at all points.

4.3 Reverse process for diffusion generative models

Now we consider how we can reverse the diffusion process and reconstruct the original density. The velocity field of the forward update in Eq. (54) is

$$\mathbf{v}_t = -\frac{\beta_t}{2}\left(\mathbf{x} + \sigma^2 \nabla \log \rho(\mathbf{x}; t)\right), \tag{92}$$

for $0 \leq t \leq T$ as in Eq. (82). After we finish the forward process, we consider reversing the process by implementing the following opposite velocity field during $T < t' \leq 2T$:

$$\begin{aligned} \mathbf{v}_{t'} &= -\mathbf{v}_{2T-t'} && (93) \\ &= \frac{\beta_{2T-t'}}{2}\left(\mathbf{x} \ + \ \sigma^2 \nabla \log \rho\left(\mathbf{x};\ 2T-t'\right)\right). && (94) \end{aligned}$$

Then the data density will be reconstructed when t' reaches $2T$. Here, $\beta_{2T-t'}$ and $\rho(\mathbf{x};\ 2T-t')$ represent β_t and $\rho(\mathbf{x};t)$ at $t = 2T - t'$, with the reverse process time t' falling between T and $2T$.

We implement the reverse process by applying an advection control $\mathbf{u}_c$ which reverses the total velocity. $\mathbf{u}_{c,t'} = -2\mathbf{v}_{2T-t'}$, and the update rule changes into

$$\mathbf{x}_{t'+\Delta t} \ = \ \sqrt{1-\beta_{2T-t'}\Delta t}\ \mathbf{x}_{t'} + \sqrt{\sigma^2 \beta_{2T-t'}\Delta t}\ \epsilon_{t'} + \mathbf{u}_{c,t'}\Delta t. \tag{95}$$

By the plug in of $\mathbf{u}_{c,t'} \ = \ -2\mathbf{v}_{2T-t'} \ = \ \beta_{2T-t'}\left(\mathbf{x} + \sigma^2 \nabla \log \rho(\mathbf{x};\ 2T-t')\right)$, we obtain the update rule for reverse process.

$$\begin{aligned} \mathbf{x}_{t'+\Delta t} \ = \ & \left(\sqrt{1-\beta_{2T-t'}\Delta t} \ + \ \beta_{2T-t}\Delta t\right)\mathbf{x}_t \\ & + \Delta t\, \beta_{2T-t'}\sigma^2 \nabla \log \rho(\mathbf{x};\ 2T-t') \ + \ \sqrt{\sigma^2 \beta_{2T-t'}\Delta t}\ \epsilon_{t'}, && (96) \end{aligned}$$

for $0 \leq t \leq T$. If we memorize the history of score $\nabla \log \rho(\mathbf{x};t)$ during forward processes $T < t' \leq 2T$, the update will make reverse processes to produce data generated from original density.

Note that the update equation for the reverse processes continue the diffusion, while the advection control creates a flow that reverses the movement. The control mechanism in the reverse process can be likened to a physics system in which diffusion and advection coexist. For example, consider a breeze that directs a flock of flies in a specific direction or a moving belt that produces directed and consistent movement amidst random motion on the surface of the belt. The velocity from the control and the diffusive velocity together create a flow that returns to the original density.

4.4 Relation to kernel density estimation

Nonparametric methods that utilize kernel density estimation can be analyzed by examining the convoluted density. The smoothing of a density through colvolution intoduces a bias in the prediction, at the cost of enhanced generalization and reduced variance. The derivations presented in [3, 4] explain the bias resulting from convoluted densities when solving problems that involve comparing two densities, such as in classification or f-divergence estimation.

For example, we consider kernel density estimation defined by

$$\widehat{\rho}(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} k_h(\mathbf{x} - \mathbf{x}_i), \tag{97}$$

using data $\mathbf{x}_1, \ldots, \mathbf{x}_N \in \mathbb{R}^d$, with a kernel function $k_h(\mathbf{x}-\mathbf{x}') = \frac{1}{\sqrt{2\pi h}^d} \exp\left(-\frac{(\mathbf{x}-\mathbf{x}')^2}{2h}\right)$ having a small bandwidth h. We can derive the deviation of the expected estimation from the true value. Let the posterior for class 1 be

$$f(\mathbf{x}) = \frac{\rho_1(\mathbf{x})}{\rho_1(\mathbf{x}) + \rho_2(\mathbf{x})}, \tag{98}$$

where $\rho_1(\mathbf{x})$ and $\rho_2(\mathbf{x})$ are the densities for class 1 and 2, respectively, and we assume that the class priors are equivalent. Let $\widehat{\rho}_1(\mathbf{x})$ and $\widehat{\rho}_2(\mathbf{x})$ represent the corresponding kernel density estimates using data from classes 1 and 2, respectively.

The expectation of the kernel density estimate can be calculated as

$$\mathbb{E}[\widehat{\rho}(\mathbf{x})] \approx \rho(\mathbf{x}) + \frac{h^2}{2}\nabla^2\rho|_{\mathbf{x}}, \tag{99}$$

and the posterior, when using the plug-in of the individual kernel density estimators, has the expectation

$$\begin{aligned}
\mathbb{E}[f(\mathbf{x})] &= \frac{\mathbb{E}[\widehat{\rho}_1(\mathbf{x})]}{\mathbb{E}[\widehat{\rho}_1(\mathbf{x})] + \mathbb{E}[\widehat{\rho}_2(\mathbf{x})]} && (100)\\
&\approx \frac{\rho_1(\mathbf{x}) + \frac{h^2}{2}\nabla^2\rho_1|_{\mathbf{x}}}{\rho_1(\mathbf{x}) + \frac{h^2}{2}\nabla^2\rho_1|_{\mathbf{x}} \ + \ \rho_2(\mathbf{x}) + \frac{h^2}{2}\nabla^2\rho_2|_{\mathbf{x}}} && (101)\\
&\approx f(\mathbf{x}) + \frac{h^2 f(\mathbf{x})(1-f(\mathbf{x}))}{2}\left(\frac{\nabla^2\rho_1|_{\mathbf{x}}}{\rho_1(\mathbf{x})} - \frac{\nabla^2\rho_2|_{\mathbf{x}}}{\rho_2(\mathbf{x})}\right), && (102)
\end{aligned}$$

because each kernel density estimator, $\widehat{\rho}_1(\mathbf{x})$ and $\widehat{\rho}_2(\mathbf{x})$, converges to its respective expectation.

When we use empirically estimated kernel density estimators, the bias in posterior estimation is related to the diffusion properties of the original densities. The difference $\frac{\nabla^2\rho_1}{\rho_1(\mathbf{x})} - \frac{\nabla^2\rho_2}{\rho_2(\mathbf{x})}$ is proportional to the difference $\frac{1}{\rho_1(\mathbf{x})}\frac{d\rho_1}{dt} - \frac{1}{\rho_2(\mathbf{x})}\frac{d\rho_2}{dt}$ in a diffusive system. The difference $\frac{\nabla^2\rho_1}{\rho_1(\mathbf{x})} - \frac{\nabla^2\rho_2}{\rho_2(\mathbf{x})}$ appears in many bias derivations comparing two densities as shown in [3, 4]. The rate of change in density, $\frac{d\rho}{dt}$, relative to the density amount, $\rho(\mathbf{x})$, is associated with the bias.

5 Physics-based Models Using Force Field

The motivation for the continuous forward and reverse processes lies in the physics system that mimics the advection and diffusion processes. When considering other tasks that contrast two classes, introducing a potential function may be beneficial. In [5], advection using potential function has been introduced. A Bhattacharyya criterion is a well-known surrogate loss that can measure the separation of two densities $\rho_1(\mathbf{x})$

and $\rho_2(\mathbf{x})$:

$$U(\rho_1, \rho_2) = \int \sqrt{\rho_1(\mathbf{x})\rho_2(\mathbf{x})} d\mathbf{x}. \tag{103}$$

We can consider the dynamics of $\rho_2(\mathbf{x})$ when a physical system attempts to minimize the potential function. The system will decrease the potential amount by

$$\frac{dU}{dt} = \int \frac{d}{dt}\left(\sqrt{\rho_1(\mathbf{x})\rho_2(\mathbf{x};t)}\right) d\mathbf{x} \tag{104}$$

$$= \int \frac{1}{2}\sqrt{\frac{\rho_1(\mathbf{x})}{\rho_2(\mathbf{x};t)}} \frac{d\rho_2(\mathbf{x};t)}{dt} d\mathbf{x}. \tag{105}$$

We use the following equation of continuity to represent the time derivative of $\rho_2(\mathbf{x};t)$

$$\frac{d\rho_2(\mathbf{x};t)}{dt} = -\nabla \cdot (\rho_2 \mathbf{v}) \tag{106}$$

Consequently, we can derive the equation for the physical system that relates the time derivative of the potential to the movement of mass:

$$\frac{dU}{dt} = \frac{1}{2}\int \sqrt{\frac{\rho_1(\mathbf{x})}{\rho_2(\mathbf{x};t)}} \left(-\nabla \cdot (\rho_2(\mathbf{x};t)\mathbf{v})\right) d\mathbf{x} \tag{107}$$

$$= \frac{1}{2}\int \left(\rho_2(\mathbf{x};t)\nabla\sqrt{\frac{\rho_1(\mathbf{x})}{\rho_2(\mathbf{x};t)}}\right) \cdot \mathbf{v}\ d\mathbf{x}. \tag{108}$$

Here, the final equation is derived using integration by parts. The change in potential is due to the displacement of total mass in response to the force field. The following equation can be introduced to determine the force field in this system:

$$dU = -\int F \cdot d\mathbf{s}\ d\mathbf{x}. \tag{109}$$

Here, $d\mathbf{s}$ represents the infinitesimal movement of mass, and $d\mathbf{x}$ is the volume element. Then, from Eq. (108), the force field acting on ρ_2 from ρ_1 can be obtained as

$$F_{1\to 2} = -\frac{1}{2}\rho_2 \nabla \sqrt{\frac{\rho_1}{\rho_2}}\ . \tag{110}$$

The derivations in [5] show that this force field is well-connected to statistical properties. For example, if both $\rho_1(\mathbf{x})$ and $\rho_2(\mathbf{x})$ are Gaussians that share an equivalent covariance matrix, then the direction of acceleration chosen by the rigid mass ρ_2 aligns with the embedding direction selected by traditional Fisher discriminant analysis:

$$\ddot{\mathbf{x}} \propto \Sigma^{-1}(\mu_1 - \mu_2), \tag{111}$$

where Σ is the shared covariance matrix and μ_1 and μ_2 are the means of the two Gaussians.

6 Conclusion

A physical system that illustrates the forward and reverse diffusion processes for data generation was presented. The forward process is a diffusion process that generates the diffusive flow, and this flow pattern should be stored in the neural networks. The reverse process utilizes the stored information to reconstruct the original density.

The derivations involving infinitesimal time updates lead to a flow that utilizes the score vector field. Comparisons with deterministic flow-based models demonstrate the flexibility of achieving a simple mapping from data density to a Gaussian.

A force field can be derived when we aim to separate one class from the others. A derivation from the literature has been introduced, which generates a force field for the movement to minimize the overlap between two classes in terms of the information-theoretic measure, the Bhattacharyya criterion.

References

[1] Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in Neural Information Processing Systems **33**, 6840–6851 (2020)

[2] Papamakarios, G., Nalisnick, E., Rezende, D.J., Mohamed, S., Lakshminarayanan, B.: Normalizing flows for probabilistic modeling and inference. The Journal of Machine Learning Research **22**(57), 2617–2680 (2021)

[3] Yoon, S., Park, F.C., Yun, G., Kim, I., Noh, Y.-K.: Variational weighting for kernel density ratios. Advances in Neural Information Processing Systems **37**, 5010–5027 (2023)

[4] Noh, Y.-K., Zhang, B.-T., Lee, D.D.: Generative local metric learning for nearest neighbor classification. IEEE Transactions on Pattern Analysis and Machine Intelligence **40**(1), 106–118 (2018) https://doi.org/10.1109/TPAMI.2017.2666151

[5] Noh, Y.-K., Hamm, J., Park, F.C., Zhang, B.-T., Lee, D.D.: Fluid dynamic models for Bhattacharyya-based discriminant analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence **40**(1), 92–105 (2018) https://doi.org/10.1109/TPAMI.2017.2666148