# Deep Generative Models

## 15. Score-based model through SDE


- 국가수리과학연구소 산업수학혁신센터 김민중

# Recap. of score-based model

- **Fisher divergence** between $p(\boldsymbol{x})$ and $q(\boldsymbol{x})$:

$$D_F(p,q) := \frac{1}{2} E_{\boldsymbol{x}\sim p}[\, \|\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}) - \nabla_{\boldsymbol{x}} \log q(\boldsymbol{x})\|_2^2]$$

- Score matching(Hyvärinen, 2005)

$$\frac{1}{2} E_{\boldsymbol{x}\sim p_{data}}[\, \|\boldsymbol{s}_\theta(\boldsymbol{x}) - \nabla_{\boldsymbol{x}} \log p_{data}(\boldsymbol{x})\|_2^2]$$

$$= E_{\boldsymbol{x}\sim p_{data}}\left[\frac{1}{2}\|\boldsymbol{s}_\theta(\boldsymbol{x})\|_2^2 + tr\big(\nabla_{\boldsymbol{x}}\boldsymbol{s}_\theta(\boldsymbol{x})\big)\right] + const.$$

# Denoising score matching with Langevin dynamics

$x \sim p_{data}(x)$
Data distribution

$q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x})$

Perturbation distribution/kernel

$\tilde{x} \sim q_\sigma(\tilde{x})$
Noise-perturbed data distribution

$$E_{\tilde{x}\sim q_\sigma}[\,\|\nabla_{\tilde{x}} \log q_\sigma(\tilde{x}) - s_\theta(\tilde{x})\|_2^2]$$
$$= E_{x\sim p_{data}(x)} E_{\tilde{x}\sim q_\sigma(\tilde{x}|x)}[\|\nabla_{\tilde{x}} \log q_\sigma(\tilde{x}|x) - s_\theta(\tilde{x})\|_2^2] + \text{const.}$$
$$= E_{x\sim p_{data}(x)} E_{z\sim N(\mathbf{0},\mathbf{I})}\left[\left\|\frac{1}{\sigma}z + s_\theta(x+\sigma z)\right\|_2^2\right] + \text{const.}$$

- **Pros**
  - more scalable than score matching
  - reduces score estimation to a denoising task
- **Con**: cannot estimate the score of clean data (noise-free)

$$s_\theta(x) \approx \nabla_x \log q_\sigma(x) \neq \nabla_x \log p_{data}(x)$$

# Denoising score matching with Langevin dynamics

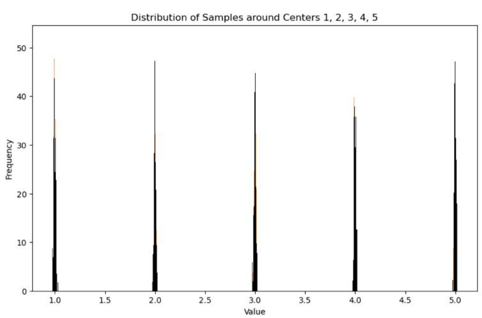
$\boldsymbol{x} \sim p_{data}(\boldsymbol{x})$
Data distribution

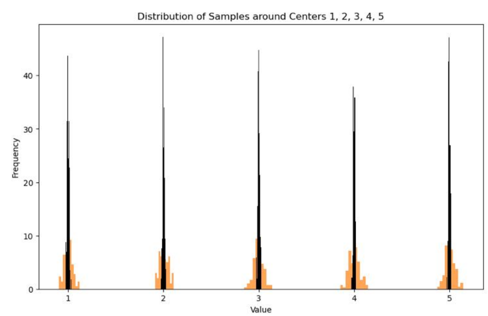
$q_\sigma(\tilde{\mathbf{x}} \mid \mathbf{x})$

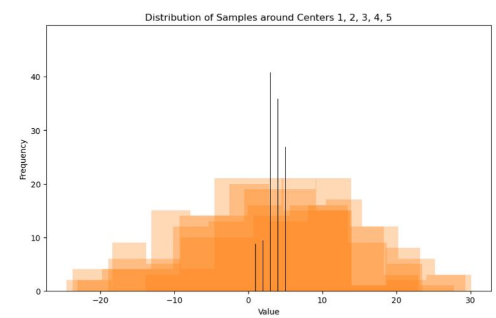
Perturbation distribution/kernel

$\tilde{\boldsymbol{x}} \sim q_\sigma(\tilde{\boldsymbol{x}})$
Noise-perturbed data distribution

$$q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x}) := N(\tilde{\boldsymbol{x}}|\boldsymbol{x}, \sigma^2 \boldsymbol{I}), \qquad q_\sigma(\tilde{\boldsymbol{x}}) = \int p_{data}(\boldsymbol{x}) q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x}) d\boldsymbol{x}$$




$p_{data}(\boldsymbol{x})$ $q_\sigma(\boldsymbol{x})$

# Denoising score matching with Langevin dynamics

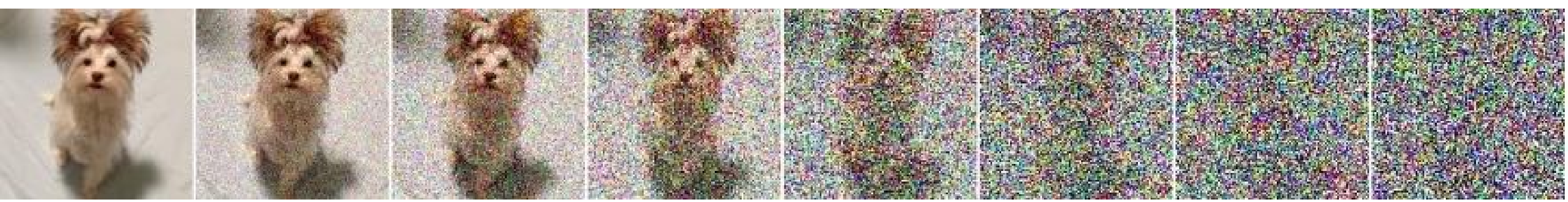
- Let $q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x}) := N(\tilde{\boldsymbol{x}}|\boldsymbol{x}, \sigma^2 I)$, $q_\sigma(\tilde{\boldsymbol{x}}) := \int p_{data}(\boldsymbol{x}) q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x}) d\boldsymbol{x}$
- Consider a sequence of positive noise scales

$$\sigma_1 < \sigma_2 < \cdots < \sigma_L$$

- $\sigma_1$ is small enough $q_{\sigma_1}(\boldsymbol{x}) \approx p_{data}(\boldsymbol{x})$
- $\sigma_L$ is large enough $q_{\sigma_L}(\boldsymbol{x}) \approx N(\boldsymbol{x}|\boldsymbol{0}, \sigma_L^2 \boldsymbol{I})$

Data space

Noise space



$p_{data}$ $q_{\sigma_1}$ $q_{\sigma_2}$ $\cdots$ $q_{\sigma_L} \approx N(\boldsymbol{0}, \sigma_L^2 \boldsymbol{I})$

# Denoising score matching with Langevin dynamics

- Let $q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x}) := N(\tilde{\boldsymbol{x}}|\boldsymbol{x}, \sigma^2 I)$, $q_\sigma(\tilde{\boldsymbol{x}}) := \int p_{data}(\boldsymbol{x}) q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x}) d\boldsymbol{x}$
- Consider a sequence of positive noise scales

$$\sigma_1 < \sigma_2 < \cdots < \sigma_L$$

- **Noise conditional score network**

$$\sum_{i=1}^{L} \sigma_i^2 E_{\boldsymbol{x} \sim p_{data}(\boldsymbol{x})} E_{\tilde{\boldsymbol{x}} \sim q_{\sigma_i}(\tilde{\boldsymbol{x}}|\boldsymbol{x})} \left[ \left\| \boldsymbol{s}_\theta(\tilde{\boldsymbol{x}}, \sigma_i) - \nabla_{\tilde{\boldsymbol{x}}} \log q_{\sigma_i}(\tilde{\boldsymbol{x}}|\boldsymbol{x}) \right\|_2^2 \right]$$

- Given sufficient data and model capacity, the optimal score-based model

$$\boldsymbol{s}_{\theta^*}(\boldsymbol{x}, \sigma_i) \approx \nabla_{\boldsymbol{x}} \log q_{\sigma_i}(\boldsymbol{x}) \quad \text{for} \quad \sigma \in \{\sigma_1, \dots, \sigma_L\}$$

- The weights $\sigma_i^2$ are related to $\sigma_i^2 \propto 1/E\left[ \left\| \nabla_{\tilde{\boldsymbol{x}}} \log p_{\sigma_i}(\tilde{\boldsymbol{x}}|\boldsymbol{x}) \right\|_2^2 \right]$

# Generation with annealed Langevin dynamics

- For each $q_{\sigma_i}(\boldsymbol{x})$ with $\sigma_1 < \sigma_2 < \cdots < \sigma_L$, Song & Ermond run $T$ steps of Langevin MCMC to get a sample sequentially

$$\boldsymbol{x}_i^t := \boldsymbol{x}_i^{t-1} + \frac{\alpha_i}{2} \boldsymbol{s}_{\theta^*}\left(\boldsymbol{x}_i^{t-1}, \sigma_i\right) + \sqrt{\alpha_i} \boldsymbol{z}, \qquad t = 1,2, \dots, T$$

- where $\alpha_i > 0$ is the step size and $\boldsymbol{z} \sim N(\boldsymbol{0}, \boldsymbol{I})$

$$\alpha_i := \epsilon \frac{\sigma_i^2}{\sigma_1^2}$$

- $\epsilon > 0$

**Generative Modeling by Estimating Gradients of the Data Distribution**
Song Yang, and Stefano Ermon. NeurIPS 2019

# Denoising diffusion probabilistic models(DDPM)

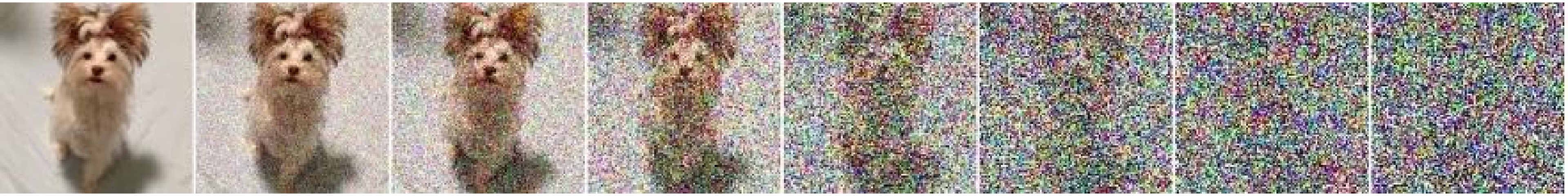
- Positive noise scales $0 < \beta_1 < \beta_2 \cdots < \beta_T < 1$
- $\boldsymbol{x}_0 \sim p_{data}(\boldsymbol{x})$, construct latent variables $\{\boldsymbol{x}_0, \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T\}$ s.t.

$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) \coloneqq \boldsymbol{N}(\boldsymbol{x}_t|\sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t\boldsymbol{I})$$

- I.e., $q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \boldsymbol{N}(\boldsymbol{x}_0|\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)\boldsymbol{I})$ where $\alpha_t \coloneqq 1-\beta_t$, $\bar{\alpha}_t \coloneqq \prod_{s=1}^{t} \alpha_s$
- Similar to SMLD, we can denote the perturbed data distribution

$$q(\boldsymbol{x}_t) \coloneqq \int q(\boldsymbol{x}_t|\boldsymbol{x}) p_{data}(\boldsymbol{x}) d\boldsymbol{x}$$

- The noise scales are prescribed s.t. $\boldsymbol{x}_T \sim q(\boldsymbol{x}_T) \approx N(\boldsymbol{0}, \boldsymbol{I})$



$p_{data}$ $q(\boldsymbol{x}_1)$ $q(\boldsymbol{x}_2)$ $\cdots$ $q(\boldsymbol{x}_T)$

# Denoising diffusion probabilistic models(DDPM)

- A variational Markov chain in the reverse direction is parametrized with

$$p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = N(\boldsymbol{x}_{t-1}|\boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t), \beta_t \boldsymbol{I})$$

- where $\boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t + \beta_t \boldsymbol{s}_\theta(\boldsymbol{x}_t, t)\right)$
- Re-weighted variant of the evidence lower bound

$$\sum_{t=1}^{T} (1-\bar{\alpha}_t)\, E_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})} E_{\boldsymbol{x}_t \sim q(\boldsymbol{x}_t|\boldsymbol{x})} \left[ \left\| \boldsymbol{s}_\theta(\boldsymbol{x}_t, t) - \nabla_{\boldsymbol{x}_t} \log q(\boldsymbol{x}_t|\boldsymbol{x}) \right\|_2^2 \right]$$

- which is a weighted sum of denoising score matching

$$\boldsymbol{s}_{\theta^*}(\boldsymbol{x}_t, t) \approx \nabla_{\boldsymbol{x}_t} \log q(\boldsymbol{x}_t)$$

- The weights $(1-\bar{\alpha}_t)$ are related to

$$(1-\bar{\alpha}_t) \propto 1/E\left[ \left\| \nabla_{\boldsymbol{x}_t} \log q(\boldsymbol{x}_t|\boldsymbol{x}) \right\|_2^2 \right]$$

# Denoising diffusion probabilistic models(DDPM)

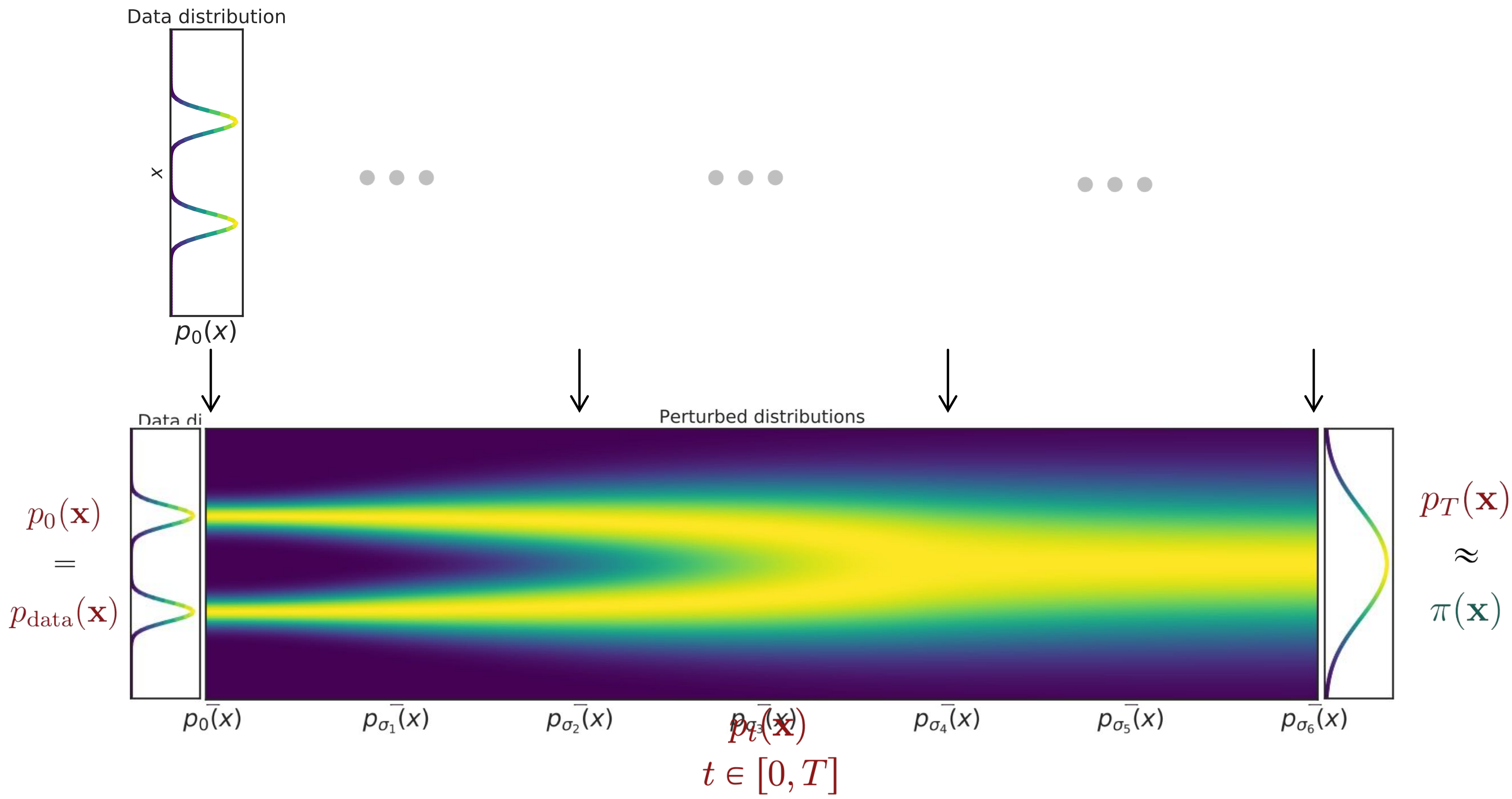
- Generate samples by starting from $\boldsymbol{x}_T \sim N(\boldsymbol{0}, \boldsymbol{I})$
- $\boldsymbol{x}_{t-1} \coloneqq \underbrace{\frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t + \beta_t \boldsymbol{s}_{\theta^*}(\boldsymbol{x}_t, t)\right)}_{=\boldsymbol{\mu}_{\theta^*}(\boldsymbol{x}_t, t)} + \sqrt{\beta_t}\boldsymbol{z},\ t = T, T-1, \ldots, 2$
- We call this method **ancestral sampling** $\left(\prod_{t=1}^{T} p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)\right)$

**Denoising Diffusion Probabilistic Models**
Jonathan Ho, Ajay Jain, Pieter Abbeel. NeurIPS 2020

# Summary of score-based models

- **SMLD** and **DDPM** involve sequentially corrupting training data with slowly increasing noise, and then learning to reverse this corruption to form a generative model of the data
- **SMLD** estimates the score at each noise scale and then use Langevin dynamics to sample from a sequence of decreasing noise scales during generation
- **DDPM** trains a sequence of probabilistic models to reverse each step of the noise corruption, using knowledge of the functional form of the reverse distributions to make training tractable

# Infinite noise levels



# Score-based model through SDE

- Extend the analysis to an infinite range of noise scales, where the evolution of perturbed data distributions follows an SDE as the noise level increases
- The process follows a **predefined SDE**, which does not depend on $p_{data}$ without trainable parameters
- This framework provides a way to understand and connect both the SMLD and DDPM methods by using SDEs

# Score-based model through SDE

- By generalizing the number of noise scales to infinity, we obtain:
  - **higher quality samples**
  - **exact log-likelihood computation**
  - **controllable generation for inverse problem solving**

# Ordinary differential equation

- For $t \geq 0$, consider an ODE which possesses the following form

$$d\boldsymbol{x}_t = \boldsymbol{f}(\boldsymbol{x}_t, t)dt$$

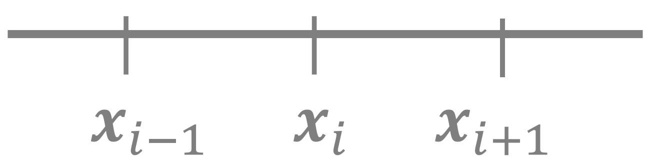
  - $\boldsymbol{x}_t \in \mathbb{R}^d$
  - $\boldsymbol{f}(\cdot, t): \mathbb{R}^d \to \mathbb{R}^d$ (drift coefficient)

- Then $\{\boldsymbol{x}_t\}_{t\in[0,T]}$ is a deterministic curve

- Numerically, the ODE can be seen as the limit

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \Delta t f(\boldsymbol{x}_i, i\Delta t), \qquad i = 0,1,\cdots$$

- Under $\Delta t \to 0$, where $t = i\Delta t$



# Solution of ODE

- $\{\boldsymbol{x}_t\}_{t\in[0,T]}$ solves ODE if it satisfies the
  - Differential form of the ODE

$$\frac{d\boldsymbol{x}_t}{dt} = \boldsymbol{f}(\boldsymbol{x}_t, t)$$

  - Or the integral form of the ODE

$$\boldsymbol{x}_t = \boldsymbol{x}_0 + \int_0^t f(\boldsymbol{x}_s, s)ds$$

- Example: $x_t \in \mathbb{R}$

$$dx_t = -\theta x_t dt$$

  - Then the solution of this ODE is

$$x_t = x_0 e^{-\theta t}$$

# Probability space

- $\Omega$: Sample space (e.g., $\{H, T\}$ or $\mathbb{R}^d$)
- $\mathcal{F}$: $\sigma$-algebra($\sigma$-field) on $\Omega$
  - $\Omega \in \mathcal{F}$
  - If $\mathrm{A} \in \mathcal{F}$, then $\mathrm{A}^{\mathrm{c}} \in \mathcal{F}$
  - closed under countable union
- Probability measure $P$ on $(\Omega, \mathcal{F})$
  - set function $P: \mathcal{F} \rightarrow \mathbb{R}_+$ with $P(\Omega) = 1$ (non negativity, null empty set, countable additivity)
- Probability distribution can be regarded as probability space $(\Omega, \mathcal{F}, P)$

# Random variable

- Measurable function $\boldsymbol{x}: \Omega \to E$ is called **random variable** if $\boldsymbol{x}$ is a function from a probability space $(\Omega, \mathcal{F}, P)$ to a measurable space $(E, \Sigma)$

- The probability that $\boldsymbol{x}$ takes on a value in a measurable set $S \subset E$ is written as

$$P(\boldsymbol{x} \in S) = P(\{\omega \in \Omega | \boldsymbol{x}(\omega) \in S\})$$

- We are interested in the image of $\boldsymbol{x}$
- $E$ is called state space

# Stochastic process

- $T$: index set (e.g., $\{0,1,2,\cdots\}$, $[0,1]$, $[0,\infty)$)
- If for each $t \in T$, $\boldsymbol{x}_t$ is a random variable, then $\{\boldsymbol{x}_t\}_{t\in T}$ is called stochastic process
  - $\{\boldsymbol{x}_t\}_{t\in T}$, $\{\boldsymbol{x}(t)\}_{t\in T}$, $\{\boldsymbol{x}_t, t \in T\}$, $\{\boldsymbol{x}(\omega, t), \omega \in \Omega, t \in T\}$

- $(\Omega, \mathcal{F}, P)$ with $\{\mathcal{F}_t\}_{t\in T}$

- In other words, stochastic process is a collection of random variables indexed by some index set $T$

# Brownian motion(a.k.a. Wiener process)

- The random motion of particles suspended in a medium
- Mathematically, 1-dim BM is characterized by
  - $w_0 = 0$
  - $w_t$ is almost surely continuous
  - $w_t$ has independent increments
  - $w_t - w_s \sim N(0, t-s)$ when $0 \le s < t$

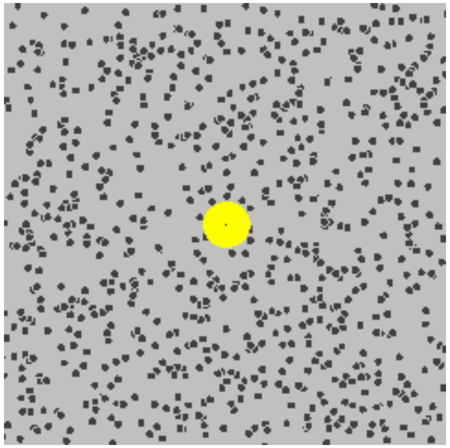
# Brownian motion(a.k.a. Wiener process)

- The random motion of particles suspended in a medium
- Mathematically, 1-dim BM is characterized by
  - $w_0 = 0$
  - $w_t$ is almost surely continuous
  - $w_t$ has independent increments
  - $w_t - w_s \sim N(0, t-s)$ when $0 \leq s < t$

- $d$-dim BM

$$\boldsymbol{w}_t = \left(w_{1,t}, w_{2,t}, \cdots, w_{d,t}\right)^T$$

- where $w_{i,t}$ are mutually independent 1-dim BM



# Brownian motion

- $T = [0, \infty)$
- $E = \mathbb{R}$
- $\Omega = C\big([0, \infty)\big)$
- $\mathcal{F}$: Borel $\sigma$-algebra of $\Omega$
- $P$: Wiener measure

$$P(w_t \in S) = \int_A \frac{1}{\sqrt{2t}} e^{-x^2/2t} dx$$

# Stochastic differential equation

- For $t \geq 0$, consider an SDE which possesses the following form
$$d\boldsymbol{x}_t = \boldsymbol{f}(\boldsymbol{x}_t, t)dt + g(t)d\boldsymbol{w}_t$$
  - $\boldsymbol{f}(\cdot, t): \mathbb{R}^d \rightarrow \mathbb{R}^d$ (drift coefficient)
  - $g(t) \in \mathbb{R}$ (diffusion coefficient)
  - $\boldsymbol{w}_t$ denotes a standard Brownian motion
  - $d\boldsymbol{w}_t$ can be viewed as infinitesimal white noise
  - $\{\boldsymbol{x}_t\}_{t\in[0,T]}$ is a stochastic process

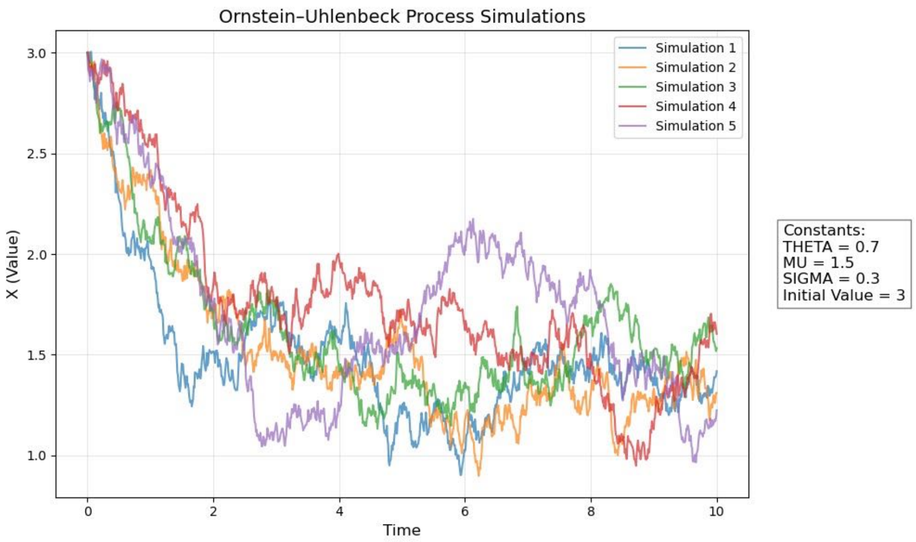
- Numerically, the SDE can be seen as the limit
$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \Delta t f(\boldsymbol{x}_i, i\Delta t) + g(i\Delta t)\sqrt{\Delta t}\boldsymbol{z}_i \quad i = 0,1,\cdots$$
- Under $\Delta t \rightarrow 0$, where $t = i\Delta t$ and $\boldsymbol{z}_i \sim N(\boldsymbol{0}, \boldsymbol{I})$

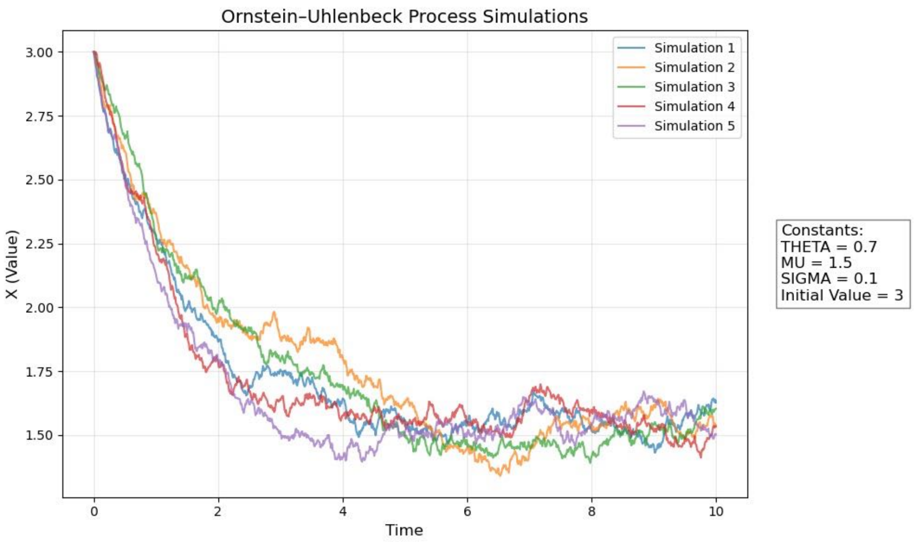
# Example: 1-dim Ornstein-Uhlenbeck process

- The Ornstein-Uhlenbeck process $x_t$ is defined by

$$dx_t = \theta(\mu - x_t)dt + \sigma dw_t$$

- where $\theta > 0$, $\sigma > 0$, $\mu \in \mathbb{R}$ and $w_t$ is 1-dim standard Brownian motion

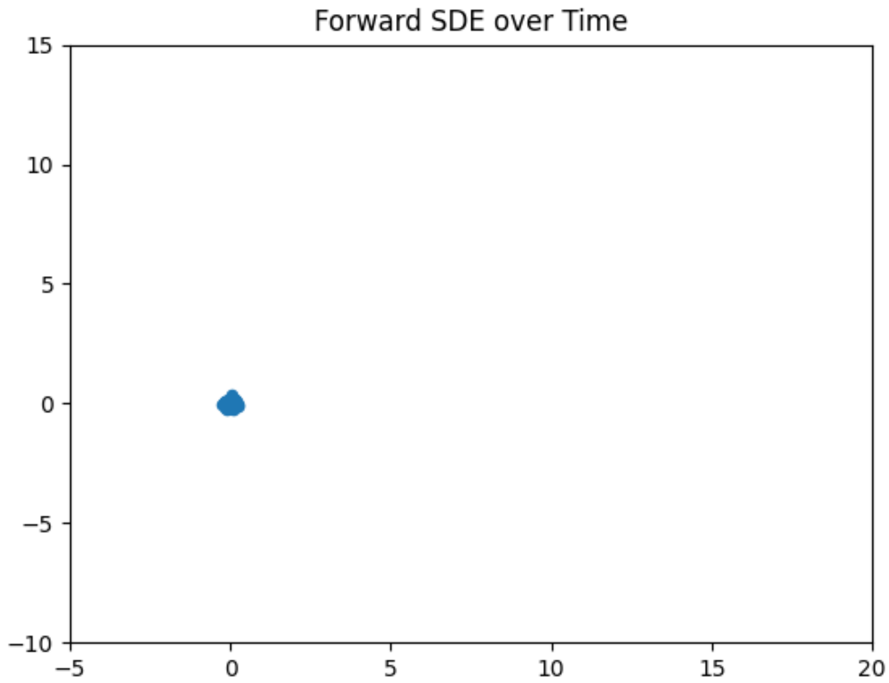


# Example: Forward SDE

$$d\boldsymbol{x}_t = \begin{pmatrix}1\\0\end{pmatrix} dt + \begin{pmatrix}0.1 & 0\\ 0 & 0.1\end{pmatrix} d\boldsymbol{w}_t, \qquad p_0(\boldsymbol{x}) = N\left(\boldsymbol{x} \middle| \begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}0.1 & 0\\ 0 & 0.1\end{pmatrix}\right)$$

- Then, $p_t(\boldsymbol{x}) = N\left(\boldsymbol{x} \middle| \begin{pmatrix}t\\0\end{pmatrix}, \begin{pmatrix}0.1+t & 0\\ 0 & 0.1+t\end{pmatrix}\right)$



# Solution of SDE

- $\{\boldsymbol{x}_t\}_{t\in[0,T]}$ is a solution for SDE if

$$\boldsymbol{x}_t = \boldsymbol{x}_0 + \int_0^t f(\boldsymbol{x}_s, s)ds + \int_0^t g(s)d\boldsymbol{w}_s$$

- The Itô stochastic integral is defined as

$$\int_0^t g(s)d\boldsymbol{w}_s = \lim_{\Delta t\to 0}\sum_{i=0} g(i\Delta t)\sqrt{\Delta t}\boldsymbol{z}_i$$

- where $\boldsymbol{z}_i \sim N(\boldsymbol{0}, \boldsymbol{I})$

# Representation of SDE

- For $t \geq 0$, consider an SDE which possesses the following form
$$d\boldsymbol{x}_t = \boldsymbol{f}(\boldsymbol{x}_t, t)dt + g(t)d\boldsymbol{w}_t$$
- The solution of an SDE is a continuous collection of random variables $\{\boldsymbol{x}_t\}_{t\in[0,T]}$

- These random variables trace stochastic trajectories as the time index $t$ grows from the start time $0$ to the end time $T$

- Let $p_t(\boldsymbol{x})$ denote the (marginal) probability density function of $\boldsymbol{x}_t$. I.e., $\int_A p_t(\boldsymbol{x})d\boldsymbol{x} = P(\boldsymbol{x}_t \in A)$
- The transition kernel from $\boldsymbol{x}_s$ to $\boldsymbol{x}_t$ where $0 \leq s < t \leq T$ is denoted by
$$p(\boldsymbol{x}_t|\boldsymbol{x}_s)$$

# Representation of SDE

- For $t \geq 0$, consider an SDE which possesses the following form
$$d\boldsymbol{x}_t = \boldsymbol{f}(\boldsymbol{x}_t, t)dt + g(t)d\boldsymbol{w}_t$$
- The solution of an SDE is a continuous collection of random variables $\{\boldsymbol{x}_t\}_{t\in[0,T]}$

- Here, $t \in [0, T]$ is analogous
  - multiple noise scales index $i = 1,2, \cdots, L$ with SMLD
  - variance schedules index $t = 1,2, \cdots, T$ with DDPM

- $p_0(\boldsymbol{x}) = p_{data}(\boldsymbol{x})$ data distribution
- After perturbing $p_{data}(\boldsymbol{x})$ with the stochastic process for a sufficiently long time $T$, $p_T(\boldsymbol{x})$ becomes close to a tractable noise distribution $\pi(\boldsymbol{x})$, called a prior distribution

# Fokker-Planck equation

- The noise perturbation procedure $p_t(\boldsymbol{x})$ under the SDE

$$d\boldsymbol{x}_t = \boldsymbol{f}(\boldsymbol{x}_t, t)dt + g(t)d\boldsymbol{w}_t$$

- is governed by the Fokker-Planck(FP) equation

- For $\boldsymbol{d} = \mathbf{1}$, the FP equation is

$$\partial_t p_t = -\partial_x(f p_t) + \frac{g^2}{2}\partial_x^2(p_t)$$

- More precisely, this means

$$\partial_t p_t(x) = -\partial_x\big(f(x,t)p_t(x)\big) + \frac{g^2(t)}{2}\partial_x^2\big(p_t(x)\big)$$

- for all $t > 0$ and $x \in \mathbb{R}$
- This is a partial differential equation(PDE)

# Fokker-Planck equation (multi-dim)

- The noise perturbation procedure $p_t(\boldsymbol{x})$ under the SDE

$$d\boldsymbol{x}_t = \boldsymbol{f}(\boldsymbol{x}_t, t)dt + \boldsymbol{g}(t)d\boldsymbol{w}_t$$

- is governed by the Fokker-Planck(FP) equation where $\boldsymbol{g}(\cdot): \mathbb{R} \to \mathbb{R}^{d\times d}$
- The multi-dim FP equation is

$$\partial_t p_t(\boldsymbol{x}) = -\sum_{i=1}^{d} \frac{\partial}{\partial x_i}\big(f_i(\boldsymbol{x}, t)p_t(\boldsymbol{x})\big) + \frac{1}{2}\sum_{i=1}^{d}\sum_{j=1}^{d} \frac{\partial^2}{\partial x_i \partial x_j} p_t(\boldsymbol{x}) \sum_{k=1}^{d} g_{ik}(t) g_{jk}(t)$$

$$= -\sum_{i=1}^{d} \frac{\partial}{\partial x_i}\big(f_i(\boldsymbol{x}, t)p_t(\boldsymbol{x})\big) + \frac{1}{2}\sum_{i=1}^{d}\sum_{j=1}^{d} \frac{\partial^2}{\partial x_i \partial x_j} p_t(\boldsymbol{x})\, g_{i,:}(t) g_{j,:}^T(t)$$

$$= -\nabla_{\boldsymbol{x}} \cdot \big(\boldsymbol{f}(\boldsymbol{x}, t)p_t(\boldsymbol{x})\big) + \frac{1}{2}\mathrm{Tr}\big(\boldsymbol{g}\boldsymbol{g}^T \nabla_{\boldsymbol{x}}^2 p_t(\boldsymbol{x})\big)$$

$$= -\nabla_{\boldsymbol{x}} \cdot \big(\boldsymbol{f}(\boldsymbol{x}, t)p_t(\boldsymbol{x})\big) + \frac{1}{2}\mathrm{Tr}(\boldsymbol{g}^T \nabla_{\boldsymbol{x}}^2 p_t(\boldsymbol{x})\boldsymbol{g})$$

# Example: Brownian Motion

- For a standard Brownian motion, the Fokker-Planck equation reduces to the **heat equation**

$$\partial_t p_t(\boldsymbol{x}) = \frac{1}{2}\mathrm{Tr}\left(\nabla_x^2 p_t(\boldsymbol{x})\right) = \frac{1}{2}\Delta_x p_t(\boldsymbol{x})$$

# Example: 1-dim Ornstein-Uhlenbeck process

- Consider the Ornstein-Uhlenbeck process $x_t$ is defined by

$$dx_t = -\theta x_t dt + \sigma dw_t$$

- Then,

$$p(x_t|x_0) = N\left(x_t \middle| e^{-\theta t}x_0, \frac{\sigma^2}{2\theta}\left(1 - e^{-2\theta t}\right)\right)$$

- If $x_0 \sim N\left(0, \frac{\sigma^2}{\theta}\right)$, then

$$x_t \sim N\left(0, \frac{\sigma^2}{2\theta}\right), \qquad p_t(x) = \frac{1}{\sqrt{\pi\sigma^2/\theta}} \exp\left[-\frac{\theta}{\sigma^2}x^2\right]$$

- $p_t(x)$ satisfies the FP equation

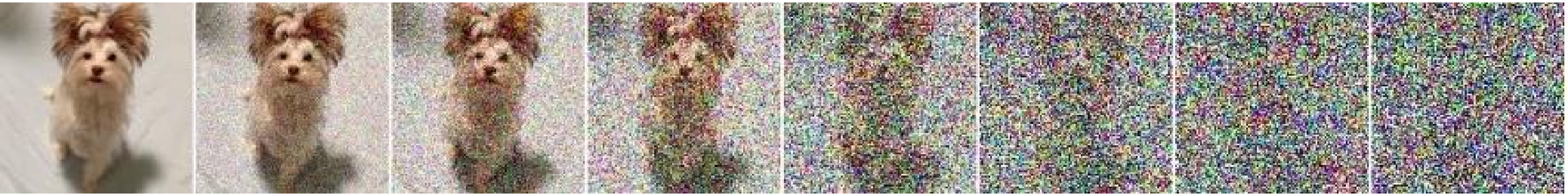
$$0 = \partial_t p_t(x) - \partial_x\left(f p_t(x)\right) + \frac{g^2}{2}\partial_x^2\left(p_t(x)\right)$$
$$= \partial_x\left(\theta x p_t(x)\right) + \frac{g^2}{2}\partial_x^2\left(p_t(x)\right) = 0$$

# Example: Ornstein-Uhlenbeck process

- The Ornstein-Uhlenbeck process

$$d\boldsymbol{x}_t = -\theta \boldsymbol{x}_t dt + \sigma d\boldsymbol{w}_t$$

- with $\theta \geq 0$ and $\sigma > 0$ adds noise to the datapoint $\boldsymbol{x}_t$
- As $T \to \infty$, all information is lost



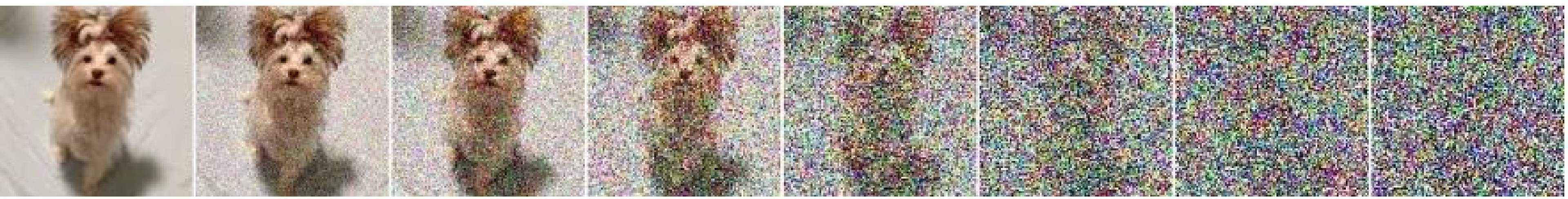
$p_{data}$ $\cdots$ $p_t(\boldsymbol{x})$ $\cdots$ $p_T(\boldsymbol{x})$

# Example: Ornstein-Uhlenbeck process

- The Ornstein-Uhlenbeck process

$$d\boldsymbol{x}_t = -\theta \boldsymbol{x}_t dt + \sigma d\boldsymbol{w}_t$$

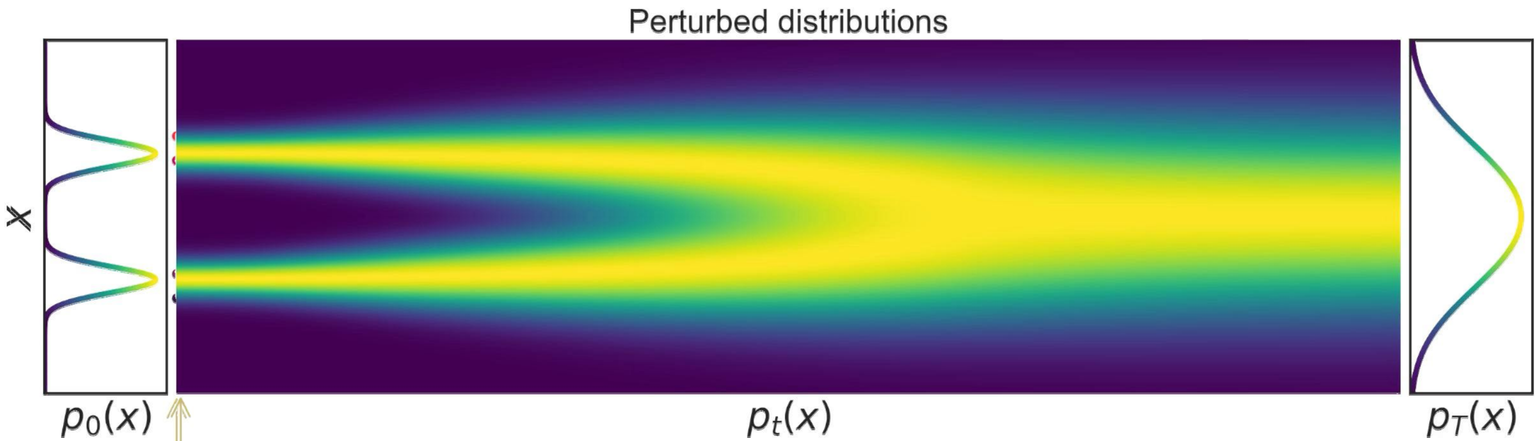
- with $\theta \geq 0$ and $\sigma > 0$ adds noise to the datapoint $\boldsymbol{x}_t$
- As $T \to \infty$, all information is lost

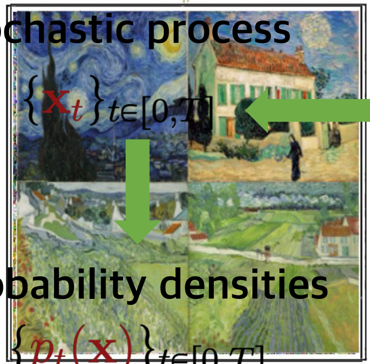


$p_{data}$ … $p_t(\boldsymbol{x})$ … $p_T(\boldsymbol{x})$

- Since $p(\boldsymbol{x}_t|\boldsymbol{x}_0) = N\left(\boldsymbol{x}_t \middle| e^{-\theta t}\boldsymbol{x}_0, \frac{\sigma^2}{2\theta}(1 - e^{-2\theta t})\boldsymbol{I}\right)$, we have $\boldsymbol{x}_T$ is approximately distributed as $N\left(\boldsymbol{0}, \frac{\sigma^2}{2\theta}\boldsymbol{I}\right)$ if $\theta > 0$ and $T \approx \infty$
- Sampling $\boldsymbol{x}_T \sim N\left(\boldsymbol{0}, \frac{\sigma^2}{2\theta}\boldsymbol{I}\right)$ is easy. Can we reverse the SDE to sample $\boldsymbol{x}_0$?

# Perturbing data with stochastic processes



**Stochastic process**

$\{\mathbf{x}_t\}_{t\in[0,T]}$

**Probability densities**

$\{p_t(\mathbf{x})\}_{t\in[0,T]}$

**Stochastic differential equation (SDE)**

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

Deterministic drift

Infinitesimal noise

$$p_T(\mathbf{x}) \approx \pi(\mathbf{x})$$

# Forward-time ODE

- To simulate

$$d\boldsymbol{x}_t = \boldsymbol{f}(\boldsymbol{x}_t, t)dt, \qquad \boldsymbol{x}_0 \text{ given}$$

- for $0 < t$ compute

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \Delta t \boldsymbol{f}(\boldsymbol{x}_i, i\Delta t), \qquad i = 0,1,\cdots$$

- for sufficiently small $\Delta t$ with $t = i\Delta t$

# Reverse-time ODE

- To simulate

$$d\boldsymbol{x}_t = \boldsymbol{f}(\boldsymbol{x}_t, t)dt, \qquad \boldsymbol{x}_T \text{ given}$$

- for $0 < t < T$, set $L = \lfloor T/\Delta t \rfloor$ and compute

$$\boldsymbol{x}_{i-1} = \boldsymbol{x}_i - \Delta t f(x_i, i\Delta t), \qquad i = L, L-1, \cdots, 1$$

- for sufficiently small $\Delta t > 0$

- Reversing time for ODEs is easy
  - Mapping from $\boldsymbol{x}_0$ to $\boldsymbol{x}_T$ is a one-to-one map

# Forward-time SDE

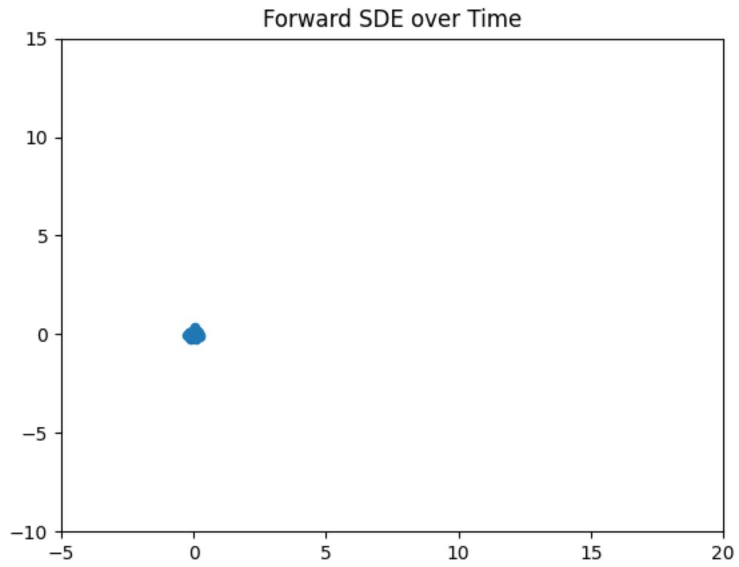
- To simulate

$$d\boldsymbol{x}_t = \boldsymbol{f}(\boldsymbol{x}_t, t)dt + g(t)d\boldsymbol{w}_t, \qquad \boldsymbol{x}_0 \sim p_0$$

- for $0 < t$, sample $\boldsymbol{x}_0 \sim p_0$ and compute

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + \Delta t f(x_i, i\Delta t) + g(i\Delta t)\sqrt{\Delta t}\boldsymbol{z}_i \quad i = 0,1,\cdots$$

- for sufficiently small $\Delta t > 0$ and $\boldsymbol{z}_i \sim N(\boldsymbol{0}, \boldsymbol{I})$



# Reverse-time SDE

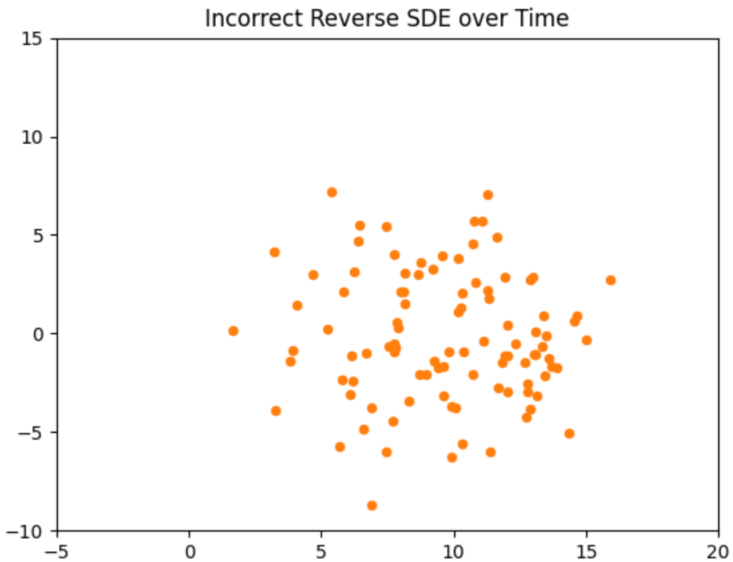
- To simulate

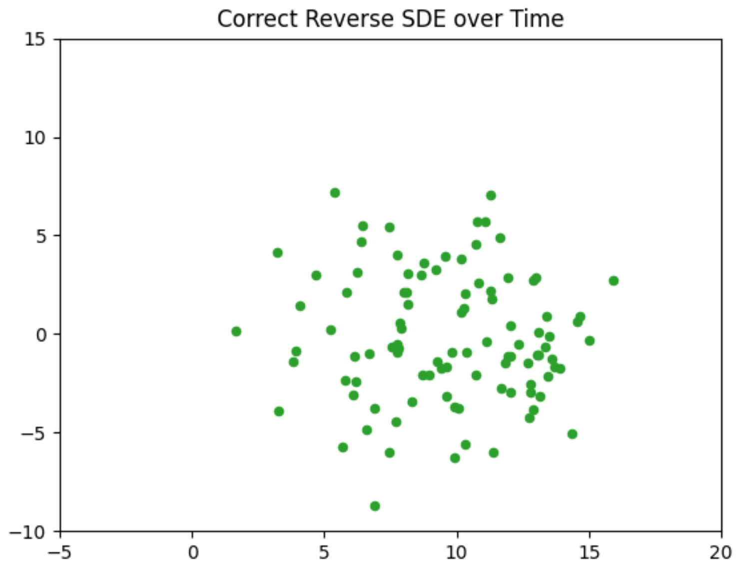
$$d\boldsymbol{x}_t = \boldsymbol{f}(\boldsymbol{x}_t, t)dt, + g(t)d\boldsymbol{w}_t, \qquad \boldsymbol{x}_T \sim p_T$$

- for $0 < t < T$, set $L = \lfloor T/\Delta t \rfloor$ and compute

$$\boldsymbol{x}_{i-1} = \boldsymbol{x}_i - \Delta t f(x_i, i\Delta t) - g(i\Delta t)\sqrt{\Delta t}\boldsymbol{z}_i, \qquad i = L, L-1, \cdots, 1$$

- **This does not work.** Rewinding time in SDEs takes more care



# Generating samples by reversing the SDE

- For an SDE,

$$d\boldsymbol{x}_t = \boldsymbol{f}(\boldsymbol{x}_t, t)dt + g(t)d\boldsymbol{w}_t, \qquad \boldsymbol{x}_0 \sim p_0$$

- has a corresponding reverse SDE, whose closed form is given by

$$d\boldsymbol{x}_t = \left[\boldsymbol{f}(\boldsymbol{x}_t, t) - g^2(t)\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)\right]dt + g(t)d\bar{\boldsymbol{w}}_t, \qquad \boldsymbol{x}_T \sim p_T$$

  - $dt$ represents a negative infinitesimal time step
  - $\bar{\boldsymbol{w}}_t$ is a standard BM when time flows backwards from $T$ to $0$. I.e. $\bar{\boldsymbol{w}}_t = \boldsymbol{w}_T - \boldsymbol{w}_{T-t}$

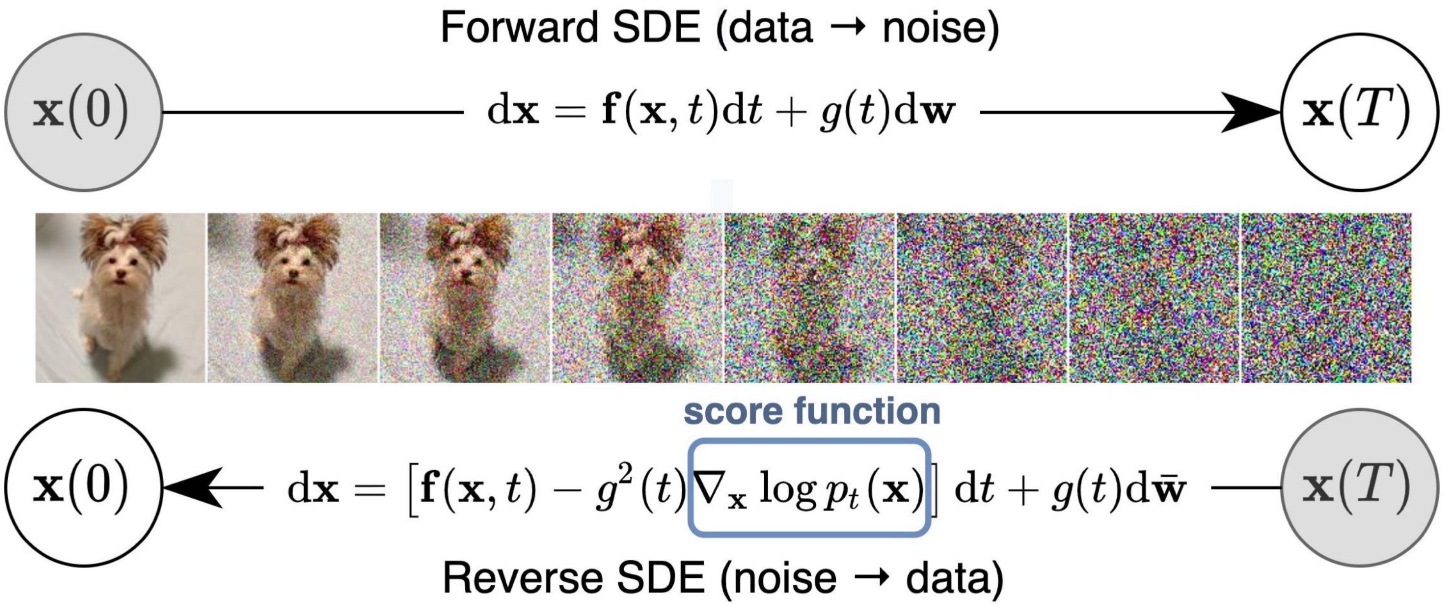
- In order to compute the reverse SDE, we need to estimate $\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})$ which is the score function of $p_t(\boldsymbol{x})$

**Reverse-time diffusion equation models**

B. D. O. Anderson. Stochastic Processes and their Applications. 1982

# Generating samples by reversing the SDE

- In order to compute the reverse SDE, we need to estimate $\nabla_x \log p_t(x)$ which is the score function of $p_t(x)$



# Estimating the reverse SDE with score-based models

- Solving the reverse SDE requires us to know the terminal distribution $p_T(\boldsymbol{x})$, and the score function $\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})$
- By design, $p_T(\boldsymbol{x})$ is close to the prior distribution $\pi(\boldsymbol{x})$ which is fully tractable

- In order to estimate $\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})$, train a time-dependent score-based model $\boldsymbol{s}_\theta(\boldsymbol{x}, t)$ such that

$$\boldsymbol{s}_\theta(\boldsymbol{x}, t) \approx \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})$$

- This is analogous to the NCSM $\boldsymbol{s}_\theta(\boldsymbol{x}, i)$ used for finite noise scales, trained such that $\boldsymbol{s}_\theta(\boldsymbol{x}, i) \approx \nabla_{\boldsymbol{x}} \log p_{\sigma_i}(\boldsymbol{x})$

# Estimating the reverse SDE with score-based models

- Training objective for $\mathbf{s}_\theta(\mathbf{x}, t)$ is a continuous weighted combination of Fisher divergences, given by

$$E_{t\sim U(0,T)}\left[\lambda(t) E_{\mathbf{x}\sim p_t(\mathbf{x})}[\,\|\mathbf{s}_\theta(\mathbf{x}, t) - \nabla_{\mathbf{x}} \log p_t(\mathbf{x})\|_2^2]\right]$$

- where $U(0, T)$ denotes a uniform distribution over the time interval $[0, T]$ and $\lambda: \mathbb{R}_+ \rightarrow \mathbb{R}_+$ is a positive weighting function

# (Recap) Foundation of DDPM

$$\underset{\theta}{\operatorname{argmin}} D\left(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) \parallel p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)\right)$$

$$= \underset{\theta}{\operatorname{argmin}} E_{\boldsymbol{x}_0 \sim p_{data}}\left[D\left(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) \parallel p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)\right)\right]$$

# Foundation of score-based models

$$\underset{\theta}{\operatorname{argmin}}\, E_{\boldsymbol{x}\sim p_t(\boldsymbol{x})}\left[\, \left\|\boldsymbol{s}_\theta(\boldsymbol{x},t) - \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})\right\|_2^2\right]$$

$$= \underset{\theta}{\operatorname{argmin}}\, E_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})} E_{\boldsymbol{x}_t\sim p(\boldsymbol{x}_t|\boldsymbol{x})}\left[\, \left\|\boldsymbol{s}_\theta(\boldsymbol{x}_t,t) - \nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t|\boldsymbol{x})\right\|_2^2\right]$$

# Estimating the reverse SDE with score-based models

- Training objective for $\mathbf{s}_\theta(\mathbf{x}, t)$ is a continuous weighted combination of Fisher divergences, given by

$$E_{t\sim U(0,T)}\left[\lambda(t)E_{\mathbf{x}\sim p_t(\mathbf{x})}[\,\|\mathbf{s}_\theta(\mathbf{x},t) - \nabla_{\mathbf{x}}\log p_t(\mathbf{x})\|_2^2]\right]$$

- Where $U(0,T)$ denotes a uniform distribution over the time interval $[0,T]$ and $\lambda: \mathbb{R}_+ \to \mathbb{R}_+$ is a positive weighting function

- The objective can be written as

$$E_{t\sim U(0,T)}\left[\lambda(t)E_{\mathbf{x}\sim p_{data}(\mathbf{x})}E_{\mathbf{x}_t\sim p(\mathbf{x}_t|\mathbf{x})}\left[\,\left\|\mathbf{s}_\theta(\mathbf{x}_t,t) - \nabla_{\mathbf{x}_t}\log p(\mathbf{x}_t|\mathbf{x})\right\|_2^2\right]\right]$$

- Typically, we use $\lambda(t) \propto 1/E\left[\left\|\nabla_{\mathbf{x}_t}\log p(\mathbf{x}_t|\mathbf{x})\right\|_2^2\right]$ to balance the magnitude of different score matching losses across time

# Remark of the transition kernel $p(\boldsymbol{x}_t|\boldsymbol{x})$

- We typically need to know the transition kernel $p(\boldsymbol{x}_t|\boldsymbol{x})$
- When $\boldsymbol{f}(\cdot, t)$ is affine, the transition kernel is always a (conditional) Gaussian distribution, where the mean and variance are often known in closed-forms

# Estimating the reverse SDE with score-based models

- Once our model $\boldsymbol{s}_\theta(\boldsymbol{x}, t)$ is trained to optimality, we can plug it into the reverse SDE to obtain an estimated reverse SDE

$$d\boldsymbol{x}_t = [\boldsymbol{f}(\boldsymbol{x}_t, t) - g^2(t)\boldsymbol{s}_\theta(\boldsymbol{x}_t, t)]dt + g(t)d\bar{\boldsymbol{w}}_t$$

- We can start with $\boldsymbol{x}_T \sim \pi$ and solve the above reverse SDE to obtain a sample $\boldsymbol{x}_0$ obtained in such way as $p_\theta$

- If weighting function $\lambda(t) = g^2(t)$, then

$$D\big(p_0(x) \parallel p_\theta(x)\big) \leq \frac{T}{2} E_{t\sim U(0,T)}\Big[\lambda(t) E_{\boldsymbol{x}\sim p_t(\boldsymbol{x})}[\,\|\boldsymbol{s}_\theta(\boldsymbol{x}, t) - \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})\|_2^2]\Big] + D(p_T \parallel \pi)$$

**Maximum Likelihood Training of Score-Based Diffusion Models**
Y. Song, C. Durkan, I. Murray, S. Ermon. NeurIPS 2021.

# How to solve the reverse SDE

- By solving the estimated reverse SDE with numerical SDE solvers, we can simulate the reverse stochastic process for sample generation
- **Euler-Maruyama method**(analogous to Euler for ODEs)
  - Small positive time step $\Delta t \approx 0$
  - Initializes $t = T$, and iterates the following procedure until $t \approx 0$

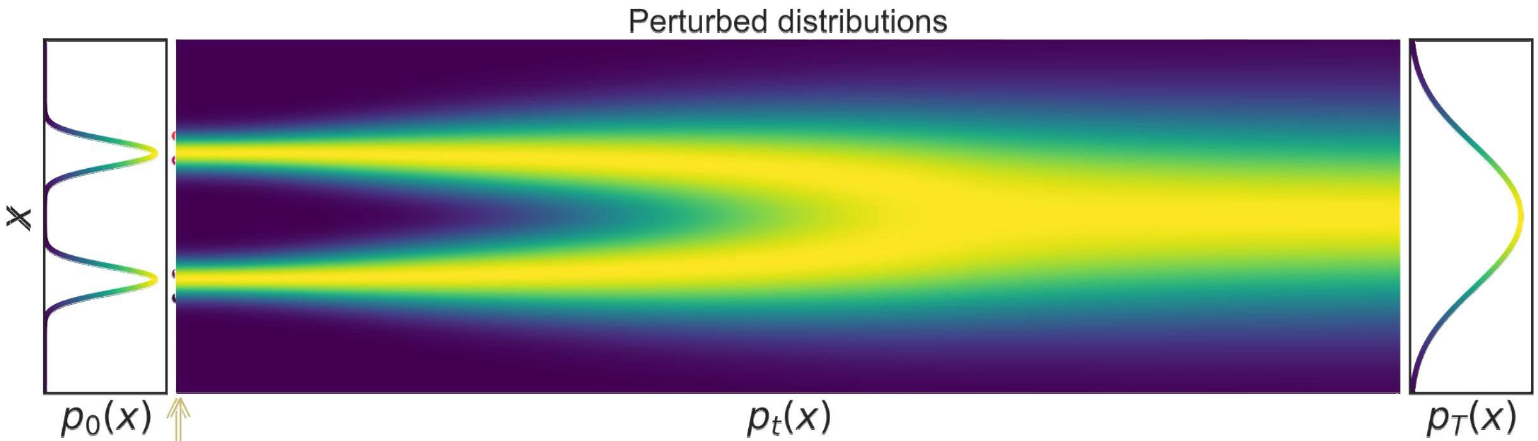
$$\Delta \boldsymbol{x} \leftarrow [\boldsymbol{f}(\boldsymbol{x}, t) - g^2(t)\boldsymbol{s}_\theta(\boldsymbol{x}, t)]\Delta t + g(t)\sqrt{\Delta t}\boldsymbol{z}$$
$$\boldsymbol{x} \leftarrow \boldsymbol{x} + \Delta \boldsymbol{x}$$
$$t \leftarrow t - \Delta t$$

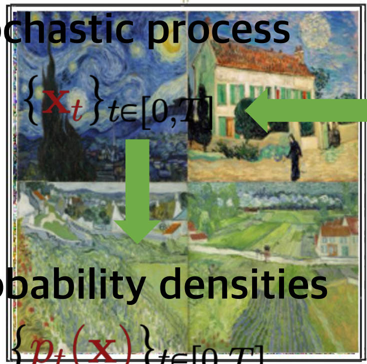
  - Here $\boldsymbol{z} \sim N(\boldsymbol{0}, \Delta t \boldsymbol{I})$
  - I.e. $\boldsymbol{x}_{t-\Delta t} = \boldsymbol{x}_t - \Delta t[\boldsymbol{f}(\boldsymbol{x}_t, t) - g^2(t)\boldsymbol{s}_\theta(\boldsymbol{x}_t, t)] + g(t)\sqrt{\Delta t}\boldsymbol{z}$

# How to solve the reverse SDE

- By solving the estimated reverse SDE with numerical SDE solvers, we can simulate the reverse stochastic process for sample generation
- Other numerical SDE solvers can be employed for example **Milstein method** and **stochastic Runge-Kutta method**

# Perturbing data with stochastic processes



**Stochastic process**

$\{\mathbf{x}_t\}_{t\in[0,T]}$

**Probability densities**

$\{p_t(\mathbf{x})\}_{t\in[0,T]}$

**Stochastic differential equation (SDE)**

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

Deterministic drift

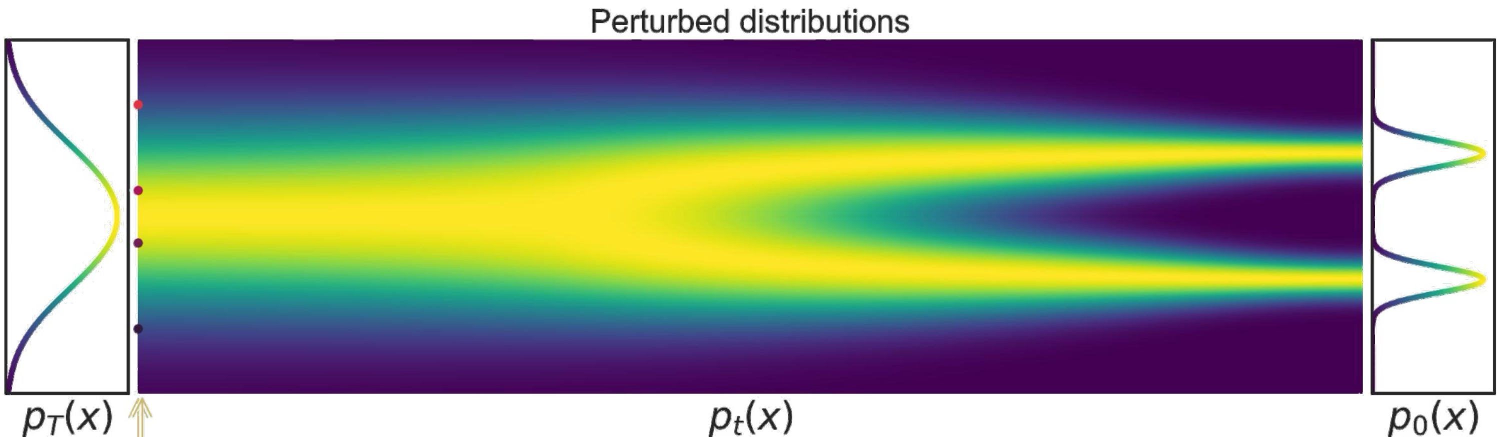
Infinitesimal noise

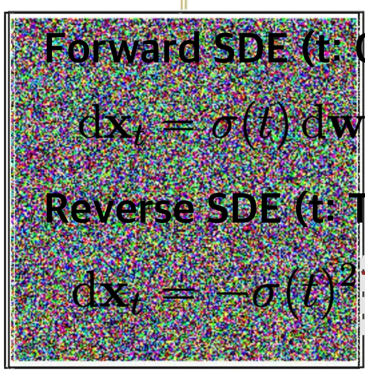
**WLOG: Toy SDE**

$$\mathrm{d}\mathbf{x}_t = \sigma(t)\,\mathrm{d}\mathbf{w}_t$$

$p_T(\mathbf{x}) \approx \pi(\mathbf{x})$

# Generation via reverse stochastic processes



$\pi(\mathbf{x}) \approx p_T(\mathbf{x})$

**Forward SDE (t: 0→T)**

$$\mathrm{d}\mathbf{x}_t = \sigma(t)\,\mathrm{d}\mathbf{w}_t$$

**Reverse SDE (t: T→0)**

$$\mathrm{d}\mathbf{x}_t = -\sigma(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t)\,\mathrm{d}t + \sigma(t)\,\mathrm{d}\bar{\mathbf{w}}_t$$

Score function!

Infinitesimal noise in the reverse time direction

# Score-based generative modeling via SDEs

- Time-dependent score-based model

$$\boldsymbol{s}_\theta(\boldsymbol{x}, t) \approx \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})$$

- Training objective

$$E_{t\sim U(0,T)}\left[\lambda(t) E_{\boldsymbol{x}\sim p_t(\boldsymbol{x})}[\, \|\boldsymbol{s}_\theta(\boldsymbol{x}, t) - \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})\|_2^2]\right]$$

# Score-based generative modeling via SDEs

- Time-dependent score-based model

$$\boldsymbol{s}_\theta(\boldsymbol{x}, t) \approx \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})$$

- Training objective

$$E_{t\sim U(0,T)}\left[\lambda(t)E_{\boldsymbol{x}\sim p_t(\boldsymbol{x})}[\,\|\boldsymbol{s}_\theta(\boldsymbol{x}, t) - \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})\|_2^2]\right]$$

- In case of $d\boldsymbol{x}_t = \sigma(t)d\boldsymbol{w}_t$ with $0 \le t \le T$, the reverse-time SDE is

$$d\boldsymbol{x}_t = -\sigma^2(t)\boldsymbol{s}_\theta(\boldsymbol{x}_t, t)dt + \sigma(t)d\bar{\boldsymbol{w}}_t$$

- Euler-Maruyama method

$$\boldsymbol{x}_{t-\Delta t} = \boldsymbol{x}_t - \sigma^2(t)\boldsymbol{s}_\theta(\boldsymbol{x}_t, t)\Delta t + \sigma(t)\boldsymbol{z}$$

- where $\boldsymbol{z}\sim N(\boldsymbol{0}, \Delta t\boldsymbol{I})$

# Predictor-Corrector sampling methods

- In addition, there are two special properties of our reverse SDE that allow for even more flexible sampling methods:
  - estimation of $\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})$ via time-dependent score-based model $\boldsymbol{s}_\theta(\boldsymbol{x}, t)$
  - sampling from each marginal distribution $p_t(\boldsymbol{x})$

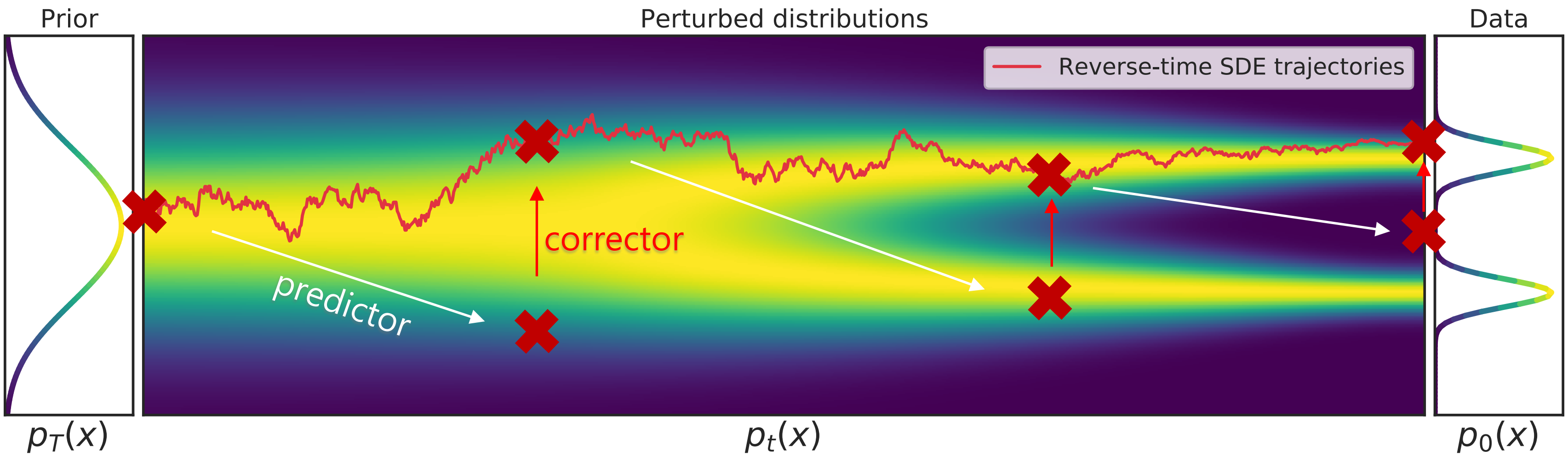
# Predictor-Corrector sampling methods

- Thus, we can apply score-based MCMC approaches to fine-tune the trajectories obtained from numerical SDE solvers
- We propose **Predictor-Corrector samplers**
  - **Predictor**: any numerical SDE solver predicting $\boldsymbol{x}_{t-\Delta t} \sim p_{t-\Delta t}(\boldsymbol{x})$ from an existing sample $\boldsymbol{x}_t \sim p_t(\boldsymbol{x})$
  - **Corrector**: score-based MCMC procedure

- At each step of the Predictor-Corrector sampler, we first use the **predictor** to choose a proper step size $\Delta t > 0$, and then predict $\boldsymbol{x}_{t-\Delta t}$ based on the current sample $\boldsymbol{x}_t$
- Next, we run several **corrector** steps to improve the sample $\boldsymbol{x}_{t-\Delta t}$ according to our score-based model $\boldsymbol{s}_\theta(\boldsymbol{x}_{t-\Delta t}, t - \Delta t)$ so that $\boldsymbol{x}_{t-\Delta t}$ becomes a high-quality sample from $p_{t-\Delta t}(\boldsymbol{x})$

# Predictor-Corrector sampling methods

- Predictor-Corrector sampling
  - **Predictor**: Numerical SDE solver
  - **Corrector**: Score-based MCMC



# Results on predictor-corrector sampling

Table 1: Comparing different reverse-time SDE solvers on CIFAR-10. Shaded regions are obtained with the same computation (number of score function evaluations). Mean and standard deviation are reported over five sampling runs. "P1000" or "P2000": predictor-only samplers using 1000 or 2000 steps. "C2000": corrector-only samplers using 2000 steps. "PC1000": Predictor-Corrector (PC) samplers using 1000 predictor and 1000 corrector steps.

| FID↓ Sampler / Predictor | Variance Exploding SDE (SMLD) | | | | Variance Preserving SDE (DDPM) | | | |
|---|---|---|---|---|---|---|---|---|
| | P1000 | P2000 | C2000 | PC1000 | P1000 | P2000 | C2000 | PC1000 |
| ancestral sampling | 4.98 ± .06 | 4.88 ± .06 | | **3.62 ± .03** | 3.24 ± .02 | 3.24 ± .02 | | **3.21 ± .02** |
| reverse diffusion | 4.79 ± .07 | 4.74 ± .08 | 20.43 ± .07 | **3.60 ± .02** | 3.21 ± .02 | 3.19 ± .02 | 19.06 ± .06 | **3.18 ± .01** |
| probability flow | 15.41 ± .15 | 10.54 ± .08 | | **3.51 ± .04** | 3.59 ± .04 | 3.23 ± .03 | | **3.06 ± .03** |

**Score-Based Generative Modeling through Stochastic Differential Equations**
Song, Sohl-Dickstein, Kingma, Kumar, Ermon, Poole. ICLR 2021.

# High-Fidelity Generation for 1024x1024 Images

**Score-Based Generative Modeling through Stochastic Differential Equations**
Song, Sohl-Dickstein, Kingma, Kumar, Ermon, Poole. ICLR 2021.

# VE and VP forward SDEs

- The O-U process $\boldsymbol{x}_t$ is defined by

$$d\boldsymbol{x}_t = -\theta \boldsymbol{x}_t dt + \sigma d\boldsymbol{w}_t$$

- where $\theta > 0$, $\sigma > 0$ and $\boldsymbol{w}_t$ is $d$-dim standard Brownian motion

- Two types O-U processes are primarily considered for the forward SDE
  - Variance-exploding(VE)

$$d\boldsymbol{x}_t = \sigma d\boldsymbol{w}_t$$

$$p(\boldsymbol{x}_t|\boldsymbol{x}_0) = (\boldsymbol{x}_t|\gamma_t \boldsymbol{x}_0, \sigma_t^2 \boldsymbol{I}), \qquad \gamma_t = 1, \sigma_t^2 = t\sigma^2$$

  - Variance -preserving(VP)

$$d\boldsymbol{x}_t = -\theta \boldsymbol{x}_t dt + \sigma d\boldsymbol{w}_t$$

$$p(\boldsymbol{x}_t|\boldsymbol{x}_0) = (\boldsymbol{x}_t|\gamma_t \boldsymbol{x}_0, \sigma_t^2 \boldsymbol{I}), \qquad \gamma_t = e^{-\theta t}, \sigma_t^2 = \frac{\sigma^2}{2\theta}\left(1 - e^{-2\theta t}\right)$$

# VE and VP forward SDEs

- Two types O-U processes are primarily considered for the forward SDE
  - Variance-exploding(VE)

$$d\boldsymbol{x}_t = \sigma d\boldsymbol{w}_t$$

$$p(\boldsymbol{x}_t|\boldsymbol{x}_0) = (\boldsymbol{x}_t|\gamma_t\boldsymbol{x}_0, \sigma_t^2\boldsymbol{I}), \qquad \gamma_t = 1, \sigma_t^2 = t\sigma^2$$

  - Variance -preserving(VP)

$$d\boldsymbol{x}_t = -\theta\boldsymbol{x}_t dt + \sigma d\boldsymbol{w}_t$$

$$p(\boldsymbol{x}_t|\boldsymbol{x}_0) = (\boldsymbol{x}_t|\gamma_t\boldsymbol{x}_0, \sigma_t^2\boldsymbol{I}), \qquad \gamma_t = e^{-\theta t}, \sigma_t^2 = \frac{\sigma^2}{2\theta}\left(1 - e^{-2\theta t}\right)$$

  - In both cases,

$$p(\boldsymbol{x}_t|\boldsymbol{x}_0) = (\boldsymbol{x}_t|\gamma_t\boldsymbol{x}_0, \sigma_t^2\boldsymbol{I})$$

  - i.e. $\boldsymbol{x}_t|\boldsymbol{x}_0 = \gamma_t\boldsymbol{x}_0 + \sigma_t\boldsymbol{\epsilon}$ where $\boldsymbol{\epsilon} \sim N(\boldsymbol{0}, \boldsymbol{I})$

# General VE SDE

- Let $\sigma(t)$ be a non-decreasing function of $t$

- General VE SDE:

$$d\boldsymbol{x}_t = \sqrt{\frac{d[\sigma^2(t)]}{dt}}\, d\boldsymbol{w}_t$$

$$p(\boldsymbol{x}_t|\boldsymbol{x}_0) = N(\boldsymbol{x}_t|\gamma_t \boldsymbol{x}_0, \sigma_t^2 \boldsymbol{I}), \qquad \gamma_t = 1, \sigma_t^2 = \sigma^2(t)$$

- Although the mean is preserved, the variance explodes

# General VP SDE

- Let $\theta: [0, \infty) \to \mathbb{R}_+$ be a function
- General VP SDE:

$$d\boldsymbol{x}_t = -\frac{\theta(t)}{2}\boldsymbol{x}_t dt + \sqrt{\theta(t)}d\boldsymbol{w}_t$$

$$p(\boldsymbol{x}_t|\boldsymbol{x}_0) = N(\boldsymbol{x}_t|\gamma_t\boldsymbol{x}_0, \sigma_t^2\boldsymbol{I}),$$

$$\gamma_t = e^{-\frac{1}{2}\int_0^t \theta(s)ds}, \sigma_t^2 = 1 - e^{-\int_0^t \theta(s)ds}$$

- In particular,

$$\mathrm{Var}(\boldsymbol{x}_t) = \boldsymbol{I} + e^{-\int_0^t \theta(s)ds}(\mathrm{Var}(\boldsymbol{x}_0) - \boldsymbol{I})$$

  - If $\mathrm{Var}(\boldsymbol{x}_0) = \boldsymbol{I}$, then

$$\mathrm{Var}(\boldsymbol{x}_t) = \boldsymbol{I}$$

# Training with O-U and DSM

- Using $\boldsymbol{x}_t|\boldsymbol{x}_0 = \gamma_t \boldsymbol{x}_0 + \sigma_t \boldsymbol{\epsilon}$ where $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \boldsymbol{I})$, the score function simplifies to

$$\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t|\boldsymbol{x}) = \frac{\gamma_t \boldsymbol{x} - \boldsymbol{x}_t}{\sigma_t^2} = -\frac{\boldsymbol{\epsilon}}{\sigma_t}$$

# Variance exploding SDEs (SMLD)

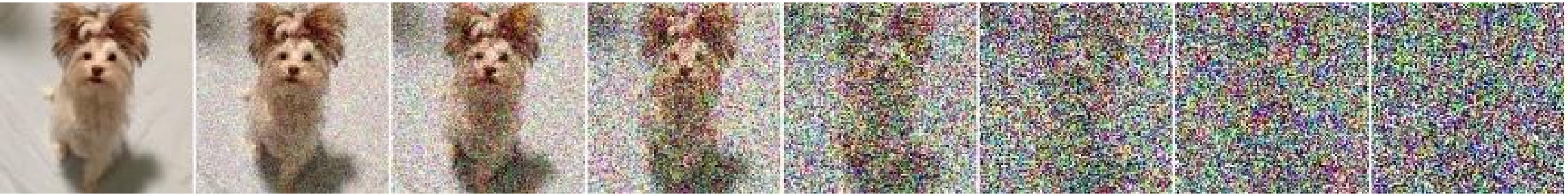
- Let $q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x}) \coloneqq N(\tilde{\boldsymbol{x}}|\boldsymbol{x}, \sigma^2 I)$, $q_\sigma(\tilde{\boldsymbol{x}}) \coloneqq \int p_{data}(\boldsymbol{x}) q_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x}) d\boldsymbol{x}$
- Consider a sequence of positive noise scales $\sigma_1 < \sigma_2 < \cdots < \sigma_L$
- Each perturbation kernel $q_{\sigma_i}(\tilde{\boldsymbol{x}}|\boldsymbol{x})$ can be derived from the following Markov chain:

$$\boldsymbol{x}_i = \boldsymbol{x}_{i-1} + \sqrt{\sigma_i^2 - \sigma_{i-1}^2}\boldsymbol{z}_{i-1}, \qquad i = 1, \cdots, L$$

- where $\boldsymbol{z}_{i-1} \sim N(\boldsymbol{0}, \boldsymbol{I})$, $\boldsymbol{x}_0 \sim \boldsymbol{p}_{data}$ and $\sigma_0 \coloneqq 0$ to simplify the notation

Data space

Noise space



$p_{data}$ $q_{\sigma_1}$ $q_{\sigma_2}$ $\cdots$ $q_{\sigma_L} \approx N(\boldsymbol{0}, \sigma_L^2 \boldsymbol{I})$

# Variance exploding SDEs (SMLD)

- In the limit of $L \to \infty$, $\{\sigma_i\}_{i=1}^{L}$ becomes a function $\sigma(t)$ and $\boldsymbol{z}_i$ becomes $\boldsymbol{z}(t)$
- The Markov chain $\{\boldsymbol{x}_i\}_{i=1}^{L}$ becomes a continuous stochastic process $\{\boldsymbol{x}_t\}_{t=0}^{1}$ (or $\{\boldsymbol{x}_t, 0 \le t \le 1\}$)
- Let

$$\boldsymbol{x}_{i/L} \coloneqq \boldsymbol{x}_i, \qquad \sigma(i/L) \coloneqq \sigma_i, \qquad \boldsymbol{z}(i/L) = \boldsymbol{z}_i$$

- Then we can rewrite

$$\boldsymbol{x}_i = \boldsymbol{x}_{i-1} + \sqrt{\sigma_i^2 - \sigma_{i-1}^2}\boldsymbol{z}_{i-1}, \qquad i = 1, \cdots, L$$

- as follows with $\Delta t = 1/L$ and $t \in \left\{0, \frac{1}{L}, \cdots, \frac{L-1}{L}\right\}$:

$$\boldsymbol{x}_{t+\Delta t} = \boldsymbol{x}_t + \sqrt{\sigma^2(t+\Delta t) - \sigma^2(t)}\boldsymbol{z}_t \approx \boldsymbol{x}_t + \sqrt{\frac{d\sigma^2(t)}{dt}\Delta t}\boldsymbol{z}_t$$

# Variance exploding SDEs (SMLD)

- In the limit of $\Delta t \to 0$,

$$\boldsymbol{x}_{t+\Delta t} = \boldsymbol{x}_t + \sqrt{\sigma^2(t+\Delta t) - \sigma^2(t)}\boldsymbol{z}_t \approx \boldsymbol{x}_t + \sqrt{\frac{d[\sigma^2(t)]}{dt}}\sqrt{\Delta t}\boldsymbol{z}_t$$

- converges to

$$d\boldsymbol{x}_t = \sqrt{\frac{d[\sigma^2(t)]}{dt}}d\boldsymbol{w}_t$$

- VE SDE always yields a process with exploding variance when $t \to \infty$

# SDE in the wild (SMLD)

- In SMLD, the noise scales $\{\sigma_i\}_{i=1}^{L}$ is a geometric sequence
- SMLD models normalize image inputs to the range $[0,1]$
- Since $\{\sigma_i\}_{i=1}^{L}$ is a geometric sequence, we have

$$\sigma\left(\frac{i}{L}\right) = \sigma_i = \sigma_{\min}\left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)^{\frac{i-1}{L-1}}, \qquad i = 1,2,\cdots,L$$

- In the limit of $L \to \infty$, we have $\sigma(t) = \sigma_{\min}\left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)^{t}$ for $t \in (0,1]$
- Thus, the corresponding VE SDE is

$$d\boldsymbol{x}_t = \sigma_{\min}\left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)^{t}\sqrt{2\log\frac{\sigma_{\max}}{\sigma_{\min}}}\,d\boldsymbol{w}_t, \qquad t \in (0,1]$$

- and the perturbation kernel can be derived:

$$p(\boldsymbol{x}_t|\boldsymbol{x}) = \boldsymbol{N}\left(\boldsymbol{x}_t|\boldsymbol{x}, \sigma_{\min}^2\left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)^{2t}\boldsymbol{I}\right)$$

# SDE in the wild (SMLD)

- There is one subtlety when $t = 0$: by definition $\sigma(0) = \sigma_0 = 0$
- However, $\sigma(0^+) \coloneqq \lim_{t \to 0+} \sigma(t) = \sigma_{\min} \neq 0$
- It means that $\sigma(t)$ for SMLD is not differentiable at $t = 0$
- Thus, we bypass this issue by always solving the SDE and its associated probability flow ODE in the range $t \in [\epsilon, 1]$ for some small $\epsilon > 0$. e.g., $\epsilon = 10^{-5}$

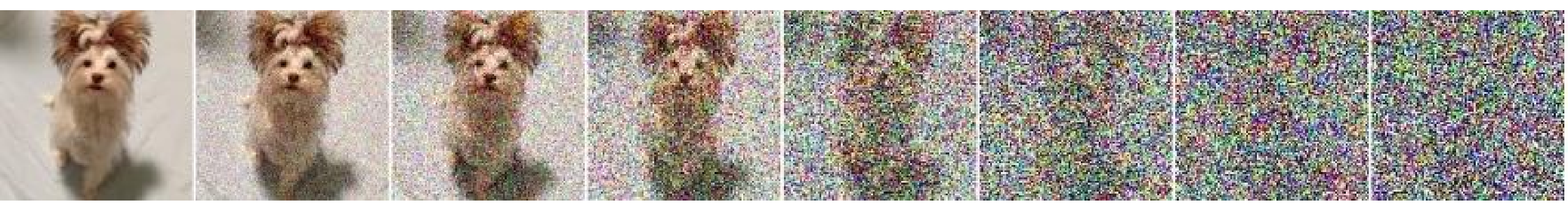
# Variance preserving SDEs (DDPM)

- Positive noise scales $0 < \beta_1 < \beta_2 \cdots < \beta_L < 1$
- In DDPM, the Markov chain is

$$\boldsymbol{x}_i = \sqrt{1-\beta_i}\,\boldsymbol{x}_{i-1} + \sqrt{\beta_i}\,\boldsymbol{z}_{i-1}, \qquad i = 1,2,\cdots,L$$

- To obtain the limit of Markov chain when $L \to \infty$, define an auxiliary set of noise scales $\{\bar{\beta}_i = L\beta_i\}_{i=1}^{L}$ and rewrite $\boldsymbol{x}_i = \sqrt{1-\beta_i}\,\boldsymbol{x}_{i-1} + \sqrt{\beta_i}\,\boldsymbol{z}_{i-1}$ as below

$$\boldsymbol{x}_i = \sqrt{1-\frac{\bar{\beta}_i}{L}}\,\boldsymbol{x}_{i-1} + \sqrt{\frac{\bar{\beta}_i}{L}}\,\boldsymbol{z}_{i-1}, \qquad i = 1,\cdots,L$$



$p_{data}$ $q(\boldsymbol{x}_1)$ $q(\boldsymbol{x}_2)$ $\cdots$ $q(\boldsymbol{x}_L)$

# Variance preserving SDEs (DDPM)

- In the limit of $L \to \infty$, $\{\bar{\beta}_i = L\beta_i\}_{i=1}^{L}$ becomes a function $\beta(t)$ indexed by $t \in [0,1]$
- Let

$$\boldsymbol{x}_{i/L} := \boldsymbol{x}_i, \qquad \beta(i/L) := \bar{\beta}_i, \qquad \boldsymbol{z}(i/L) = \boldsymbol{z}_i$$

- Then we can rewrite the Markov chain Eq.

$$\boldsymbol{x}_i = \sqrt{1 - \frac{\bar{\beta}_i}{L}}\boldsymbol{x}_{i-1} + \sqrt{\frac{\bar{\beta}_i}{L}}\boldsymbol{z}_{i-1}, \qquad i = 1, \cdots, L$$

- as follows with $\Delta t = 1/L$ and $t \in \left\{0, \frac{1}{L}, \cdots, \frac{L-1}{L}\right\}$:

$$\begin{aligned}\boldsymbol{x}_{t+\Delta t} &= \sqrt{1 - \beta(t+\Delta t)\Delta t}\,\boldsymbol{x}_t + \sqrt{\beta(t+\Delta t)\Delta t}\,\boldsymbol{z}_t \\ &\approx \boldsymbol{x}_t - 1/2\beta(t+\Delta t)\Delta t\,\boldsymbol{x}_t + \sqrt{\beta(t+\Delta t)\Delta t}\,\boldsymbol{z}_t \\ &\approx \boldsymbol{x}_t - 1/2\beta(t)\Delta t\,\boldsymbol{x}_t + \sqrt{\beta(t)\Delta t}\,\boldsymbol{z}_t\end{aligned}$$

# Variance preserving SDEs (DDPM)

- In the limit of $\Delta t \to 0$,

$$\boldsymbol{x}_{t+\Delta t} \approx \boldsymbol{x}_t - \frac{1}{2}\beta(t)\Delta t\boldsymbol{x}_t + \sqrt{\beta(t)}\sqrt{\Delta t}\boldsymbol{z}_t$$

- converges to

$$d\boldsymbol{x}_t = -\frac{1}{2}\beta(t)\boldsymbol{x}_t dt + \sqrt{\beta(t)}d\boldsymbol{w}_t$$

- VP SDE yields a process with bounded variance

# Converting the SDE to an ODE

- Let $\{p_t(\boldsymbol{x})\}_{t\in[0,T]}$ be the marginal density functions of the forward-time SDE

$$d\boldsymbol{x}_t = \boldsymbol{f}(\boldsymbol{x}_t, t)dt + g(t)d\boldsymbol{w}_t, \qquad \boldsymbol{x}_0 \sim p_0$$

- and its reverse-time SDE

$$d\boldsymbol{x}_t = \left[\boldsymbol{f}(\boldsymbol{x}_t, t) - g^2(t)\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)\right]dt + g(t)d\overline{\boldsymbol{w}}_t, \qquad \boldsymbol{x}_T \sim p_T$$

- Then $\{p_t(\boldsymbol{x})\}_{t\in[0,T]}$ is also the marginal density function of the following reverse-time **ODE**

$$d\boldsymbol{x}_t = \left[\boldsymbol{f}(\boldsymbol{x}_t, t) - \frac{g^2(t)}{2}\nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)\right]dt, \qquad \boldsymbol{x}_T \sim p_T$$

- This ODE defines a flow model a one-to-one mapping between $\boldsymbol{x}_T$ and $\boldsymbol{x}_0$

# Sampling generation via ODE

- Consider the particular forward-time SDE

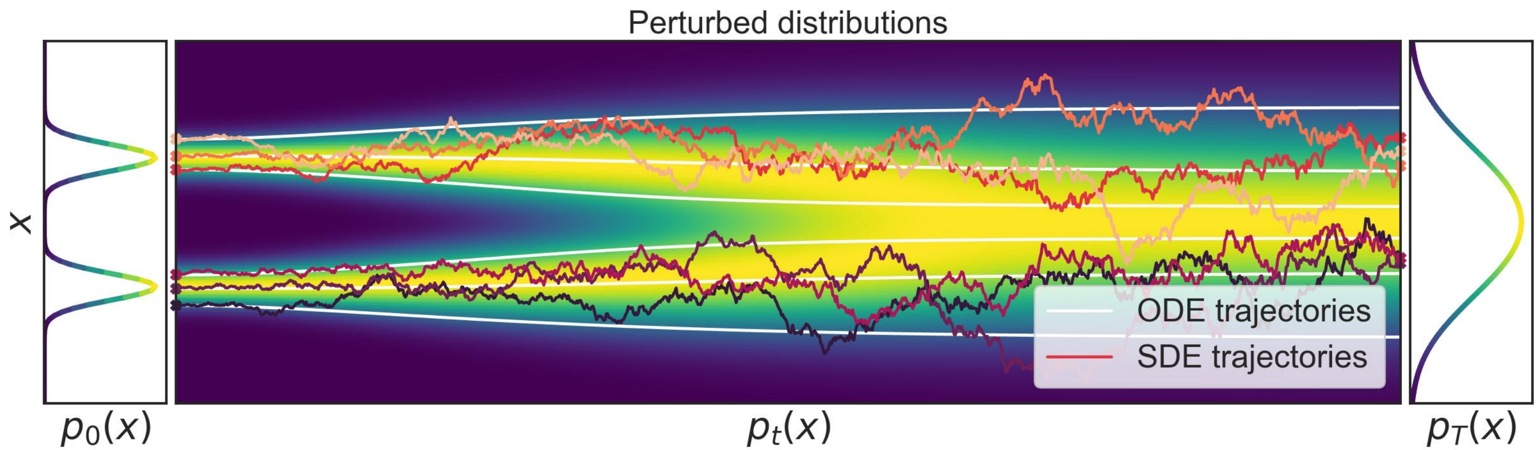
$$d\boldsymbol{x}_t = -\theta \boldsymbol{x}_t dt + \sigma d\boldsymbol{w}_t, \qquad \boldsymbol{x}_0 \sim p_0$$

- If $T$ is sufficiently large, $p_T \sim N(0, \sigma_T^2 I)$

- Consider the reverse-time ODE

$$d\boldsymbol{x}_t = \left(-\theta \boldsymbol{x}_t - \frac{\sigma^2}{2} \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)\right) dt, \qquad \boldsymbol{x}_T \sim p_T$$

# Converting the SDE to an ODE




**SDE**

$$d\boldsymbol{x}_t = -\theta \boldsymbol{x}_t dt + \sigma d\boldsymbol{w}_t$$

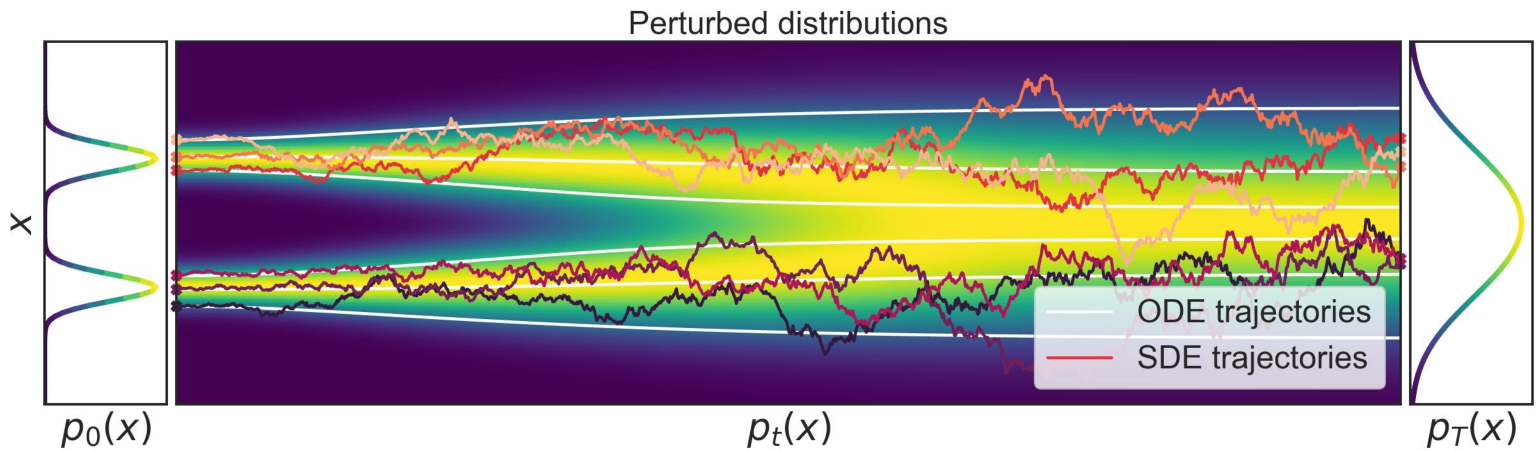
**Ordinary differential equation (ODE)**

$$d\boldsymbol{x}_t = \left(-\theta \boldsymbol{x}_t - \frac{\sigma^2}{2} \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)\right) dt$$

$$\approx \boldsymbol{s}_{\boldsymbol{\theta}}(\mathbf{x}, t)$$

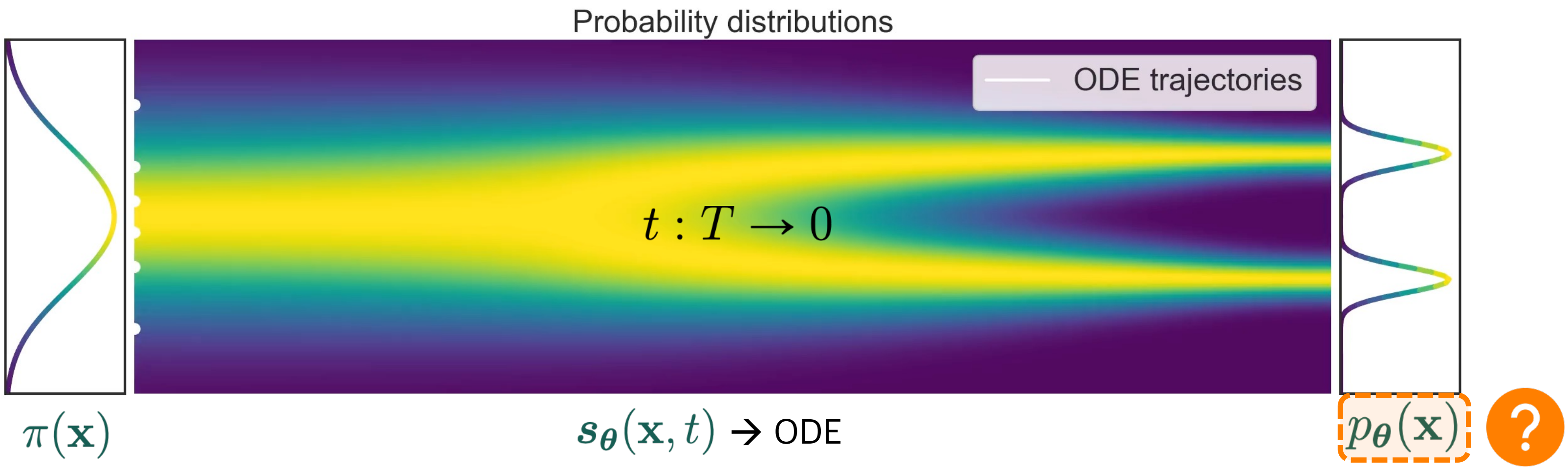
Score function

# Converting the SDE to an ODE




- We can think of this as a (continuous time, infinite depth) normalizing flow
  - Unique ODE solution implies invertible mapping
  - To invert, solve ODE backwards from $T$ to $0$

# Evaluating likelihoods with ODEs (flow model)




**Computing the probability density function (change of variables formula)**

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}_0) = \log \pi(\mathbf{x}_T) - \frac{1}{2}\int_0^T \sigma(t)^2 \operatorname{trace}(\nabla_{\mathbf{x}} \boldsymbol{s}_{\boldsymbol{\theta}}(\mathbf{x}, t))\, \mathrm{d}t$$

ODE solver

Computable in polynomial time

Unbiased estimator

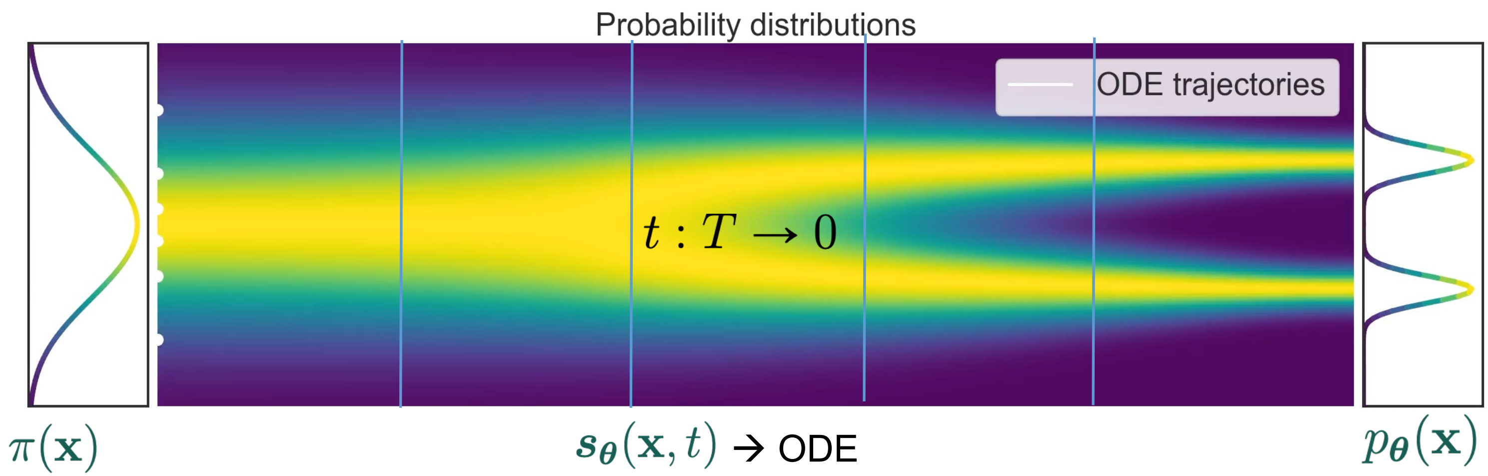
# Competitive likelihoods on test data

Negative log-probability ↓ (**bits/dim**)

| Method | CIFAR-10 | ImageNet 32x32 |
|---|---|---|
| PixelSNAIL [Chen et al. 2018] | 2.85 | 3.80 |
| Delta-VAE [Razavi et al. 2019] | 2.83 | 3.77 |
| Sparse Transformer [Child et al. 2019] | 2.80 | - |

✓ Challenges years of dominance of autoregressive models and VAEs

# Accelerated sampling




- Numerical methods + ODE formulation to accelerate sampling
- DDIM [Song and Ermon, 2021]:
  - Coarsely discretize the time axis, take big steps
  - Corresponds to exponential integrator (semi-linear ODE) [Lu et al, 2022; Zhang and Chen, 2022]
  - 10x-50x speedups, comparable sample quality

# Thanks