DIFFUSION MODEL

SMLD and Score-based model

Author
Haitong Lan
lanhaitong@bupt.edu.cn
August 1, 2024

Contents

1 SMLD

SMLD(Score Matching with Langevin dynamics),see Fig(1).

Generative Modeling by Estimating Gradients of the Data Distribution

Yang Song
Stanford University
yangsong@cs.stanford.edu

Stefano Ermon
Stanford University
ermon@cs.stanford.edu

Figure 1: paper of SMLD

1.1 Genrative Model

What is generative AI,see Fig(2).

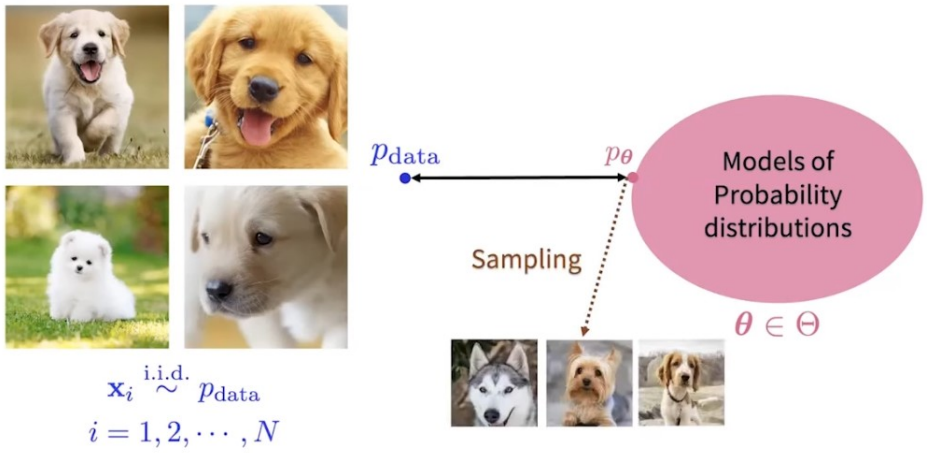


Figure 2: Generative AI schematic

Consider building deep neural networks that use Gaussian distributions to fit complex true distributions,see Fig(3).

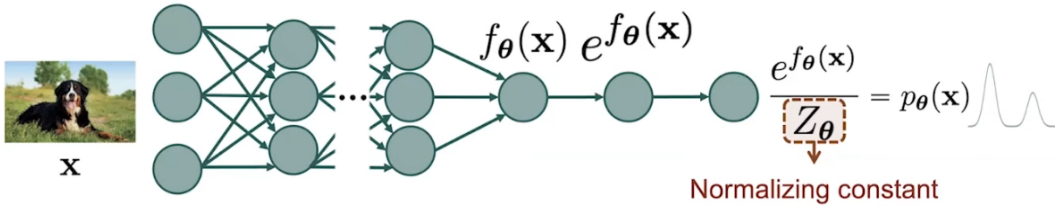


Figure 3: Models of Probability distribution

We often write the resulting distribution as

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = \frac{e^{-f_{\boldsymbol{\theta}}(\mathbf{x})}}{Z_\theta} \tag{1}$$

Proposition 1.1. Generative model = Models of Probability + Sampling.

1.2 Score Matching

Distributions can be expressed equivalent to Stein's score:

$$\mathbf{s}_\theta(\mathbf{x}) = \nabla_{\mathbf{x}} \log p_\theta(\mathbf{x}) = -\nabla_{\mathbf{x}} f_\theta(\mathbf{x}) - \underbrace{\nabla_{\mathbf{x}} \log Z_\theta}_{=0} = -\nabla_{\mathbf{x}} f_\theta(\mathbf{x}). \tag{2}$$

If we can find the Score of a variable, it is equivalent to finding its distribution, and score-matching is a way to estimate the Score,see Fig(4).

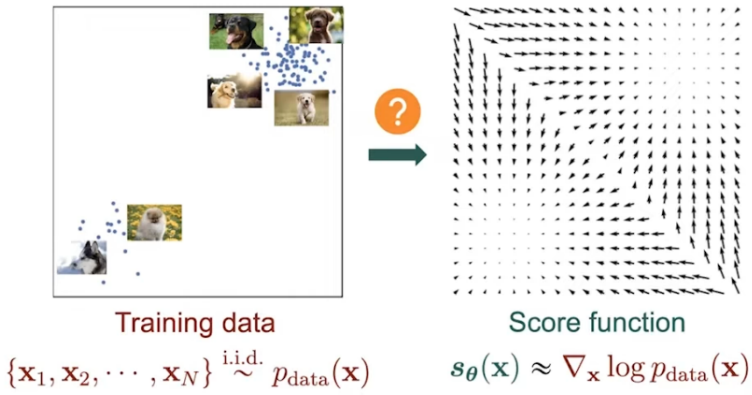


Figure 4: Estimate the score from the sample

The Objective function of Score-Matching is

$$\frac{1}{2}\mathbb{E}_{p_{\text{data}}}\left[\left\|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}) - \nabla_{\mathbf{x}} \log p_{\text{data}}\,(\mathbf{x})\right\|_2^2\right] \tag{3}$$

However, $p_{data}(\mathbf{x})$is unknown and cannot be computed, eq.(3)can be equivalent written as

$$\mathbb{E}_{p_{\text{data}}\,(\mathbf{x})}\left[\operatorname{tr}\left(\nabla_{\mathbf{x}}\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x})\right) + \frac{1}{2}\left\|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x})\right\|_2^2\right] \tag{4}$$

where $\nabla_{\mathbf{x}}\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x})$ denotes the Jacobian of $\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x})$.
Here are a few ways to Score-Matching:Eplicit score-matching,Denoising score-matching and Sliced score-matching.

- **Eplicit score-matching.** The kernel density estimation method is used to estimate the distribution directly, and then the loss function is constructed for

approximation.

$$q(\mathbf{x}) = \frac{1}{M}\sum_{m=1}^{M}\frac{1}{h}K\left(\frac{\mathbf{x}-\mathbf{x}_m}{h}\right) \tag{5}$$

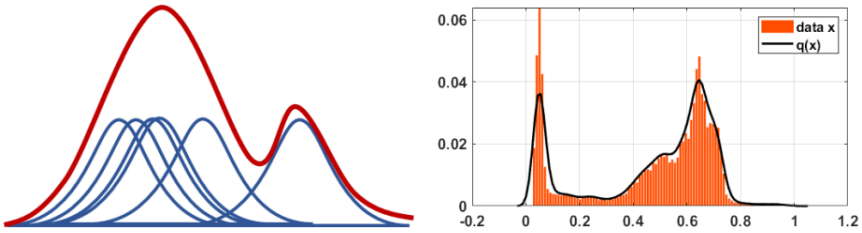


Figure 5: Illustration of kernel density estimation

$$\begin{aligned} J_{\mathrm{ESM}}(\boldsymbol{\theta}) &\overset{\text{def}}{=} \mathbb{E}_{q(\mathbf{x})}\left\|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}) - \nabla_{\mathbf{x}}\log q(\mathbf{x})\right\|^2 \\ &= \int \left\|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}) - \nabla_{\mathbf{x}}\log q(\mathbf{x})\right\|^2 \left[\frac{1}{M}\sum_{m=1}^{M}\frac{1}{h}K\left(\frac{\mathbf{x}-\mathbf{x}_m}{h}\right)\right] d\mathbf{x} \\ &= \frac{1}{M}\sum_{m=1}^{M}\int \left\|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}) - \nabla_{\mathbf{x}}\log q(\mathbf{x})\right\|^2 \frac{1}{h}K\left(\frac{\mathbf{x}-\mathbf{x}_m}{h}\right) d\mathbf{x}. \end{aligned} \tag{6}$$

- **Sliced score-matching.** The high-dimensional score is projected into the low-dimensional space for contrast training, thus avoiding the Jacobian matrix calculation in eq.(4).

$$\mathbb{E}_{p_{\mathbf{v}}}\mathbb{E}_{p_{\text{data}}}\left[\mathbf{v}^{\top}\nabla_{\mathbf{x}}\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x})\mathbf{v} + \frac{1}{2}\left\|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x})\right\|_2^2\right] \tag{7}$$

- **Denoising score-matching.** Denoising score-matching bypassing Jancobian's calculations, DSM's loss function is

$$\frac{1}{2}\mathbb{E}_{q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})p_{\text{data}}(\mathbf{x})}\left[\left\|\mathbf{s}_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}) - \nabla_{\tilde{\mathbf{x}}}\log q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x})\right\|_2^2\right] \tag{8}$$

Where, x is the random variable before adding noise, and $\tilde{\mathbf{x}}$is the random variable after adding noise, because the author finds that, If the score is approximated directly, the score estimate in the low data density area is inaccurate, so the method of adding noise is used for a more accurate estimate of score, now prove the validity of $J_{DSM}(\theta)$, starting with the noise-added eq(3):

$$\begin{aligned} J_{ESM}(\theta) &= E_{q(\tilde{x})}\left[\frac{1}{2}\left\|s_\theta(\tilde{x}) - \nabla_{\tilde{x}}\log q(\tilde{x})\right\|^2\right] \\ &= E_q(\tilde{x})\left[\frac{1}{2}\left\|s_\theta(\tilde{x})\right\|^2 - s_\theta(x)^{\top}\nabla_{\tilde{x}}\log q_{(x)} + \frac{1}{2}\left\|\nabla_x \log q(\tilde{x})\right\|^2\right] \\ &= E_q(\tilde{x})\left[\frac{1}{2}\left\|s_\theta(\tilde{x})\right\|^2 - s_\theta(x)^{\top}\nabla_{\tilde{x}}\log q(\tilde{x})\right] + C_1 \end{aligned} \tag{9}$$

Now calculate the second term, there is

$$\begin{aligned} E_{q(x)}\left[s_\theta(\tilde{x})^\top \nabla_{\tilde{x}} \log q(\tilde{x})\right] &= \int s_\theta(\tilde{x})^\top \nabla_{\tilde{x}} \log q(\tilde{x}) q(\tilde{x}) d\tilde{x} \\ &= \int s_\theta(\tilde{x})^\top \nabla_{\tilde{x}} q(\tilde{x}) \frac{q(\tilde{x})}{q(\tilde{x})} d\tilde{x} \\ &= \int s_\theta(\tilde{x})^\top \nabla_{\tilde{x}} q(\tilde{x}) d\tilde{x} \end{aligned} \tag{10}$$

We know $q(\tilde{x}) = \int q(x) q(\tilde{x} \mid x) dx$,take into the above formula, there is

$$\begin{aligned} \int s_\theta(\tilde{x})^\top \nabla_{\tilde{x}} q(\tilde{x}) d\tilde{x} &= \int s_\theta(\tilde{x})^\top \left(\nabla_{\tilde{x}} \int q(\tilde{x} \mid x) q(x) dx \right) d\tilde{x} \\ &= \iint s_\theta(\tilde{x})^\top q(x) \nabla_{\tilde{x}} q(\tilde{x}|x) \frac{q(\tilde{x} \mid x)}{q(\tilde{x} \mid x)} dx d\tilde{x} \\ &= \iint s_\theta(\tilde{x})^\top q(\tilde{x} \mid x) q(x) \nabla_{\tilde{x}} \log q(\tilde{x} \mid x) dx d\tilde{x} \\ &= \iint s_\theta(\tilde{x})^\top \nabla_{\tilde{x}} \log q(\tilde{x} \mid x) q(\tilde{x}, x) dx d\tilde{x} \\ &= E_{q(\tilde{x},x)}\left[s_\theta(\tilde{x})^\top \nabla_{\tilde{x}} \log q(\tilde{x} \mid x)\right] \end{aligned} \tag{11}$$

Substitute the second computed term into the original expression

$$\begin{aligned} &J_{ESM}(\theta) = E_{q(\tilde{x})}\left[\frac{1}{2}\|s_\theta(\tilde{x})\|^2\right] - E_{q(x,\tilde{x})}\left[s_\theta(\tilde{x})^\top \nabla x \log(\tilde{x}|x)\right] + C_1 \\ &= \bar{E}_q(\tilde{x}, x)\left[\frac{1}{2}\left\|s_\theta(\tilde{x})\right\|^2 - s_\theta(\tilde{x})^\top \nabla_{\tilde{x}} \log q(\tilde{x} \mid x)\right] + C_1 \\ &= E_{q(\tilde{x},x)}\left[\frac{1}{2}\left\|s_\theta(\tilde{x}) - \nabla_{\tilde{x}} \log q(\tilde{x} \mid x)\right\|^2\right] + C_1 + C_2 \\ &= J_{DSM}(\theta) + C \end{aligned} \tag{12}$$

So we verify the validity of DSM's objective function.

Several score-matching effects are shown in Fig(6).

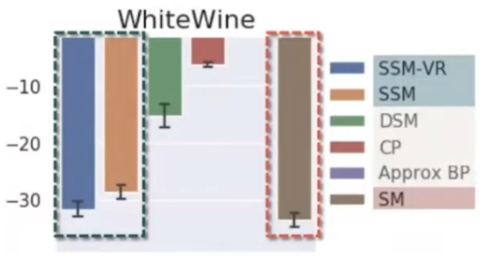


Figure 6: score-matching effect

1.3 Langevin Dynamcs

Langevin dynamics is a formula to describe the Brownian motion of molecules in a potential field, which can be expressed as

$$\frac{d\mathbf{x}}{dt} = -\frac{1}{\lambda}\nabla_{\mathbf{x}}U(\mathbf{x}) + \frac{\sigma}{\lambda}\mathbf{z} \tag{13}$$

Discretize it,and get

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \tau\nabla_{\mathbf{x}}U\left(\mathbf{x}_t\right) + \sigma\tau\mathbf{z}_t \tag{14}$$

The Fokker-Plank formula describes the steady state of Langevin dynamics as the Boltzmann distribution

$$p(\mathbf{x}) = \frac{1}{Z}\exp\{-U(\mathbf{x})\} \tag{15}$$

The score of Boltzmann distribution is the gradient of the potential field

$$\nabla_{\mathbf{x}}\log p(\mathbf{x}) = \nabla_{\mathbf{x}}\{-U(\mathbf{x}) - \log Z\} = -\nabla_{\mathbf{x}}U(\mathbf{x}) \tag{16}$$

By assigning some weights in the formula through the Fokker-Plank formula, Langevin dynamics can be written as

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \tau\nabla_{\mathbf{x}}\log p\left(\mathbf{x}_t\right) + \sqrt{2\tau}\mathbf{z}_t \tag{17}$$

Fig(7)is a toy example of Langevin dynamics.

Example. Consider a Gaussian mixture $p(x) = \pi_1\mathcal{N}(x\,|\,\mu_1,\sigma_1^2) + \pi_2\mathcal{N}(x\,|\,\mu_2,\sigma_2^2)$. We can numerically calculate $\nabla_x \log p(x)$. For demonstration, we choose $\pi_1 = 0.6$. $\mu_1 = 2$, $\sigma_1 = 0.5$, $\pi_2 = 0.4$, $\mu_2 = -2$, $\sigma_2 = 0.2$. Suppose we initialize $M = 10000$ uniformly distributed samples $x_0 \sim \text{Uniform}[-3, 3]$. We run Langevin updates for $t = 100$ steps. The histograms of generated samples are shown in the figures below.

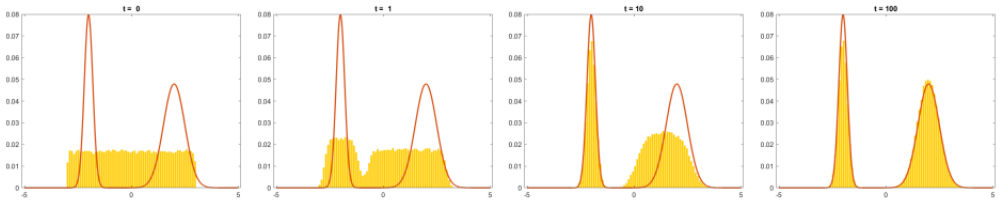


Figure 7: Example of Langevin dynamics

Proposition 1.2. Langevin dynamics is stochastic gradient descent. Without the noise term, Langevin dynamics is gradient descent.

1.4 Challenges of score-based generative modling

Using score-matching to estimate score and using Langevin dynamics to sample is a natural idea, but it doesn't produce any results. There are three main Challenges.

1.4.1 The manifold hypothesis

Hypothesis 1.3. The manifold hypothesis states that data in the real world tend to concentrate on low dimensional manifolds embedded in a high dimensional space (a.k.a., the ambient space).

Because of the manifold hypothesis, the gradient in ambient space is not well defined, and there will be multiple scores corresponding to the same point. A simple solution is to add noise to the original distribution to break the manifold hypothesis, and use SSM to estimate the score of the same distribution (the difference is whether noise is added or not). The result see Fig(8).

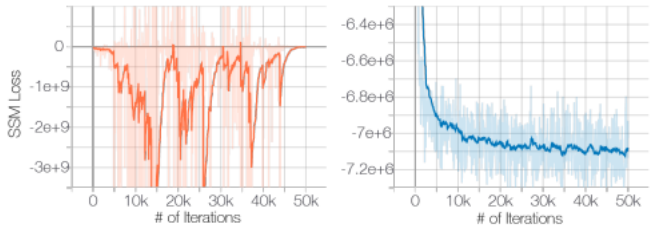


Figure 8: The manifold hypothesis

1.4.2 Inaccurate score estimation with score matching

Think about the loss function $\frac{1}{2}\mathbb{E}_{p_{\text{data}}}\left[\|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}) - \nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})\|_2^2\right]$,The probability $p_{\text{data}}$is used to obtain the expectation. In the real distribution space, $p_{\text{data}}$of most regions is 0. Therefore, for regions with low data density, we cannot accurately estimate the score. see Fig(9), This problem can be solved by adding disturbance noise to fill the probability space. see Fig(10).

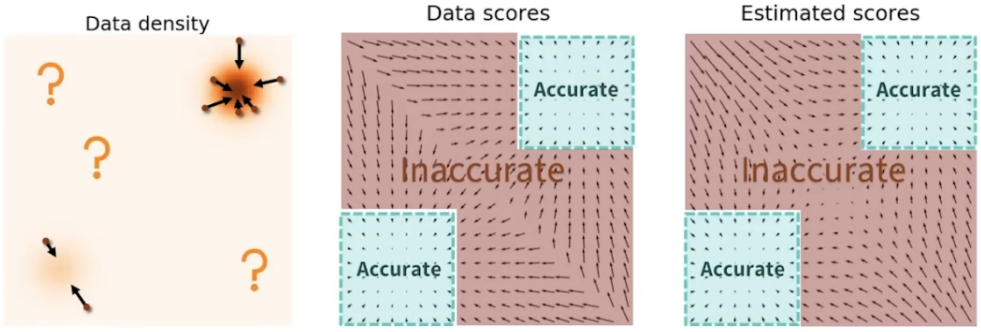


Figure 9: Inaccurate score estimation

1.4.3 Slow mixing of Langevin dynamics

When two modes of the data distribution are separated by low density regions, Langevin dynamics will not be able to correctly recover the relative weights of these two modes

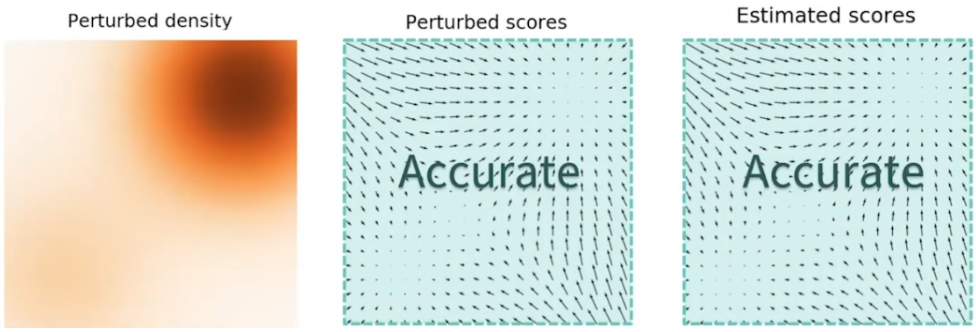


Figure 10: Perturbed score estimation

in reasonable time, and therefore might not converge to the true distribution. Consider a simple example $p_{\text{data}}(\mathbf{x}) = \pi p_1(\mathbf{x}) + (1-\pi)p_2(\mathbf{x})$, where $p_1(x)$ and $p_2(x)$ are two p.d.f , but the support sets do not intersect, in $p_1(x)$'s support set,$\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x}) = \nabla_{\mathbf{x}}(\log \pi + \log p_1(\mathbf{x})) = \nabla_{\mathbf{x}} \log p_1(\mathbf{x})$, and in $p_2(x)$'s support set,$\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x}) = \nabla_{\mathbf{x}}(\log(1-\pi) + \log p_2(\mathbf{x})) = \nabla_{\mathbf{x}} \log p_2(\mathbf{x})$, It can be seen that Langevin dynamics cannot capture the real distribution weight when the support sets are disjoint. The solution is to add noise to the original distribution to make the support sets intersect, so as to estimate the score more accurately.

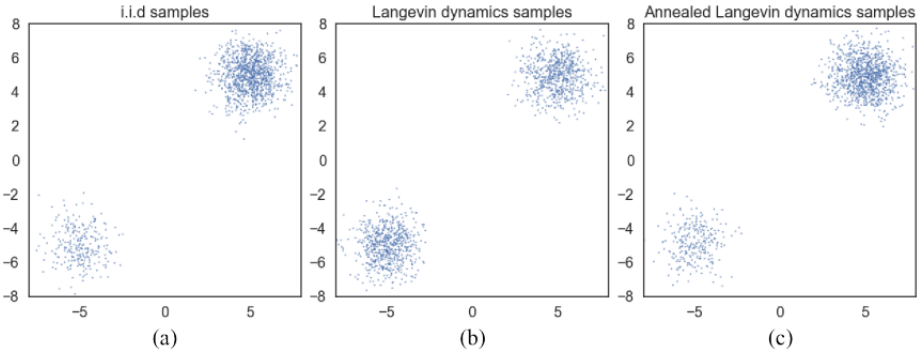


Figure 11: Slow mixing of Langevin dynamics

1.5 Noise Conditional Score Networks

The author proposes a new score matching method, Noise Conditional Score Networks (NCSNs), to solve the above three challenges. The solution is also clear, that is, to add multiple levels of noise to the same distribution, so that the minimum disturbance distribution can be close to the original distribution. The maximum disturbance distribution enabled us to accurately estimate the score, and then annealed Langevin dynamics was used to sample the original distribution according to the score estimated by NCSN, see Fig(12).

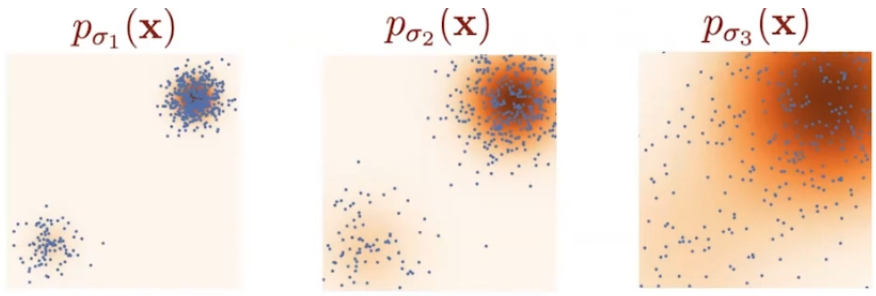


Figure 12: Diagram of annealed Langevin dynamics

Algorithm 1 Annealed Langevin dynamics.

Require: $\{\sigma_i\}_{i=1}^{L}, \epsilon, T$.
1: Initialize $\tilde{\mathbf{x}}_0$
2: **for** $i \leftarrow 1$ to L **do**
3: $\quad \alpha_i \leftarrow \epsilon \cdot \sigma_i^2/\sigma_L^2$ ▷ α_i is the step size.
4: $\quad$ **for** $t \leftarrow 1$ to T **do**
5: $\quad\quad$ Draw $\mathbf{z}_t \sim \mathcal{N}(0, I)$
6: $\quad\quad \tilde{\mathbf{x}}_t \leftarrow \tilde{\mathbf{x}}_{t-1} + \frac{\alpha_i}{2}\mathbf{s}_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}_{t-1}, \sigma_i) + \sqrt{\alpha_i}\,\mathbf{z}_t$
7: $\quad$ **end for**
8: $\quad \tilde{\mathbf{x}}_0 \leftarrow \tilde{\mathbf{x}}_T$
9: **end for**
return $\tilde{\mathbf{x}}_T$

(a) Annealed Langevin dynamics

Model	Inception	FID
CIFAR-10 Unconditional		
PixelCNN [59]	4.60	65.93
PixelIQN [42]	5.29	49.46
EBM [12]	6.02	40.58
WGAN-GP [18]	7.86 ± .07	36.4
MoLM [45]	7.90 ± .10	**18.9**
SNGAN [36]	8.22 ± .05	21.7
ProgressiveGAN [25]	8.80 ± .05	-
NCSN (Ours)	**8.87** ± .12	25.32
CIFAR-10 Conditional		
EBM [12]	8.30	37.9
SNGAN [36]	8.60 ± .08	25.5
BigGAN [6]	**9.22**	**14.73**

(b) Inception and FID scores for CIFAR-10

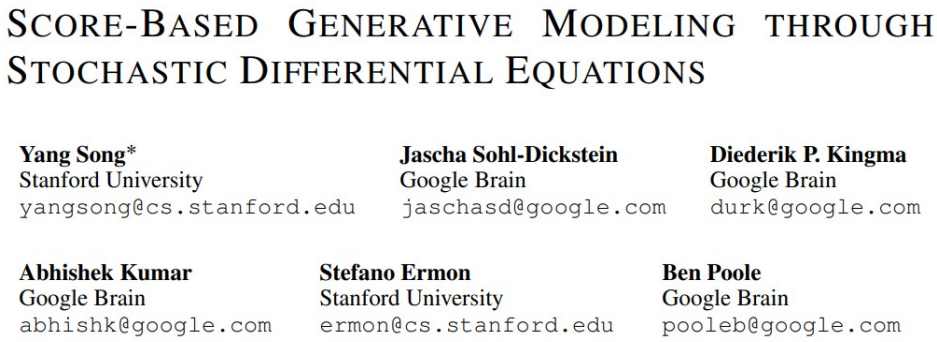
The objective function of NCSN is

$$\mathcal{L}\left(\boldsymbol{\theta}; \{\sigma_i\}_{i=1}^{L}\right) \triangleq \frac{1}{N}\sum_{i=1}^{N} \overbrace{\lambda(\sigma_i)}^{\text{weight}} \underbrace{\mathbb{E}_{p_{\sigma_i}(\mathrm{x})}\left[\left\|\nabla_{\mathrm{x}} \log p_{\sigma_i}(\mathrm{x}) - s_\theta(\mathrm{x}, \sigma_i)\right\|_2^2\right]}_{\text{Score matching loss}} \tag{18}$$

$\sigma_i$is to fix the weights of different disturbance distributions the same, Sampling algorithm see Fig(13a).In the original paper, the step size selection strategy is to make the signal-to-noise ratio $\frac{\alpha_i \mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}, \sigma_i)}{2\sqrt{\alpha_i}\mathbf{z}}$ become constant.The experimental results are shown in Fig(13b).

2 Score-basd Model

Score-Based genrative model through SDE,see Fig(14).



SCORE-BASED GENERATIVE MODELING THROUGH STOCHASTIC DIFFERENTIAL EQUATIONS

Yang Song*
Stanford University
yangsong@cs.stanford.edu

Jascha Sohl-Dickstein
Google Brain
jaschasd@google.com

Diederik P. Kingma
Google Brain
durk@google.com

Abhishek Kumar
Google Brain
abhishk@google.com

Stefano Ermon
Stanford University
ermon@cs.stanford.edu

Ben Poole
Google Brain
pooleb@google.com

Figure 14: paper of Score-based SDE

2.1 Use SDE to represent the diffusion model

The diffusion process or reverse diffusion process we describe is a discrete Markov process. Using SDE to describe a continuous stochastic process can help us analyze it better in random process theory.

2.1.1 Perturbing data with SDEs

The diffusion process has been described by equations in statistical physics and thermodynamics, i.e. Ito SDE:

$$\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w} \tag{19}$$

where $\mathbf{f}(\mathbf{x}, t)$is drift coefficient,$g(t)$is the diffusion coefficient, $w$is the standard Wiener process, that is, $w_t \sim N(0, tI)$, discretization of this SDE can obtain the discrete iterative formula:

$$\mathbf{x}_{t+\Delta t} - \mathbf{x}_t = \mathbf{f}(\mathbf{x}, t)\Delta t + g(t)\sqrt{\Delta t}z \tag{20}$$

2.1.2 GENERATING SAMPLES BY REVERSING THE SDE

Different from ODE, the inverse process of SDE cannot be solved simply because of the existence of random terms. Karmogorov's inverse equation shows that the diffusion process has its inverse process. Aderson deduced the inverse process of Ito SDE in a paper in 1982, which is also a diffusion process

$$\mathrm{d}\mathbf{x} = \left[\mathbf{f}(\mathbf{x}, t) - g(t)^2\nabla_{\mathbf{x}} \log p_t(\mathbf{x})\right] \mathrm{d}t + g(t)\mathrm{d}\overline{\mathbf{w}} \tag{21}$$

where $\overline{\mathbf{w}}$ is a standard Wiener process when time flows backwards from T to 0, and dt is an infinitesimal negative timestep.The unknown part is $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$, which is the score of the target distribution. If we get the score of the distribution, we can sample the target distribution from the initial distribution.

Now to prove that the inverse SDE of Ito SDE is also a diffusion process and is the SDE indicated by eq.(21), consider the discretized Ito SDE to find its inverse conditional probability:

$$\begin{aligned}
p\left(x_t \mid x_{t+\Delta t}\right) &= \frac{p\left(x_{t+\Delta t} \mid x_t\right) p\left(x_t\right)}{p\left(x_{t+\Delta t}\right)} \\
&= p\left(x_{t+\Delta t} \mid x_t\right) \exp\left(\log p\left(x_t\right) - \log p\left(x_{t+\Delta t}\right)\right) \\
&\approx p\left(x_{t+\Delta t} \mid x_t\right) \exp\left\{-\left(x_{t+\Delta t} - x_t\right) \nabla_{x_t} \log p\left(x_t\right) - \Delta t \frac{\partial}{\partial t} \log p\left(x_t\right)\right\} \\
&\propto \exp\left\{-\frac{\left\|x_{t+\Delta t} - x_t - f\left(x_t, t\right) \Delta t\right\|_2^2}{2 g^2(t) \Delta t} - \left(x_{t+\Delta t} - x_t\right) \nabla_{x_t} \log p\left(x_t\right) - \Delta t \frac{\partial}{\partial t} \log p\left(x_t\right)\right\} \\
&= \exp\left\{-\frac{1}{2 g^2(t) \Delta t}\left\|\left(x_{t+\Delta t} - x_t\right) - \left(f\left(x_t, t\right) - g^2(t) \nabla_{x_t} \log p\left(x_t\right)\right) \Delta t\right\|_2^2\right. \\
&\qquad \left. -\Delta t \frac{\partial}{\partial t} \log p\left(x_t\right) - \frac{f^2\left(x_t, t\right) \Delta t}{2 g^2(t)} + \frac{\left(f\left(x_t, t\right) - g^2(t) \nabla_{x_t} \log p\left(x_t\right)\right)^2 \Delta t}{2 g^2(t)}\right\} \\
&\overset{\Delta t \to 0}{=} \exp\left\{-\frac{1}{2 g^2(t+\Delta t) \Delta t} \| \left(x_{t+\Delta t} - x_t\right) - \left(f\left(x_{t+\Delta t}, t+\Delta t\right)\right.\right. \\
&\qquad \left.\left. -g^2(t+\Delta t) \nabla_{x_{t+\Delta t}} \log p\left(x_{t+\Delta t}\right)\right) \Delta t\|_2^2\right\}
\end{aligned} \tag{22}$$

Then the conditional probability of the reverse process is also a Gaussian distribution, and its mean and variance are:

$$\begin{gathered}
\mu = x_{t+\Delta t} - \left(f\left(x_{t+\Delta t}, t+\Delta t\right) - g^2(t+\Delta t) \nabla_{x_{t+\Delta t}} \log p\left(x_{t+\Delta t}\right)\right) \Delta t \\
\sigma^2 = g^2(t+\Delta t) \Delta t
\end{gathered} \tag{23}$$

So the discrete Reverse SDE could be written as

$$x_{t+\Delta t} - x_t = \left(f\left(x_{t+\Delta t}, t+\Delta t\right) - g^2(t+\Delta t) \nabla_{x_{t+\Delta t}} \log p\left(x_{t+\Delta t}\right)\right) \Delta t + g(t+\Delta t) \sqrt{\Delta t} z \tag{24}$$

So we can take it in a continuum, and get eq.(21).

2.1.3 Estimating Scores for the SDE

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_t \left\{\lambda(t) \mathbb{E}_{\mathbf{x}(0)} \mathbb{E}_{\mathbf{x}(t) \mid \mathbf{x}(0)} \left[\left\|\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}(t), t) - \nabla_{\mathbf{x}(t)} \log p_{0t}(\mathbf{x}(t) \mid \mathbf{x}(0))\right\|_2^2\right]\right\}. \tag{25}$$

To average the weights, set$\lambda \propto 1 / \mathbb{E}\left[\left\|\nabla_{\mathbf{x}(t)} \log p_{0t}(\mathbf{x}(t) \mid \mathbf{x}(0))\right\|_2^2\right]$.

2.2 VP SDE

2.2.1 DDPM

Diffusion:

$$\mathbf{x}_i = \sqrt{1-\beta_i} \mathbf{x}_{i-1} + \sqrt{\beta_i} \mathbf{z}_{i-1}, \quad i = 1, \cdots, N \tag{26}$$

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_\theta \tag{27}$$

Samping:

$$\begin{aligned} & q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0\right) = \mathcal{N}\left(\mathbf{x}_{t-1}; \tilde{\boldsymbol{\mu}}_t\left(\mathbf{x}_t, \mathbf{x}_0\right), \tilde{\beta}_t \mathbf{I}\right), \\ \text{where} \quad & \tilde{\boldsymbol{\mu}}_t\left(\mathbf{x}_t, \mathbf{x}_0\right) := \frac{\sqrt{\alpha_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0 + \frac{\sqrt{\alpha_t}\left(1-\bar{\alpha}_{t-1}\right)}{1-\bar{\alpha}_t}\mathbf{x}_t = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\varepsilon_\theta\right) \\ & \text{and} \quad \tilde{\beta}_t := \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t \approx \beta_t \end{aligned} \tag{28}$$

Loss function:

$$\min \sum_{t>0} \underbrace{D_{\mathrm{KL}}\left(q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0\right) \| p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)\right)}_{L_{t-1}} \tag{29}$$

$$L(\theta) := \mathbb{E}_{\mathbf{x}_0, \boldsymbol{\epsilon}}\left[\frac{\beta_t^2}{2\sigma_t^2 \alpha_t\left(1-\bar{\alpha}_t\right)}\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\left(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t\right)\right\|^2\right] \tag{30}$$

$$L_{\text{simple}}(\theta) := \mathbb{E}_{t, \mathbf{x}_0, \boldsymbol{\epsilon}}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\left(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t\right)\right\|^2\right] \tag{31}$$

2.2.2 The SDE of DDPM

To continuous a discrete Markov chain, we need to define a minimum time scale, let $\left\{\bar{\beta}_i = N\beta_i\right\}_{i=1}^N$, $N$is the number of diffusiion step,re-write eq.(26)

$$\mathbf{x}_i = \sqrt{1 - \frac{\bar{\beta}_i}{N}}\mathbf{x}_{i-1} + \sqrt{\frac{\bar{\beta}_i}{N}}\mathbf{z}_{i-1}, \quad i = 1, \cdots, N \tag{32}$$

In the limit of $N \to \infty, \left\{\bar{\beta}_i\right\}_{i=1}^N$ becomes a function $\beta(t)$ indexed by $t \in [0,1]$. Let $\beta(t+\Delta t)$correspond to $\bar{\beta}_i$, there is

$$\begin{aligned} \mathbf{x}(t+\Delta t) &= \sqrt{1-\beta(t+\Delta t)\Delta t}\mathbf{x}(t) + \sqrt{\beta(t+\Delta t)\Delta t}\mathbf{z}(t) \\ &\approx \mathbf{x}(t) - \frac{1}{2}\beta(t+\Delta t)\Delta t\mathbf{x}(t) + \sqrt{\beta(t+\Delta t)\Delta t}\mathbf{z}(t) \\ &\approx \mathbf{x}(t) - \frac{1}{2}\beta(t)\Delta t\mathbf{x}(t) + \sqrt{\beta(t)\Delta t}\mathbf{z}(t), \end{aligned} \tag{33}$$

in the limit of $\Delta t \to 0$,we can get VP SDE**:**

$$\mathrm{d}\mathbf{x} = -\frac{1}{2}\beta(t)\mathbf{x}\mathrm{d}t + \sqrt{\beta(t)}\mathrm{d}\mathbf{w} \tag{34}$$

Corresponding to eq.(19), the drift coefficient and diffusion coefficient corresponding to DDPM can be obtained:

$$\begin{cases} f(x,t) = -\frac{1}{2}\beta(t)\mathbf{x} \\ g(t) = \sqrt{\beta(t)}. \end{cases} \tag{35}$$

The corresponding sampling SDE is

$$d\mathbf{x} = \left[-\frac{1}{2}\beta(t)x - \beta(t)\nabla_x \log p_t(x)\right]dt + \sqrt{\beta(t)}d\bar{w} \tag{36}$$

If the SDE framework can accommodate DDPM, then Reverse SDE discretization will inevitably yield the same result as DDPM sampling. Now let's prove that Reverse SDE is equivalent to DDPM sampling.
Rewrite the sample of DDPM as

$$x_i = \frac{1}{\sqrt{1-\beta_{i+1}}}\left(x_{i+1} - \frac{\beta_{i+1}}{\sqrt{1-\bar{\alpha}_{i+1}}}\varepsilon_\theta\left(x_{i+1}, i+1\right)\right) + \sqrt{\beta_{i+1}}z_{i+1} \tag{37}$$

Find the score of DDPM:

$$\begin{aligned} s_\theta\left(x_t, t\right) &= \nabla_x \log P_t\left(x_t \mid x_0\right) \\ &= \nabla_x \log N\left(x_t; \sqrt{\bar{\alpha}_t}x_0, 1-\bar{\alpha}_t\right) \\ &= \nabla_x\left(-\frac{\left(x_t - \sqrt{\bar{\alpha}_t}x_0\right)^2}{2\left(1-\bar{\alpha}_t\right)}\right) \\ &= -\frac{x_t - \sqrt{\alpha_t}x_0}{1-\bar{\alpha}_t} \\ &= -\frac{x_t - \sqrt{\bar{\alpha}_t}\frac{1}{\sqrt{\bar{\alpha}_t}}\left(x_t - \sqrt{1-\alpha_t}\varepsilon_\theta\left(x_t, t\right)\right)}{1-\bar{\alpha}_t} \\ &= -\frac{1}{\sqrt{1-\bar{\alpha}_t}}\varepsilon_\theta\left(x_t, t\right) \end{aligned} \tag{38}$$

Take $s_\theta\left(x_t, t\right) = -\frac{1}{\sqrt{1-\bar{\alpha}_t}}\varepsilon_\theta\left(x_t, t\right)$ into DDPM sampling equation:

$$\begin{aligned} \mathbf{x}_i &= \frac{1}{\sqrt{1-\beta_{i+1}}}\left(\mathbf{x}_{i+1} + \beta_{i+1}\mathbf{s}_{\boldsymbol{\theta}^*}\left(\mathbf{x}_{i+1}, i+1\right)\right) + \sqrt{\beta_{i+1}}\mathbf{z}_{i+1} \\ &= \left(1 + \frac{1}{2}\beta_{i+1} + o\left(\beta_{i+1}\right)\right)\left(\mathbf{x}_{i+1} + \beta_{i+1}\mathbf{s}_{\boldsymbol{\theta}^*}\left(\mathbf{x}_{i+1}, i+1\right)\right) + \sqrt{\beta_{i+1}}\mathbf{z}_{i+1} \\ &\approx \left(1 + \frac{1}{2}\beta_{i+1}\right)\left(\mathbf{x}_{i+1} + \beta_{i+1}\mathbf{s}_{\boldsymbol{\theta}^*}\left(\mathbf{x}_{i+1}, i+1\right)\right) + \sqrt{\beta_{i+1}}\mathbf{z}_{i+1} \\ &= \left(1 + \frac{1}{2}\beta_{i+1}\right)\mathbf{x}_{i+1} + \beta_{i+1}\mathbf{s}_{\boldsymbol{\theta}^*}\left(\mathbf{x}_{i+1}, i+1\right) + \frac{1}{2}\beta_{i+1}^2\mathbf{s}_{\boldsymbol{\theta}^*}\left(\mathbf{x}_{i+1}, i+1\right) + \sqrt{\beta_{i+1}}\mathbf{z}_{i+1} \\ &\approx \left(1 + \frac{1}{2}\beta_{i+1}\right)\mathbf{x}_{i+1} + \beta_{i+1}\mathbf{s}_{\boldsymbol{\theta}^*}\left(\mathbf{x}_{i+1}, i+1\right) + \sqrt{\beta_{i+1}}\mathbf{z}_{i+1} \end{aligned} \tag{39}$$

Discretization Reverse VP SDE eq.(36) and then there is

$$x_{t+\Delta t} - x_t = -\frac{1}{2}\beta(t)x_t\Delta t - \beta(t)\nabla_x \log p_t(x)\Delta t + \sqrt{\beta(t)}\sqrt{|\Delta t|}z \tag{40}$$

Let $\Delta t = -1$,$t = i+1$, we can get eq.(39) form, so we prove that DDPM can be integrated into the framework of SDE, DDPM is also a kind of score-based model.In addition, the authors call the sampling method of DDPM *ancestral sampling.*

2.2.3 Variance Preserving

Applied Stochastic Differential Equations indicates that for a diffusion process, the mean and covariance matrix of its edge distribution can be expressed by the following ODE:

$$\begin{cases} \frac{dm}{dt} = E[f(x,t)] \\ \frac{d\Sigma}{dt} = E\left[f(x,t)(x-m)^\top\right] + E\left[(x-m)f^\top(x,t)\right] + E\left[g^2(t)\right] \end{cases} \tag{41}$$

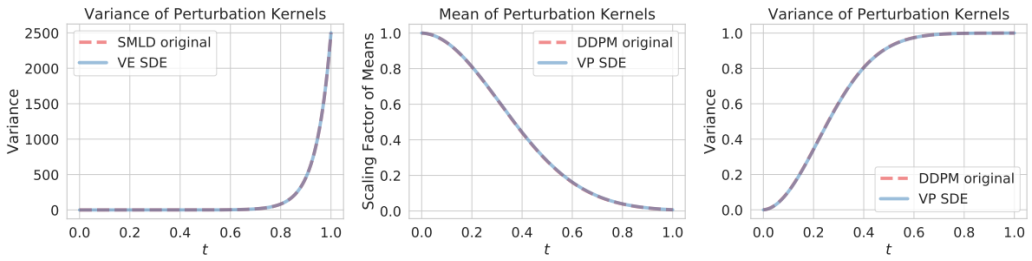
For VP SDE, the ODE of its covariance matrix is

$$\frac{\mathrm{d}\boldsymbol{\Sigma}_{\mathrm{VP}}(t)}{\mathrm{d}t} = \beta(t)\left(\mathbf{I} - \boldsymbol{\Sigma}_{\mathrm{VP}}(t)\right) \tag{42}$$

Solve the ODE, and get

$$\boldsymbol{\Sigma}_{\mathrm{VP}}(t) = \mathbf{I} + e^{\int_0^t -\beta(s)\mathrm{d}s}\left(\boldsymbol{\Sigma}_{\mathrm{VP}}(0) - \mathbf{I}\right) \tag{43}$$

The covariance matrix of VP SDE is bounded by $\boldsymbol{\Sigma}_{\mathrm{VP}}$and I. So it is called Variance Preserving SDE,see Fig(15).



(a) SMLD (b) DDPM (mean) (c) DDPM (variance)

Figure 15: Variance Preserving and Variance Exploding

2.3 VE SDE

SMLD can also be integrated into the SDE framework in the same way as DDPM.

2.3.1 SMLD

Diffusion：

$$\mathbf{x}_i = \mathbf{x}_{i-1} + \sqrt{\sigma_i^2 - \sigma_{i-1}^2}\mathbf{z}_{i-1}, \quad i = 1, \cdots, N \tag{44}$$

Sampling:

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \tau\nabla_{\mathbf{x}}\log p\left(\mathbf{x}_i\right) + \sqrt{2\tau}\mathbf{z}_i \tag{45}$$

2.3.2 The SDE of SMLD

Rewrite the corner label of eq.(44)

$$\mathbf{x}(t+\Delta t)=\mathbf{x}(t)+\sqrt{\sigma^2(t+\Delta t)-\sigma^2(t)}\mathbf{z}(t)\approx\mathbf{x}(t)+\sqrt{\frac{\mathrm{d}\left[\sigma^2(t)\right]}{\mathrm{d}t}\Delta t}\mathbf{z}(t) \tag{46}$$

Taking the limit of Δt, there is

$$\mathrm{d}\mathbf{x}=\sqrt{\frac{\mathrm{d}\left[\sigma^2(t)\right]}{\mathrm{d}t}}\,\mathrm{d}\mathbf{w} \tag{47}$$

Thus, the drift coefficient and diffusion coefficient of VE SDE are obtained as follows:

$$\begin{cases} f(x,t)=0 \\ g(t)=\sqrt{\frac{\mathrm{d}[\sigma^2(t)]}{\mathrm{d}t}} \end{cases} \tag{48}$$

So Reverse VE SDE can be written as

$$dx=-\nabla_x\log p_t(x)d\sigma^2(t)+\sqrt{\frac{d\sigma^2(t)}{dt}}d\bar{w} \tag{49}$$

Discretized Reverse VE SDE, and we can get

$$x_{t+\Delta t}-x_t=-\left[\sigma^2(t+\Delta t)-\sigma^2(t)\right]\nabla_x\log p_t(x)+\sqrt{\sigma^2(t+\Delta t)-\sigma^2(t)}z \tag{50}$$

It can be seen that this formula is not completely in the form of Langevin Dynamcs, and the diffusion of SMLD is not compatible with the sampling, which may be the reason why the effect of SMLD is worse than that of DDPM.
We can use the method in DDPM paper to infer the ancestor sampling process of SMLD from the perspective of maximum likelihood, and the corresponding sampling formula is as follows

$$\mathbf{x}_{i-1}=\mathbf{x}_i+\left(\sigma_i^2-\sigma_{i-1}^2\right)\mathbf{s}_{\boldsymbol{\theta}}*(\mathbf{x}_i,i)+\sqrt{\frac{\sigma_{i-1}^2\left(\sigma_i^2-\sigma_{i-1}^2\right)}{\sigma_i^2}}\mathbf{z}_i, i=1,2,\cdots,N \tag{51}$$

The experimental results in the paper show that both reverse diffusiion sampler and ancestor sampling are better for SMLD than Langevin dynamics sampling.

2.3.3 Variance Exploding

Use the same method as for VP SDE to list the ODE satisfied by the VE SDE covariance matrix

$$\frac{d\Sigma(t)}{dt}=\frac{d\sigma^2(t)}{dt}. \tag{52}$$

Solve the ODE and we can get

$$\Sigma(t)=\Sigma(0)+\left(\sigma^2(t)-\sigma^2(0)\right)I \tag{53}$$

Because it is usually necessary to add enough noise so that score can be accurately estimated, the covariance matrix of VE SDE is often unbounded, which is also related to the lack of weight in front of $x_0$in the process of SMLD diffusion. If we want to eventually spread to the Guass distribution, we must set a large enough variance for diffusion.

2.4 Prebabliity Flow ODE

Score-based models enable another numerical method for solving the reverse-time SDE. For all diffusion processes, there exists a corresponding *deterministic process* whose trajectories share the same marginal probability densities $\{p_t(\mathbf{x})\}_{t=0}^{T}$ as the SDE. This deterministic process satisfies an ODE:

$$\mathrm{d}\mathbf{x} = \left[\mathbf{f}(\mathbf{x}, t) - \frac{1}{2} g(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x})\right] \mathrm{d}t \tag{54}$$

We name the ODE in eq. (54) the probability flow ODE.
Now let's prove that the ODE corresponding to Ito SDE is eq.(54). Consider a more general Ito SDE:

$$\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\mathrm{d}t + \mathbf{G}(\mathbf{x}, t)\mathrm{d}\mathbf{w} \tag{55}$$

where $\mathbf{f}(\cdot, t) : \mathbb{R}^d \to \mathbb{R}^d$ and $\mathbf{G}(\cdot, t) : \mathbb{R}^d \to \mathbb{R}^{d\times d}$. The marginal probability density $p_t(\mathbf{x}(t))$ evolves according to Kolmogorov's forward equation (Fokker-Planck equation):

$$\frac{\partial p(\mathbf{x}, t)}{\partial t} = -\sum_{i=1}^{N} \frac{\partial}{\partial x_i} \left[\mu_i(\mathbf{x}, t) p(\mathbf{x}, t)\right] + \sum_{i=1}^{N}\sum_{j=1}^{N} \frac{\partial^2}{\partial x_i \partial x_j} \left[D_{ij}(\mathbf{x}, t) p(\mathbf{x}, t)\right] \tag{56}$$

Where $\mathbf{D} = \frac{1}{2}\boldsymbol{\sigma}\boldsymbol{\sigma}^{\top}$,i.e.

$$D_{ij}(\mathbf{x}, t) = \frac{1}{2} \sum_{k=1}^{M} \sigma_{ik}(\mathbf{x}, t)\sigma_{jk}(\mathbf{x}, t) \tag{57}$$

Substitute the diffusion coefficient and drift coefficient of Ito SDE into eq.(56)

$$\begin{aligned}
\frac{\partial p_t(\mathbf{x})}{\partial t} &= -\sum_{i=1}^{d} \frac{\partial}{\partial x_i} \left[f_i(\mathbf{x}, t) p_t(\mathbf{x})\right] + \frac{1}{2}\sum_{i=1}^{d}\sum_{j=1}^{d} \frac{\partial^2}{\partial x_i \partial x_j} \left[\sum_{k=1}^{d} G_{ik}(\mathbf{x}, t) G_{jk}(\mathbf{x}, t) p_t(\mathbf{x})\right] \\
&= -\sum_{i=1}^{d} \frac{\partial}{\partial x_i} \left[f_i(\mathbf{x}, t) p_t(\mathbf{x})\right] + \frac{1}{2}\sum_{i=1}^{d} \frac{\partial}{\partial x_i} \left[\sum_{j=1}^{d} \frac{\partial}{\partial x_j} \left[\sum_{k=1}^{d} G_{ik}(\mathbf{x}, t) G_{jk}(\mathbf{x}, t) p_t(\mathbf{x})\right]\right]
\end{aligned} \tag{58}$$

Note that

$$\begin{aligned}
&\sum_{j=1}^{d} \frac{\partial}{\partial x_j} \left[\sum_{k=1}^{d} G_{ik}(\mathbf{x},t) G_{jk}(\mathbf{x},t) p_t(\mathbf{x}) \right] \\
=&\sum_{j=1}^{d} \frac{\partial}{\partial x_j} \left[\sum_{k=1}^{d} G_{ik}(\mathbf{x},t) G_{jk}(\mathbf{x},t) \right] p_t(\mathbf{x}) + \sum_{j=1}^{d} \sum_{k=1}^{d} G_{ik}(\mathbf{x},t) G_{jk}(\mathbf{x},t) p_t(\mathbf{x}) \frac{\partial}{\partial x_j} \log p_t(\mathbf{x}) \\
=&p_t(\mathbf{x}) \nabla \cdot \left[\mathbf{G}(\mathbf{x},t) \mathbf{G}(\mathbf{x},t)^\top \right] + p_t(\mathbf{x}) \mathbf{G}(\mathbf{x},t) \mathbf{G}(\mathbf{x},t)^\top \nabla_{\mathbf{x}} \log p_t(\mathbf{x})
\end{aligned} \tag{59}$$

based on which we can continue the rewriting of eq. (58) to obtain

$$\begin{aligned}
\frac{\partial p_t(\mathbf{x})}{\partial t} =& -\sum_{i=1}^{d} \frac{\partial}{\partial x_i} \left[f_i(\mathbf{x},t) p_t(\mathbf{x}) \right] + \frac{1}{2} \sum_{i=1}^{d} \frac{\partial}{\partial x_i} \left[\sum_{j=1}^{d} \frac{\partial}{\partial x_j} \left[\sum_{k=1}^{d} G_{ik}(\mathbf{x},t) G_{jk}(\mathbf{x},t) p_t(\mathbf{x}) \right] \right] \\
=& -\sum_{i=1}^{d} \frac{\partial}{\partial x_i} \left[f_i(\mathbf{x},t) p_t(\mathbf{x}) \right] \\
&+ \frac{1}{2} \sum_{i=1}^{d} \frac{\partial}{\partial x_i} \left[p_t(\mathbf{x}) \nabla \cdot \left[\mathbf{G}(\mathbf{x},t) \mathbf{G}(\mathbf{x},t)^\top \right] + p_t(\mathbf{x}) \mathbf{G}(\mathbf{x},t) \mathbf{G}(\mathbf{x},t)^\top \nabla_{\mathbf{x}} \log p_t(\mathbf{x}) \right] \\
=& -\sum_{i=1}^{d} \frac{\partial}{\partial x_i} \left\{ f_i(\mathbf{x},t) p_t(\mathbf{x}) \right. \\
&\left. - \frac{1}{2} \left[\nabla \cdot \left[\mathbf{G}(\mathbf{x},t) \mathbf{G}(\mathbf{x},t)^\top \right] + \mathbf{G}(\mathbf{x},t) \mathbf{G}(\mathbf{x},t)^\top \nabla_{\mathbf{x}} \log p_t(\mathbf{x}) \right] p_t(\mathbf{x}) \right\} \\
=& -\sum_{i=1}^{d} \frac{\partial}{\partial x_i} \left[\tilde{f}_i(\mathbf{x},t) p_t(\mathbf{x}) \right]
\end{aligned} \tag{60}$$

where we define

$$\tilde{\mathbf{f}}(\mathbf{x},t) := \mathbf{f}(\mathbf{x},t) - \frac{1}{2} \nabla \cdot \left[\mathbf{G}(\mathbf{x},t) \mathbf{G}(\mathbf{x},t)^\top \right] - \frac{1}{2} \mathbf{G}(\mathbf{x},t) \mathbf{G}(\mathbf{x},t)^\top \nabla_{\mathbf{x}} \log p_t(\mathbf{x}) \tag{61}$$

Then compare Kolmogorov's forward equation,and the diffusion coefficient is zero:

$$\mathrm{d}\mathbf{x} = \tilde{\mathbf{f}}(\mathbf{x},t)\mathrm{d}t + \tilde{\mathbf{G}}(\mathbf{x},t)\mathrm{d}\mathbf{w} = \tilde{\mathbf{f}}(\mathbf{x},t)\mathrm{d}t \tag{62}$$

In this way, the ODE corresponding to Ito SDE is obtained, and they have the same edge probability density. Since the ODE has no randomness, we can sample from the ODE by any numerical solution. For example, the numerical simulation of the ODE can be performed simply by iterative algorithm, that is, the PF ODE is discretized:

$$\mathbf{x}_i = \mathbf{x}_{i+1} - \mathbf{f}_{i+1}\left(\mathbf{x}_{i+1}\right) + \frac{1}{2} \mathbf{G}_{i+1} \mathbf{G}_{i+1}^\top \mathbf{s}_{\theta^*}\left(\mathbf{x}_{i+1}, i+1\right), \quad i = 0, 1, \cdots, N-1 \tag{63}$$

Corresponding discrete PF ODE of VE SDE and VP SDE are as follows:

$$\mathbf{x}_i = \mathbf{x}_{i+1} + \frac{1}{2} \left(\sigma_{i+1}^2 - \sigma_i^2\right) \mathbf{s}_{\boldsymbol{\theta}} * \left(\mathbf{x}_{i+1}, \sigma_{i+1}\right), \quad i = 0, 1, \cdots, N-1 \tag{64}$$

$$\mathbf{x}_i = \left(2 - \sqrt{1-\beta_{i+1}}\right)\mathbf{x}_{i+1} + \frac{1}{2}\beta_{i+1}\mathbf{s}_{\theta^*}\left(\mathbf{x}_{i+1}, i+1\right), \quad i = 0, 1, \cdots, N-1 \tag{65}$$

Using PF ODE, we can do Exact likelihood computation,Manipulating latent representations ,Uniquely identifiable encoding and Efficient sampling.

2.5 Sampling

We name these sampling methods (that are based on the discretization strategy in Reverse SDE) *reverse diffusion samplers.*Any SDE solver can be used to solve the problem. The author proposes a predictor corrector sampling algorithm. reverse SDE was used as predictor and annealed Langevin dynamics as corrector. see Fig(16). The

Algorithm 2 PC sampling (VE SDE)

$\mathbf{x}_N \sim \mathcal{N}(\mathbf{0}, \sigma_{\max}^2\mathbf{I})$
for $i = N-1$ **to** 0 **do**
  $\mathbf{x}_i' \leftarrow \mathbf{x}_{i+1} + (\sigma_{i+1}^2 - \sigma_i^2)\mathbf{s}_{\boldsymbol{\theta}*}(\mathbf{x}_{i+1}, \sigma_{i+1})$
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathbf{x}_i \leftarrow \mathbf{x}_i' + \sqrt{\sigma_{i+1}^2 - \sigma_i^2}\mathbf{z}$
  for $j = 1$ **to** M **do**
    $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
    $\mathbf{x}_i \leftarrow \mathbf{x}_i + \epsilon_i \mathbf{s}_{\boldsymbol{\theta}*}(\mathbf{x}_i, \sigma_i) + \sqrt{2\epsilon_i}\mathbf{z}$
return $\mathbf{x}_0$

Algorithm 3 PC sampling (VP SDE)

$\mathbf{x}_N \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $i = N-1$ **to** 0 **do**
  $\mathbf{x}_i' \leftarrow (2 - \sqrt{1-\beta_{i+1}})\mathbf{x}_{i+1} + \beta_{i+1}\mathbf{s}_{\boldsymbol{\theta}*}(\mathbf{x}_{i+1}, i+1)$
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathbf{x}_i \leftarrow \mathbf{x}_i' + \sqrt{\beta_{i+1}}\mathbf{z}$ — Predictor
  for $j = 1$ **to** M **do** — Corrector
    $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
    $\mathbf{x}_i \leftarrow \mathbf{x}_i + \epsilon_i \mathbf{s}_{\boldsymbol{\theta}*}(\mathbf{x}_i, i) + \sqrt{2\epsilon_i}\mathbf{z}$
return $\mathbf{x}_0$

Figure 16: Predictor corrector sampling

experimental results are shown in Fig(17).

	Variance Exploding SDE (SMLD)				Variance Preserving SDE (DDPM)			
FID↓ Sampler / Predictor	P1000	P2000	C2000	PC1000	P1000	P2000	C2000	PC1000
ancestral sampling	4.98 ± .06	4.92 ± .02		**3.62 ± .03**	3.24 ± .02	**3.11 ± .03**		3.21 ± .02
reverse diffusion	4.79 ± .07	4.72 ± .07	20.43 ± .07	**3.60 ± .02**	3.21 ± .02	**3.10 ± .03**	19.06 ± .06	3.18 ± .01
probability flow	15.41 ± .15	12.87 ± .09		**3.51 ± .04**	3.59 ± .04	3.25 ± .04		**3.06 ± .03**

Figure 17: SDE experiment results