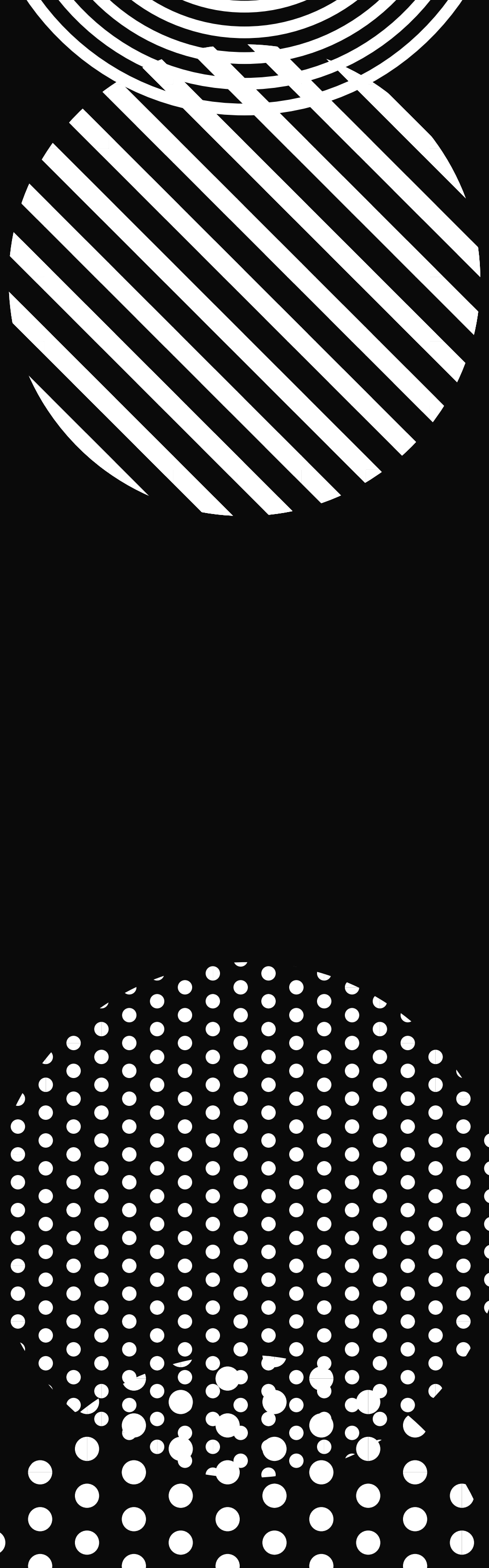

SDE → DIFFUSION MODEL

$p(x)$ → TARGET

$p_\theta(x)$ Our approximation of $p(x)$

$p_\theta(x) = \frac{e^{-f_\theta(x)}}{Z_\theta}$ => good way to parametrize because $-f_\theta(x)$ can be anything to be real negative. Z_θ normalizes such that $\int p_\theta(x) = 1$

Example:

$$\mathcal{N}(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$f_\theta(x)$ = how dense the probability is around x.

Calculating Z_θ requires us to integrate over the entire space of x => TOO EXPENSIVE

So we predict the score

$$\nabla_x \log p_\theta(x) = \nabla_x \log \frac{e^{-f_\theta(x)}}{Z_\theta} =$$

$$= \nabla_x \log e^{-f_\theta(x)} - \nabla_x \log Z_\theta = -\nabla_x f_\theta(x)$$

$$= s_\theta(x)$$

SCORE MATCHING:

$$\frac{1}{2} \mathbb{E}_{p(x)} \left[\| \nabla_x \log p(x) - s_\theta(x) \|_2^2 \right]$$

↙

score function: where to move to increase the probability of a data point

But WHO IS $\nabla_x \log p(x)$?

$$\frac{1}{2} \mathbb{E}_{p(x)} \left[\| \nabla_x \log p(x) - s_\theta(x) \|_2^2 \right] =$$

$$= \frac{1}{2} \int p(x) \left(\nabla_x \log p(x) - s_\theta(x) \right)^2 dx =$$

$$= \frac{1}{2} \int p(x) \nabla_x^2 \log p(x) + p(x) s_\theta^2(x) - 2 \nabla_x \log p(x) \cdot s_\theta(x) dx =$$

$$= \frac{1}{2} \int p(x) \nabla_x^2 \log p(x) dx + \frac{1}{2} \int p(x) s_\theta^2(x) dx - \int p_x \nabla_x \log p(x) s_\theta(x) dx$$

We know that: $\nabla_x \log p(x) = \frac{\nabla_x p(x)}{p(x)}$

So we have that:

$$\int p_x \nabla_x \log p(x)\, s_\theta(x)\, dx = \int p_x \frac{\nabla_x p(x)}{p(x)} s_\theta(x)\, dx =$$

$$= \int \nabla_x p(x)\, s_\theta(x)\, dx = \text{integration by parts}$$

$$\int_a^b dv\, u = uv\Big|_a^b - \int v\, du \qquad \text{so we have:}$$

$$= p(x) s_\theta(x)\Big|_{-\infty}^{+\infty} - \int p_x \nabla_x s_\theta(x)\, dx =$$

$p(x) s_\theta(x) \to 0$ because $x \to +\infty$ $p(x) \to 0$

$\Rightarrow$ Going back we have that:

$$\frac{1}{2} E_{p(x)}\left[\| \nabla_x \log p(x) - s_\theta(x) \|_2^2 \right] =$$

$$= \frac{1}{2} \int p(x) \nabla_x^2 \log p(x)\, dx + \frac{1}{2} \int p(x) s_\theta^2(x)\, dx -$$

$$+ \int p(x) \nabla_x s_\theta(x)\, dx$$

$\nabla_x^2 \log p(x)$: there is no θ so it is a constant and for optimization problem it is 0

$$= \frac{1}{2} p(x) s_\theta^2(x)\, dx + \int p(x) \nabla_x s_\theta(x)\, dx$$

and $\mathbb{E}_{p(x)} = \int p(x)\, dx$

$$= \frac{1}{2}\mathbb{E}_{p(x)}[s_\theta^2(x)] + \mathbb{E}_{p(x)}[\nabla_x s_\theta(x)]$$

We see now even if we don't have $p(x)$

$$\frac{1}{2}\mathbb{E}_{p(x)}[\| \nabla_x \log p(x) - s_\theta(x) \|_2^2] =$$

It doesn't depend on $p(x)$ ← $= \frac{1}{2}\mathbb{E}_{p(x)}[s_\theta^2(x)] + \mathbb{E}_{p(x)}[\nabla_x s_\theta(x)] + C$

We want to minimize it so it means we want $\nabla_x s_\theta(x) \to 0 \Rightarrow$ local maximum

while we first term $\mathbb{E}_{p(x)}[s_\theta^2(x)]$ is 0 at data points

PROBLEM: $\nabla_x s_\theta(x)$ Very expensive

It involves computing the Jacobian especially for images. This was been resolved in:

Sliced Score Matching: A Scalable Approach to Density and Score Estimation

Yang Song* Stanford University
Sahaj Garg* Stanford University
Jiaxin Shi Tsinghua University
Stefano Ermon Stanford University

2nd Problem : Data space coverage =>

=> model is good for data in the data space were he crowded ones but not in others.

To solve this problem we NOISE the images (Adding Gaussian noise)

$$\tilde{x} = x + \epsilon \qquad \epsilon \sim \mathcal{N}(0, \sigma^2 I)$$

$$p(x) = p_\sigma(\tilde{x}) \quad \text{& we solve}$$

problem 2.

How much noise depends on σ so:

$$\frac{1}{2} \mathbb{E}_{p_\sigma(x)} \left[\| \nabla_x \log p_\sigma(\tilde{x}) - s_\theta(\tilde{x}) \|_2^2 \right]$$

If σ is very large the space is no different

If σ is very small we have 2nd problem

$$\frac{1}{2} \mathbb{E}_{p_\sigma(\tilde{x})} \left[s_\theta^2(\tilde{x}) \right] + \mathbb{E}_{p_\sigma(x)} \left[\nabla_x s_\theta(\tilde{x}) \right]$$

How do we compute $\nabla_x s_\theta(\tilde{x})$?

Pascal Vincent 2010 saw a connection

Score Matching <=> Denoising Autoencoders

Train a regular AE but add noise

in the bottleneck => being able to

separate noise from data it was possible

that the model could learn important

features. So we start from

$$\frac{1}{2}\mathbb{E}_{p_\sigma(x)}\left[\| \nabla_x \log p_\sigma(x) - s_\theta(x)\|_2^2\right]$$

and our goal is:

$$\frac{1}{2}\mathbb{E}_{x\sim p(x),\tilde{x}\sim p_\sigma(\tilde{x}|x)}\left[\| s_\theta(\tilde{x}) - \nabla_{\tilde{x}} \log p_\sigma(\tilde{x}|x)\|_2^2\right]$$

So first

$$\frac{1}{2}\mathbb{E}_{p_\sigma(x)}\left[\| \nabla_x \log p_\sigma(x) - s_\theta(x)\|_2^2\right] =$$

$$= \frac{1}{2}\int p_\sigma(\tilde{x})\left(\nabla_x \log p_\sigma(x) - s_\theta(\tilde{x})\right)^2 d\tilde{x},$$

$$= \frac{1}{2}\int p_\sigma(\tilde{x})\nabla^2_{\tilde{x}} \log p_\sigma(\tilde{x})\, d\tilde{x} +$$

$$+ \frac{1}{2}\int p_\sigma(\tilde{x}) s^2_\theta(\tilde{x})\, d\tilde{x} +$$

$$- \int p_\theta(\tilde{x}) \nabla_{\tilde{x}} \log p_\theta(\tilde{x}) s_\theta(\tilde{x}) d\tilde{x}$$

$$\nabla_{\tilde{x}} \log p_\theta(\tilde{x}) = \frac{\nabla_{\tilde{x}} p_\theta(\tilde{x})}{p_\theta(\tilde{x})}$$

$$\downarrow$$

$$\int \cancel{p_\theta(\tilde{x})} \frac{\nabla_{\tilde{x}} p_\theta(\tilde{x})}{\cancel{p_\theta(\tilde{x})}} s_\theta(\tilde{x}) d\tilde{x}$$

Use marginalization and so:

$$p_\theta(\tilde{x}) = \int p(x) p_\theta(\tilde{x}|x) dx$$

$$= \int \nabla_{\tilde{x}} \left(\int p(x) p_\theta(\tilde{x}|x) dx \right) s_\theta(\tilde{x}) d\tilde{x}$$

$$\downarrow$$

Leibniz integration rule:

$$= \int \left(\int p(x) \nabla_{\tilde{x}} p_\theta(\tilde{x}|x) dx \right) s_\theta(\tilde{x}) d\tilde{x}$$

$$= \int \left(\int p(x) p_\theta(\tilde{x}|x) \nabla_{\tilde{x}} \log p_\theta(\tilde{x}|x) dx \right) s_\theta(\tilde{x}) d\tilde{x}$$

= For linearity of integrals we move $s_\theta(\tilde{x})$

$$= \iint p(x)\, p_\sigma(\tilde{x}|x)\, \nabla \log p_\sigma(\tilde{x}|x)\, s_\theta(\tilde{x})\, dx\, d\tilde{x}$$

Now we bring everything back:

$$\frac{1}{2} \mathbb{E}_{p_\sigma(\tilde{x})}\left[\| \nabla_{\tilde{x}} \log p_\sigma(\tilde{x}) - s_\theta(\tilde{x}) \|_2^2 \right] =$$

$$= \frac{1}{2} \int p_\sigma(\tilde{x})\, \nabla_{\tilde{x}}^2 \log p_\sigma(\tilde{x})\, d\tilde{x} + \frac{1}{2} \int p_\sigma(\tilde{x})\, s_\theta^2(\tilde{x})\, d\tilde{x}$$

$$+ \iint p(x)\, p_\sigma(\tilde{x}|x)\, \nabla \log p_\sigma(\tilde{x}|x)\, s_\theta(\tilde{x})\, dx\, d\tilde{x}$$

$$= \frac{1}{2} \mathbb{E}_{p_\sigma(\tilde{x})}\left[\| \nabla_{\tilde{x}} \log p_\sigma(\tilde{x}) \|_2^2 \right] +$$

$$+ \frac{1}{2} \mathbb{E}_{p_\sigma(\tilde{x})}\left[\| s_\theta(\tilde{x}) \|_2^2 \right] +$$

$$- \mathbb{E}_{x \sim p(x),\, \tilde{x} \sim p_\sigma(\tilde{x}|x)}\left[\nabla_{\tilde{x}} \log p_\sigma(\tilde{x}|x) \cdot s_\theta(\tilde{x}) \right]$$

We can use reconstruction

$$= \frac{1}{2} \mathbb{E}_{x \sim p(x),\, \tilde{x} \sim p_\sigma(\tilde{x}|x)} \left[\| s_\theta^2(\tilde{x}) \|_2^2 \right]$$

and so everything becomes:

$$= \frac{1}{2} \mathbb{E}_{p_\sigma(\tilde{x})} \left[\| \nabla_{\tilde{x}} \log p_\sigma(\tilde{x}) \|_2^2 \right] +$$

$$+ \frac{1}{2} \mathbb{E}_{x \sim p(x),\, \tilde{x} \sim p_\sigma(\tilde{x}|x)} \left[\| s_\theta^2(\tilde{x}) \|_2^2 - 2 \nabla_{\tilde{x}} \log p_\sigma(\tilde{x}|x) \cdot s_\theta(\tilde{x}) \right]$$

We still need remove $\| \nabla_{\tilde{x}} \log p_\sigma(\tilde{x}|x) \|_2^2$

$$= \frac{1}{2} \mathbb{E}_{p_\sigma(\tilde{x})} \left[\| \nabla_{\tilde{x}} \log p_\sigma(\tilde{x}) \|_2^2 \right] +$$

$$+ \frac{1}{2} \mathbb{E}_{x \sim p(x),\, \tilde{x} \sim p_\sigma(\tilde{x}|x)} \left[\| s_\theta^2(\tilde{x}) \|_2^2 - 2 \nabla_{\tilde{x}} \log p_\sigma(\tilde{x}|x) \cdot s_\theta(\tilde{x}) \right]$$

$$+ \| \nabla_{\tilde{x}} \log p_\sigma(\tilde{x}|x) \|_2^2 - \| \nabla_{\tilde{x}} \log p_\sigma(\tilde{x}|x) \|_2^2$$

$$\frac{1}{2} \mathbb{E}_{x \sim p(x),\, \tilde{x} \sim p_\sigma(\tilde{x}|x)} \left[\| s_\theta(\tilde{x}) - \nabla_{\tilde{x}} \log p_\sigma(\tilde{x}|x) \|_2^2 \right]$$

$$= \frac{1}{2} \mathbb{E}_{p_\sigma(\tilde{x})}\left[\| \nabla_{\tilde{x}} \log p_\sigma(\tilde{x}) \|_2^2 \right] +$$

$$+ \frac{1}{2} \mathbb{E}_{x \sim p(x),\, \tilde{x} \sim p_\sigma(\tilde{x}|x)}\left[\| s_\theta(\tilde{x}) - \nabla_{\tilde{x}} \log p_\sigma(\tilde{x}|x) \|_2^2 \right] +$$

$$- \| \nabla_{\tilde{x}} \log p_\sigma(\tilde{x}|x) \|_2^2$$

$$- \frac{1}{2} \mathbb{E}_{x \sim p(x),\, \tilde{x} \sim p_\sigma(\tilde{x}|x)}\left[\| \nabla_{\tilde{x}} \log p_\sigma(\tilde{x}|x) \|_2^2 \right]$$

But we see that - and - do not depend on θ so if we want to minimize they are constant and so our OBJECTIVE becomes:

$$= \mathbb{E}_{x \sim p(x),\, \tilde{x} \sim p_\sigma(\tilde{x}|x)}\left[\| s_\theta(\tilde{x}) - \nabla_{\tilde{x}} \log p_\sigma(\tilde{x}|x) \|_2^2 \right]$$

Remember $\tilde{x} = x + \epsilon$

$$p_\sigma(\tilde{x}|x) = \frac{1}{(2\pi)^{d/2} \sigma^2} e^{-\frac{1}{2\sigma^2} \|\tilde{x} - x\|^2}$$

So we have that $\nabla_{\tilde{x}} \log p_\sigma(\tilde{x}|x) =$

$$= \nabla_{\tilde{x}} \log \frac{1}{2\pi^{d/2}\sigma^2} \exp\left(-\frac{1}{2\sigma^2}\|\tilde{x}-x\|^2\right) =$$

$$= \nabla_{\tilde{x}} \log \frac{1}{2\pi^{d/2}\sigma^2} + \nabla_{\tilde{x}} \log e^{-\frac{1}{2\sigma^2}\|\tilde{x}-x\|^2} =$$

$$= -\frac{1}{2\sigma^2} 2\|\tilde{x}-x\| = -\frac{\tilde{x}-x}{\sigma^2} =$$

$$= \frac{x-\tilde{x}}{\sigma^2}$$

So we have that:

$$\frac{1}{2}\mathbb{E}_{x\sim p(x),\tilde{x}\sim p_\sigma(\tilde{x}|x)}\left[\|s_\theta(\tilde{x}) - \frac{1}{\sigma^2}(x-\tilde{x})\|_2^2\right]$$

Remember that $\tilde{x} = x + \epsilon$

$$= \frac{1}{2}\mathbb{E}_{x\sim p(x),\tilde{x}\sim p_\sigma(\tilde{x}|x)}\left[\|s_\theta(\tilde{x}) + \frac{\epsilon}{\sigma^2}\|_2^2\right]$$

So we can see how our model so predicts the negative of the noise that we added. The negative of the noise points in the direction of the data manifold where the data likely most grows

HOW DO WE GENERATE NEW DATA!

If we do just once there is an high chance we overshoot the data and fail so we do it step by step with small step

$$\tilde{x}_{i+1} \leftarrow \tilde{x}_i + \alpha \cdot s_\theta(\tilde{x}_i)$$

the size of the step is determined by α

The score points to the area where the likelihood is high because

Next's where we have implicit density of score

There is an easy fix, it is called

LANGEVIAN DYNAMICS

$$\tilde{x}_{i+1} \leftarrow \tilde{x}_i + \alpha s_\theta(\tilde{x}_i) + \sqrt{2\alpha} \cdot \epsilon$$

we add this $\epsilon \sim$ Random Gaussian noise

$\sqrt{2\alpha}$ very small

In this way we have nice arrows that do not collapse

Considering σ changing always

So the model can be different on all over the place

So we have $\tilde{x} = x + \epsilon \quad \epsilon \sim N(0, \sigma_\epsilon^2 I)$

So σ is not fixed anymore

If you increase $t \to +\infty$ the noise perturbation becomes stochastic

So we can use SDEs to model stochastic processes.

$$dx = f(x,t)\,dt + g(t)\,dW$$

- $f(x,t)$ → drift
- $g(t)$ → diffusion
- dW → Wiener process

$f(x,t)\,dt$: infinitesimal changes in time

$g(t)\,dW$: infinitesimal changes in noise

drift is fixed and describes if and how we change x in a deterministic way

diffusion describes the influence of the stochasticity overtime

In more modern we have

$\tilde{x} = x + G \qquad \epsilon \sim \mathcal{N}(0, \sigma_t^2 I)$

So change of x is only influenced by a stochastic term so $f(x,t) = 0$

$dx = g(t)\, dw$ So per $t \to +\infty$

$\sigma_t \to \sigma(t)$

$G \in [0,1]$

N.B.

IN DDPM $f(x,t) \neq 0$

Thus SDE corresponds to the process of noising our data FORWARD SDE

$d(x) = g(t)\, dw$

REVERSE SDE (MARKOVIAN DYNAMICS)

sampling

If FORWARD $dx = f(x,t)\, dt + g(t)\, dw$

The REVERSE SDE is:

$$dx = [f(x,t) - g^2(t)(\nabla_x \log p_t(x))]dt + g(t)dw$$

↙ Score function

So to reverse we can use the SCORE FUNCTION

CONNECTION TO DDPM

Score matching $\tilde{x} = x + \epsilon \quad \epsilon \sim \mathcal{N}(0, \sigma_t^2 I)$

DDPM $x_t = \sqrt{1-\beta_t}\, x_{t-1} + \sqrt{\beta_t}\,\epsilon \quad \epsilon \sim \mathcal{N}(0, I)$

So we see DRIFT is now zero

If we compute the Reverse SDE for DDPM we have Now

$$dx = [f(x,t) - g^2(t)\nabla_x \log p_t(x)]dt + g(t)dw$$

$$dx = x_t - x_{t-1}$$

$$= \sqrt{1-\beta_t}\, x_{t-1} + \sqrt{\beta_t}\,\epsilon - x_{t-1}$$

↓ Taylor expansion $= 1 - \frac{1}{2}\beta_t$

$$= \left(1 - \frac{1}{2}\beta_t\right) x_{t-1} + \sqrt{\beta_t}\,\epsilon - x_{t-1}$$

$$= x_{t-1} - \frac{1}{2}\beta_t x_{t-1} + \sqrt{\beta_t}\,\epsilon - x_{t-1}$$

$$= -\frac{1}{2}\beta_t x_{t-1} + \sqrt{\beta_t}\,\epsilon$$

Solo se $x_t - x_{t-1} \to 0$, $t \to +\infty$

$$dx = -\frac{1}{2}\beta(t)\, x\, dt + \sqrt{\beta(t)}\, dw$$

↓

FORWARD SDE

REVERSE SDE

$$f(x,t) = -\frac{1}{2}\beta(t)\, x$$

$$g(t) = \sqrt{\beta(t)}$$

We apply the formula

$$dx = \left[-\frac{1}{2}\beta(t)x - \beta(t)\nabla_x \log p_t(x) \right] dt + \sqrt{\beta(t)}\, dw$$

If we discretize it:

↓ DDPM SAMPLER

$$x_{t-1} = \frac{1}{\sqrt{1-\beta_t}} \left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right) + \sqrt{\beta_t}\, z$$

We discretize it with the

Euler-Maruyama Method