SDE and Score-based Diffusion

Weiyao Huang

Last Update: 2023.12.26

Sampling from a distribution

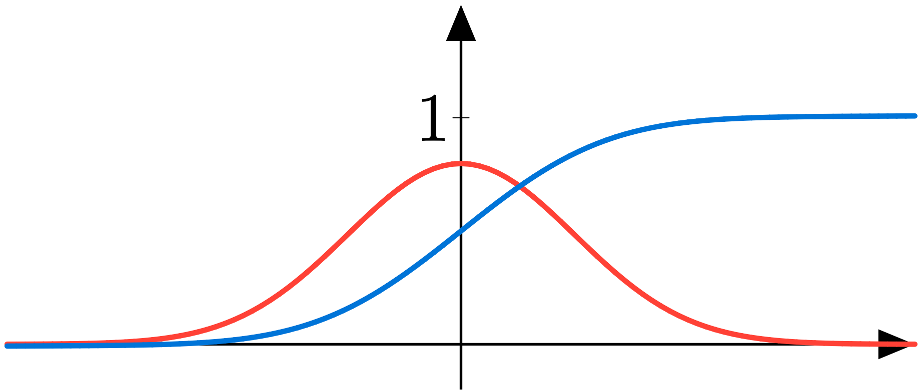
Many of the tasks in the field of generative AI can be formalized as sampling specific distribution.

Sampling from a distribution

Many of the tasks in the field of generative AI can be formalized as sampling specific distribution.

- Image generation: $P_D(X \mid c)$.
- Text generation: $P_D(X_i \mid X_1, X_2, ..., X_{i-1})$

Inversion Method: 1-Dimensional Case



To sample from $P(X)$:

1. Cumulative distribution func. $\Phi(x) = P(X \leq x) \in [0, 1]$;
2. Sample $Y \sim \text{Uniform}([0, 1])$;
3. Let $X = \Phi^{-1}(Y)$.

Metropolis-Hastings algorithm (MCMC) [1]

(Review of DMS lecture)

To sample $x \sim \pi$, we find a P s.t. $\pi P = \pi$, and sample from νP^n.

Theorem of detailed balance: If $\pi(x)P(y|x) = \pi(y)P(x|y)$, then $\pi P = \pi$.

To construct P, let Q be a markov chain:

$$Q(x|y) > 0 \Leftrightarrow Q(y|x) > 0.$$

Metropolis-Hastings algorithm (MCMC) [1]

We sample $y \sim Q(\cdot\,|x)$, and

- output y w.p. $a_{x,y}$ (accept);
- or output x w.p. $1 - a_{x,y}$ (reject).

where $P(y|x) = a_{x,y}Q(y|x)$ if $x \neq y$.

For detailed balance,

$$\pi(x)a_{x,y}Q(y|x) = \pi(y)a_{y,x}Q(x|y).$$

Let $a_{x,y} = \min\left(\frac{\pi(y)Q(x|y)}{\pi(x)Q(y|x)}, 1\right)$, then P is constructed.

Challenges

1. Precise mass func. is often difficult to give. Most of the time only samples are given.
2. Hypothesis space is very high dimensional (10^6 for image of resolution 1024×1024), which leads to very complex distribution.
3. Given samples are often too sparse to "recover" the distribution.

Power of Differential Methods

Calculate maximal point of $f : \mathbb{R}^s \to \mathbb{R}$?

There are a family of methods called gradient descenting algorithms.

Power of Differential Methods

Calculate maximal point of $f : \mathbb{R}^s \to \mathbb{R}$?

There are a family of methods called gradient descenting algorithms.

Sampling by mass func. looks like a randomized version of maximal point problem.

To achieve that, let's introduce some stochastic differential gadgets.

Stochastic Differential Equation

A general SDE is preseneted [2]

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = \boldsymbol{f}(\boldsymbol{x}, t) + \boldsymbol{g}(\boldsymbol{x}, t)\boldsymbol{w}(t).$$

$\boldsymbol{x}(t) \in \mathbb{R}^s$, $\boldsymbol{g}(\boldsymbol{x}, t) \in \mathbb{R}^{s \times s}$, $\boldsymbol{w}(t) \in \mathbb{R}^s$ is a white noise process.

White Noise Process

A white noise process $\boldsymbol{w}(t)$ is a random func. satisfying

1. $\boldsymbol{w}(t)$ and $\boldsymbol{w}(t')$ are independent if $t \neq t'$

$$\mathbb{E}[\boldsymbol{w}(t)] = \mathbf{0}.$$

White Noise Process

A white noise process $\boldsymbol{w}(t)$ is a random func. satisfying

1. $\boldsymbol{w}(t)$ and $\boldsymbol{w}(t')$ are independent if $t \neq t'$

$$\mathbb{E}[\boldsymbol{w}(t)] = \mathbf{0}.$$

2. the mapping $t \mapsto \boldsymbol{w}(t)$ is a **Guassian process** with zero mean Dirac delta correlation [3]

$$E\left[\boldsymbol{w}(t)\boldsymbol{w}^{T}(s)\right] = \delta(t-s)\boldsymbol{Q}.$$

where $\delta(x) \simeq \begin{cases} +\infty & x=0 \\ 0 & x\neq 0 \end{cases}, \int_{\mathbb{R}} \delta(x)\,\mathrm{d}x = 1$; $\boldsymbol{Q}$ is the spectral density.

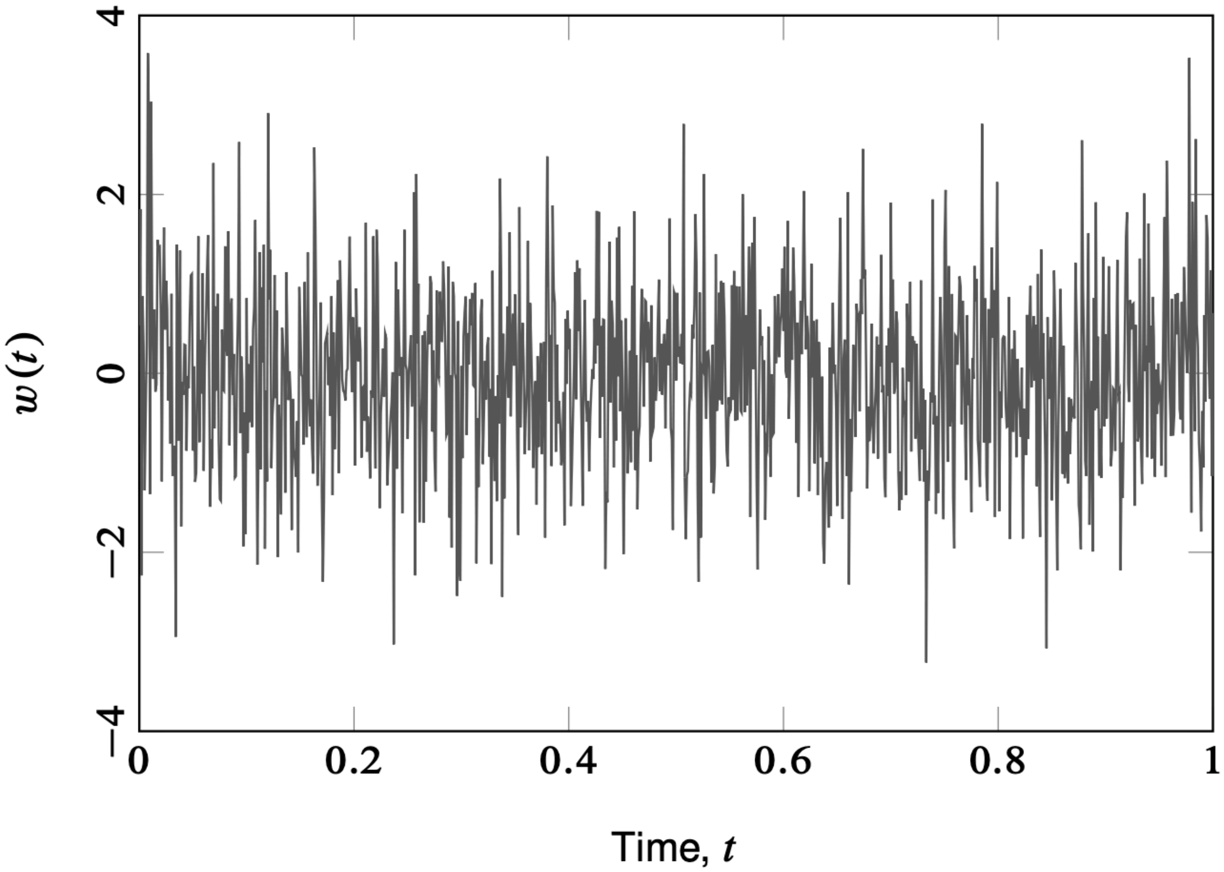


Figure 2: A possible trajectory of white noise

Itô Calculus [2]

$$\int_{t_0}^{t} \boldsymbol{g}(\boldsymbol{x}(t), t)\,\mathrm{d}\boldsymbol{\beta}(t) \stackrel{\text{def}}{=} \lim_{n\to\infty} \sum_{k=1}^{n} \boldsymbol{g}(\boldsymbol{x}(t_k), t_k)[\boldsymbol{\beta}(t_k) - \boldsymbol{\beta}(t_{k-1})].$$

where $t_0 < t_1 < \cdots < t_n = t$, $\boldsymbol{\beta} : \mathbb{R} \to \mathbb{R}^s$ denotes Brownian motion, a continuous stochastic process:

Itô Calculus [2]

$$\int_{t_0}^{t} \boldsymbol{g}(\boldsymbol{x}(t), t)\, \mathrm{d}\boldsymbol{\beta}(t) \stackrel{\text{def}}{=} \lim_{n\to\infty} \sum_{k=1}^{n} \boldsymbol{g}(\boldsymbol{x}(t_k), t_k)[\boldsymbol{\beta}(t_k) - \boldsymbol{\beta}(t_{k-1})].$$

where $t_0 < t_1 < \cdots < t_n = t$, $\boldsymbol{\beta} : \mathbb{R} \to \mathbb{R}^s$ denotes Brownian motion, a continuous stochastic process:

1. Brownian motion is nowhere differentiable.
2. White noise can be considered as the formal (or weak) derivative of Brownian motion, $\boldsymbol{w}(t) = \frac{\mathrm{d}\boldsymbol{\beta}(t)}{\mathrm{d}t}$.

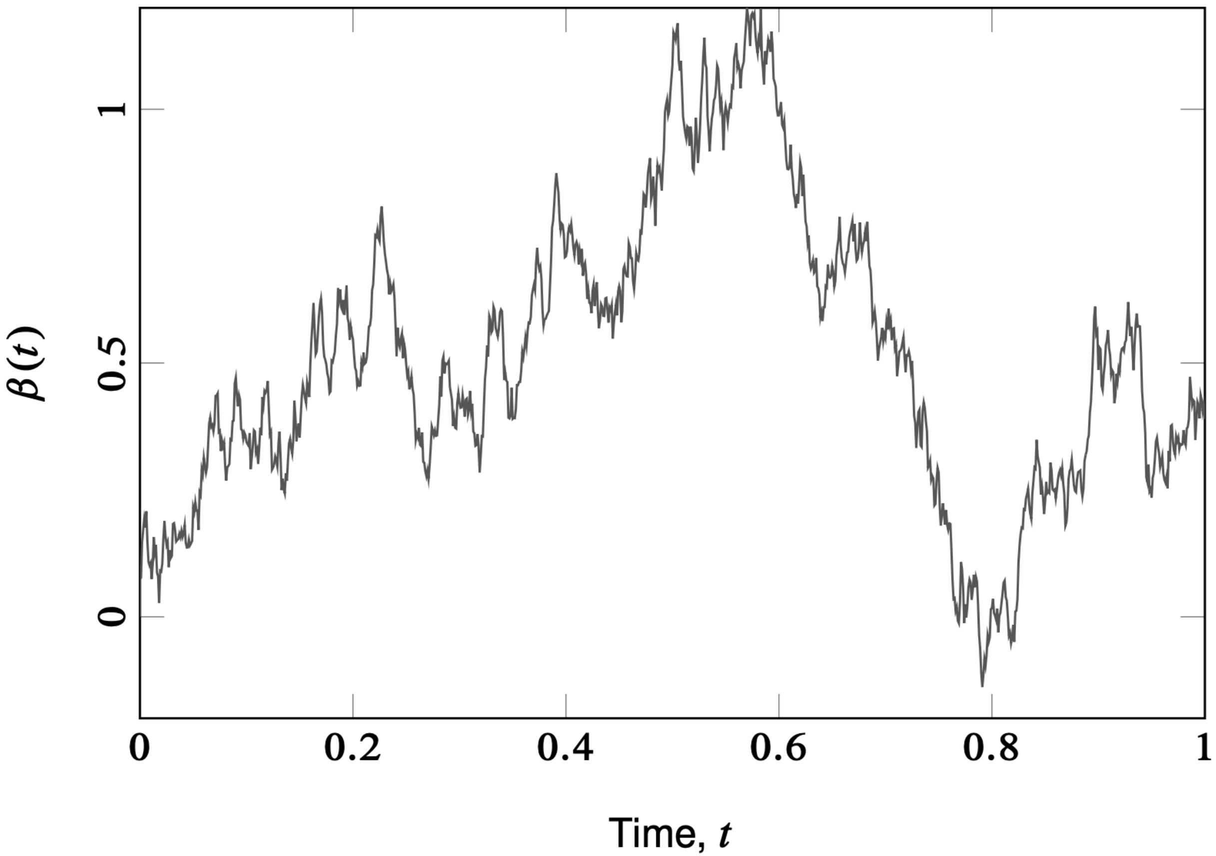


Figure 3: A possible trajectory of Brownian motion

Itô Diffusion

$$\mathrm{d}\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{x}, t)\,\mathrm{d}t + \boldsymbol{g}(\boldsymbol{x}, t)\,\mathrm{d}\boldsymbol{\beta}.$$

$$\boldsymbol{x}(t) - \boldsymbol{x}(t_0) = \int_{t_0}^{t} \boldsymbol{f}(\boldsymbol{x}(t), t)\,\mathrm{d}t + \int_{t_0}^{t} \boldsymbol{g}(\boldsymbol{x}(t), t)\boldsymbol{w}(t)\,\mathrm{d}t.$$

1. $\boldsymbol{f}(\boldsymbol{x}, t)$ is called the *drift function*, which determines the nominal dynamics of the system;
2. $\boldsymbol{g}(\boldsymbol{x}, t)$ is the *dispersion matrix*, which determines how the noise enters the system.

Sampling through Stochastic Process

Consider r.v. $\boldsymbol{x}_{\text{prior}}$ drawn from a prior distribution p_{prior}, typically a Guassian. Then we construct a stochastic process for $\boldsymbol{x}$ so that $\boldsymbol{x}_{\text{target}} \sim p_{\text{target}}$.

Sampling through Stochastic Process

Consider r.v. $\boldsymbol{x}_{\text{prior}}$ drawn from a prior distribution p_{prior}, typically a Guassian. Then we construct a stochastic process for $\boldsymbol{x}$ so that $\boldsymbol{x}_{\text{target}} \sim p_{\text{target}}$.

But this kind of process is difficult to construct or learn, since we know nothing about p_{target}.

Sampling through Stochastic Process

Consider r.v. $\boldsymbol{x}_{\text{prior}}$ drawn from a prior distribution p_{prior}, typically a Guassian. Then we construct a stochastic process for $\boldsymbol{x}$ so that $\boldsymbol{x}_{\text{target}} \sim p_{\text{target}}$.

But this kind of process is difficult to construct or learn, since we know nothing about p_{target}.

Consider another approach: Given $\boldsymbol{x}_0 \sim p_0 = p_{\text{target}}$, construct a process for $\boldsymbol{x}$ so that $\boldsymbol{x}_T \sim p_T = p_{\text{prior}}$, then we do the reverse process to draw samples.

Reversed SDE [4]

Let's say $\boldsymbol{x} \sim p(\boldsymbol{x}, t)$. Define $\boldsymbol{G} = \boldsymbol{g}(\boldsymbol{x}, t)\boldsymbol{g}(\boldsymbol{x}, t)^T$. then the reversed process is given by

$$\mathrm{d}\boldsymbol{x} = \overline{\boldsymbol{f}}(\boldsymbol{x}, t)\,\mathrm{d}t + \boldsymbol{g}(\boldsymbol{x}, t)\,\mathrm{d}\overline{\boldsymbol{\beta}}$$

$$\overline{f}^i(\boldsymbol{x}, t) = f^i(\boldsymbol{x}, t) - \left[\sum_j \nabla_{x_j} \ln p(\boldsymbol{x}, t)\boldsymbol{G}_{ij} + \frac{\partial \boldsymbol{G}_{ij}}{\partial x_j}\right]$$

$$\mathrm{d}\overline{\boldsymbol{\beta}} = \mathrm{d}\boldsymbol{\beta} + \frac{1}{p(\boldsymbol{x}, t)} \sum_{j,k} \nabla_{x_j}\left[p(\boldsymbol{x}, t) g^{jk}(\boldsymbol{x}, t)\right] \mathrm{d}t.$$

Reversed SDE: Simplified Version

When $\boldsymbol{g}$ only depends on t and $\boldsymbol{g}(t) = g(t)I$, the reversed process is simplified as

$$\begin{aligned} \mathrm{d}\boldsymbol{x} &= \left[\boldsymbol{f}(\boldsymbol{x}, t) - g^2(t)\nabla \ln p(\boldsymbol{x}, t)\right] \mathrm{d}t + g(t)\,\mathrm{d}\overline{\boldsymbol{\beta}} \\ \mathrm{d}\overline{\boldsymbol{\beta}} &= \mathrm{d}\boldsymbol{\beta} + g(t)\|\nabla \ln p(\boldsymbol{x}, t)\|_1\,\mathrm{d}t. \end{aligned}$$

By simulating this process, we can draw samples $\sim p(\boldsymbol{x}, 0)$.

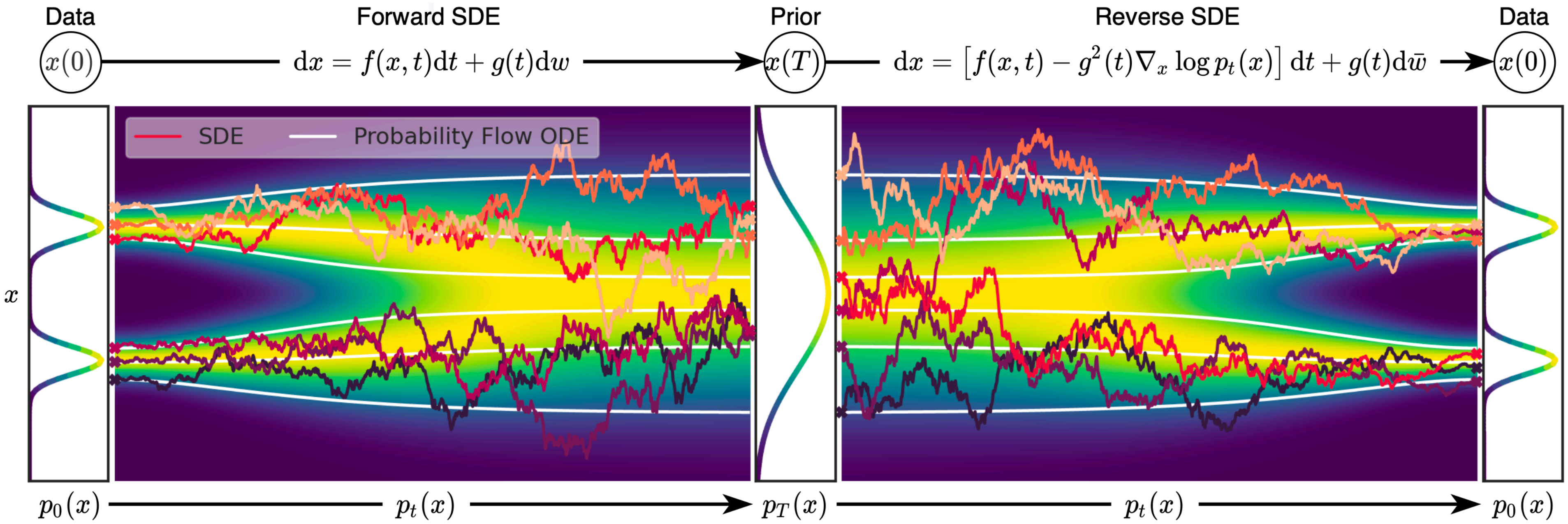


Figure 4: Score-based generative modeling through SDEs [5]

Example I: SMLD [5]

Consider the following Markov Chain ($1 \le i \le N$):

$$\boldsymbol{x}_i = \boldsymbol{x}_{i-1} + \sqrt{\sigma_i^2 - \sigma_{i-1}^2}\boldsymbol{z}_{i-1}.$$

where $\boldsymbol{z}_{i-1} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$.

Example I: SMLD [5]

Consider the following Markov Chain ($1 \le i \le N$):

$$\boldsymbol{x}_i = \boldsymbol{x}_{i-1} + \sqrt{\sigma_i^2 - \sigma_{i-1}^2}\boldsymbol{z}_{i-1}.$$

where $\boldsymbol{z}_{i-1} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$.

Let $N \to +\infty$ to make it continuous, the process is given by

$$\mathrm{d}\boldsymbol{x} = \sqrt{\frac{\mathrm{d}\sigma^2(t)}{\mathrm{d}t}}\,\mathrm{d}\boldsymbol{\beta}.$$

Example II: DDPM [5]

Consider the following Markov Chain ($1 \leq i \leq N$):

$$\boldsymbol{x}_i = \sqrt{1 - c_i}\boldsymbol{x}_{i-1} + \sqrt{c_i}\boldsymbol{z}_{i-1}.$$

where $\boldsymbol{z}_{i-1} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$.

Example II: DDPM [5]

Consider the following Markov Chain ($1 \leq i \leq N$):

$$\boldsymbol{x}_i = \sqrt{1 - c_i}\boldsymbol{x}_{i-1} + \sqrt{c_i}\boldsymbol{z}_{i-1}.$$

where $\boldsymbol{z}_{i-1} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$.

Let $N \to +\infty$ to make it continuous, the process is given by

$$\mathrm{d}\boldsymbol{x} = -\frac{1}{2}c(t)\boldsymbol{x}\,\mathrm{d}t + \sqrt{c(t)}\,\mathrm{d}\boldsymbol{\beta}.$$

SDE Simulation

$$\mathrm{d}\boldsymbol{x} = \left[\boldsymbol{f}(\boldsymbol{x}, t) - g^2(t)\nabla \ln p(\boldsymbol{x}, t)\right] \mathrm{d}t + g(t)\, \mathrm{d}\overline{\boldsymbol{\beta}}.$$

The remaining problem is:

1. How to simulate a stochastic process (SDE)?

SDE Simulation

$$\mathrm{d}\boldsymbol{x} = \left[\boldsymbol{f}(\boldsymbol{x}, t) - g^2(t)\nabla \ln p(\boldsymbol{x}, t)\right] \mathrm{d}t + g(t)\, \mathrm{d}\overline{\boldsymbol{\beta}}.$$

The remaining problem is:

1. How to simulate a stochastic process (SDE)?

   Brownion motion can be approximated by Guassian kernel.

2. How to calculate $\nabla \ln p(\boldsymbol{x}, t)$?

SDE Simulation

$$\mathrm{d}\boldsymbol{x} = \left[\boldsymbol{f}(\boldsymbol{x}, t) - g^2(t)\nabla \ln p(\boldsymbol{x}, t)\right] \mathrm{d}t + g(t)\, \mathrm{d}\overline{\boldsymbol{\beta}}.$$

The remaining problem is:

1. How to simulate a stochastic process (SDE)?

   Brownion motion can be approximated by Guassian kernel.

2. How to calculate $\nabla \ln p(\boldsymbol{x}, t)$?

   Model it using deep neural network.

References

[1] Wikipedia contributors, “Metropolis–Hastings algorithm --- Wikipedia, The Free Encyclopedia”. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Metropolis%E2%80%93Hastings_algorithm&oldid=1172902257

[2] S. Särkkä and A. Solin, *Applied Stochastic Differential Equations*. 2019. [Online]. Available: https://users.aalto.fi/~ssarkka/pub/sde_book.pdf

[3] Wikipedia contributors, “Dirac delta function --- Wikipedia, The Free Encyclopedia”. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Dirac_delta_function&oldid=1191296053

[4] B. D. Anderson, “Reverse-time diffusion equation models”, *Stochastic Processes and their Applications*, vol. 12, no. 3, pp. 313–326, 1982, doi: https://doi.org/10.1016/0304-4149(82)90051-5.

[5] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, “Score-Based Generative Modeling through Stochastic Differential Equations”. 2021.