Image and Video Generations

Lecture 3: Denoising Diffusion Implicit Models

劉育綸

Yu-Lun (Alex) Liu

Today’s Topics

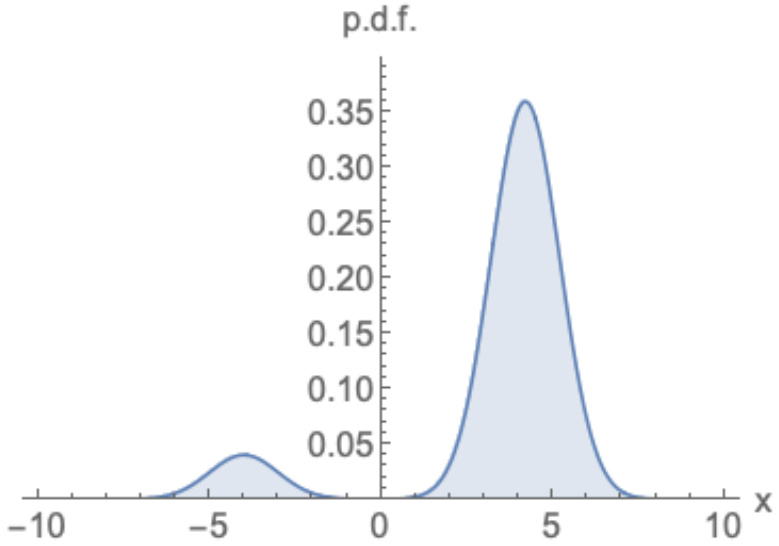
- Connection to Score-Based Models
- Denoising Diffusion Implicit Models (DDIM)

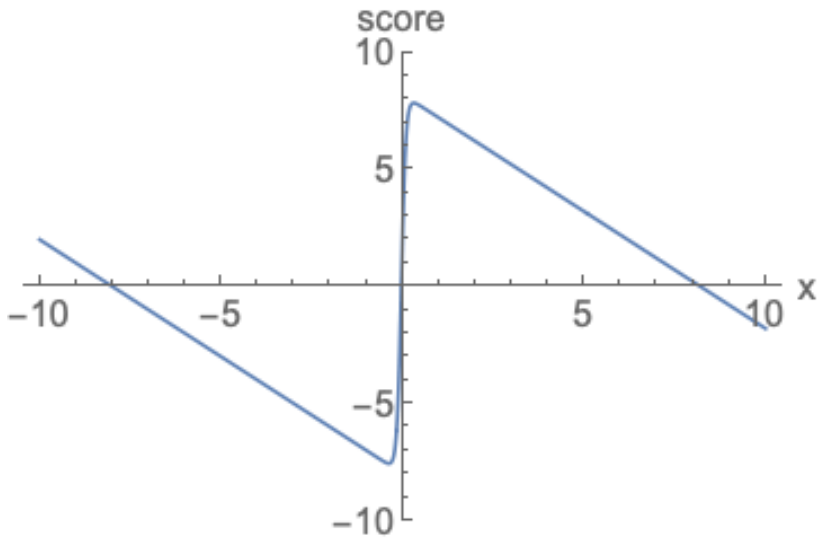
Connection to Score-Based Models

(Stein) Score Function

The (Stein) score (in statistics) is the gradient of the log- likelihood function with respect to a data point:

$$\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x})$$





(Stein) Score Function

- What is the score of $q(\boldsymbol{x}_t|\boldsymbol{x}_0)$: $\nabla_{x_t} \log q(\boldsymbol{x}_t|\boldsymbol{x}_0)$?

- How is it related to $\boldsymbol{\varepsilon}_t$?

(Stein) Score Function

$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}\left(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)\mathbf{I}\right)$$

The score of $q(\boldsymbol{x}_t|\boldsymbol{x}_0)$:

$$\nabla_{\boldsymbol{x}_t} \log q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \nabla_{\boldsymbol{x}_t}\left(-\frac{\left\|\boldsymbol{x}_t - \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0\right\|^2}{2(1-\bar{\alpha}_t)}\right) = -\frac{\boldsymbol{x}_t - \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0}{1-\bar{\alpha}_t}$$

(Stein) Score Function

$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}\left(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)\mathbf{I}\right)$$

$$\boldsymbol{\varepsilon}_t = \frac{1}{\sqrt{1-\bar{\alpha}_t}}\left(\boldsymbol{x}_t - \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0\right)$$

$$\nabla_{\boldsymbol{x}_t} \log q(\boldsymbol{x}_t|\boldsymbol{x}_0) = -\frac{\boldsymbol{x}_t - \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0}{1-\bar{\alpha}_t} = -\frac{\boldsymbol{\varepsilon}_t}{\sqrt{1-\bar{\alpha}_t}}$$

一個能夠預測 noise 的 model，他也是在預測 score！

(Stein) Score Function

$$\nabla_{\boldsymbol{x}_t} \log q(\boldsymbol{x}_t|\boldsymbol{x}_0) = -\frac{\boldsymbol{x}_t - \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0}{1-\bar{\alpha}_t} = -\frac{\boldsymbol{\varepsilon}_t}{\sqrt{1-\bar{\alpha}_t}}$$

The noise predictor $\hat{\boldsymbol{\varepsilon}}_\theta(\boldsymbol{x}_t, t)$ can be interpreted as predicting the score $\nabla_{\boldsymbol{x}_t} \log q(\boldsymbol{x}_t|\boldsymbol{x}_0)$, up to a scaling factor.

一個能夠預測 noise 的 model，他也是在預測 score！

Tweedie's Formula

How to estimate the true mean of a normal distribution from samples drawn from it?

$$\text{For a } \boldsymbol{x} \sim p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}),$$

$$\mathbb{E}[\boldsymbol{\mu}|\boldsymbol{x}] = \boldsymbol{x} + \boldsymbol{\Sigma}\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x})$$

Tweedie's Formula 白話文：
我們只知道一個分布服從**高斯**，**variance** 已知，且我們可以得到資料的 **samples**
要如何求得這個分布的 **mean**?

Tweedie's Formula

$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}\left(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)\mathbf{I}\right)$$

Given the Tweedie's formula,

$$\mathbb{E}[\boldsymbol{\mu}|\boldsymbol{x}_t] = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 = \boldsymbol{x}_t + (1-\bar{\alpha}_t)\nabla_x \log q(\boldsymbol{x}_t|\boldsymbol{x}_0)$$

$$\nabla_{x_t} \log q(\boldsymbol{x}_t|\boldsymbol{x}_0) = -\frac{\boldsymbol{x}_t - \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0}{1-\bar{\alpha}_t} = -\frac{\boldsymbol{\varepsilon}_t}{\sqrt{1-\bar{\alpha}_t}}$$

Tweedie's Formula

$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}\left(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)\mathbf{I}\right)$$

Given the Tweedie's formula,

$$\mathbb{E}[\boldsymbol{\mu}|\boldsymbol{x}_t] = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 = \boldsymbol{x}_t + (1-\bar{\alpha}_t)\nabla_{\boldsymbol{x}} \log q(\boldsymbol{x}_t|\boldsymbol{x}_0)$$

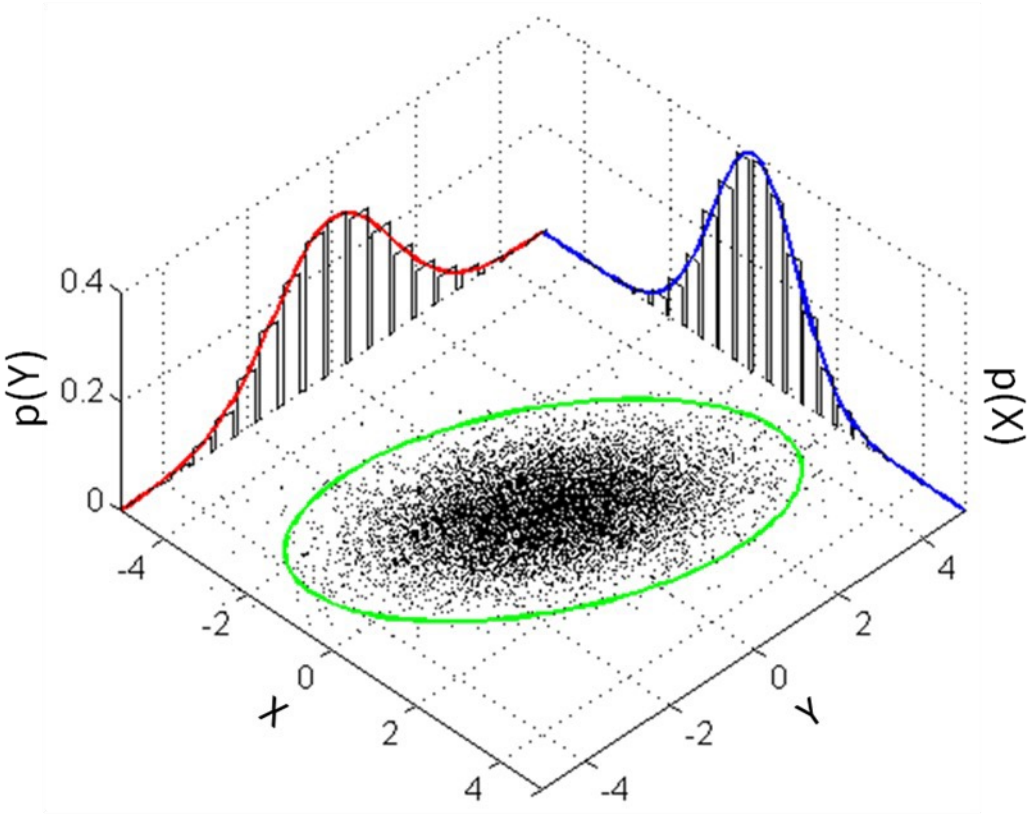
Tweedie's Formula 告訴我們：
有 $\boldsymbol{x}_t$ 與 score (scaled noise)，就可以算出 $\boldsymbol{x}_0$
與 DDPM 的從 $\boldsymbol{x}_t$ 與 noise $\boldsymbol{\varepsilon}_t$ 得到 $\boldsymbol{x}_0$ 的公式是一樣的

Langevin Dynamics

Why is the score function important?

Recall GAN / VAE

If the probability density function (PDF) of the distribution is known, we can sample from it directly.



Langevin Dynamics

Why is the score function important?

Even without knowing $q(\boldsymbol{x})$, if we have the score function $\nabla_{\boldsymbol{x}} \log q(\boldsymbol{x})$, we can sample from the distribution $q(\boldsymbol{x})$ using Langevin dynamics!

知道 $q(\boldsymbol{x})$：通常很困難，特別是高維空間
知道 $\nabla_{\boldsymbol{x}} \log q(\boldsymbol{x})$ ：相對容易，network 可以學習

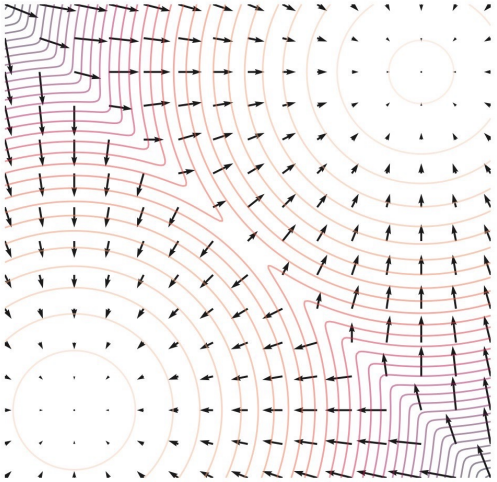
Langevin Dynamics

Similar to the Reverse process network!

- Sample $\boldsymbol{x}$ from a prior distribution.
- Iterate the following procedure T steps:

$$\boldsymbol{x} \leftarrow \boldsymbol{x} + \eta \nabla_{\boldsymbol{x}} \log q(\boldsymbol{x}) + \sqrt{2\eta}\boldsymbol{\varepsilon}, \qquad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

- It converges to $q(\boldsymbol{x})$ when $\eta \to 0$ and $T \to \infty$.
- Note that it only uses the score information $\nabla_{\boldsymbol{x}} \log q(\boldsymbol{x})$, not $q(\boldsymbol{x})$.

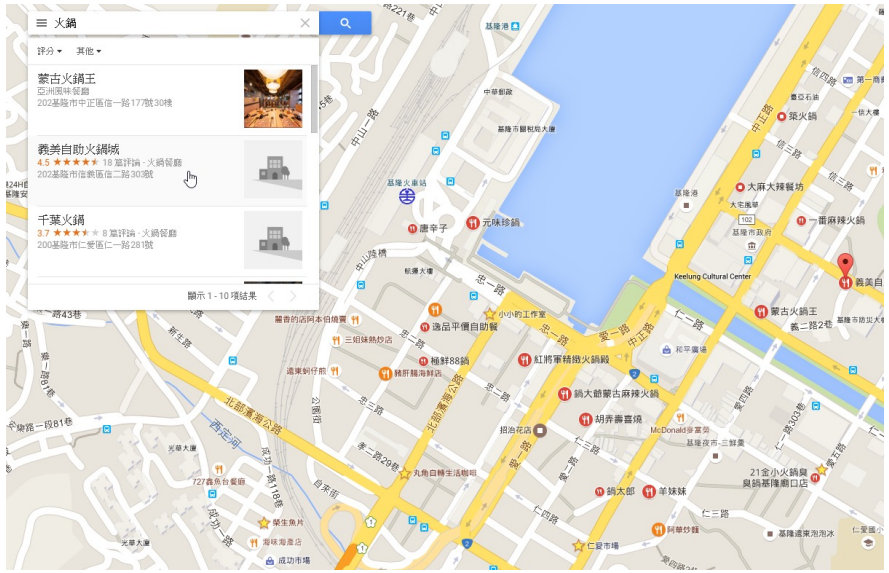


https://yang-song.net/blog/2021/score/

想像你要在城市裡找餐廳

需要完整分佈 $q(\boldsymbol{x})$

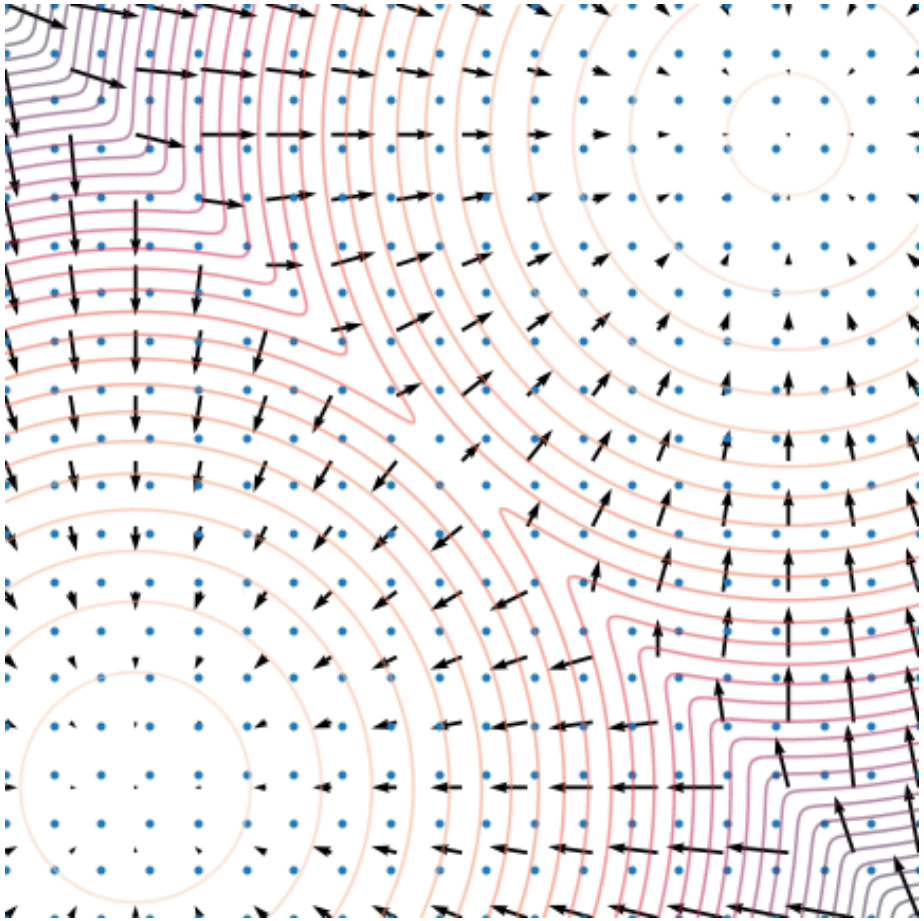
有一張標記所有餐廳位置和評分的地圖，直接去高分餐廳



只需要 score function $\nabla_{\boldsymbol{x}} \log q(\boldsymbol{x})$

沒有地圖，但每個路口都有指示牌：「往這個方向餐廳更好」，跟著指示牌走 + 偶爾隨機探索，最終也能找到好餐廳

Langevin Dynamics



https://yang-song.net/blog/2021/score/

Score Matching

Similar to the noise prediction network!

One possible way to encode the data distribution $q(\boldsymbol{x})$ into a neural network is to train a score prediction network $s_\theta(\boldsymbol{x})$ using the following loss function:

$$\mathbb{E}_{\boldsymbol{x}\sim q(\boldsymbol{x})}[\|s_\theta(\boldsymbol{x}) - \nabla_{\boldsymbol{x}} \log q(\boldsymbol{x})\|^2]$$

you

Noise-Conditional Score-Based Models

How do we compute $\nabla_{\boldsymbol{x}} \log q(\boldsymbol{x})$ when we only have samples of $q(\boldsymbol{x})$?

Now let $q(\boldsymbol{x}) = q(\boldsymbol{x}_0)$.

原始 Score Matching 的問題：如果直接在原始數據上學習 score function $\nabla_{\boldsymbol{x}} \log q(\boldsymbol{x})$，會遇到低密度區域的嚴重問題（下面會講）

Approximate $q(\boldsymbol{x}_0)$ as a mixture of Gaussians:

$$q(\boldsymbol{x}_t) = \int q(\boldsymbol{x}_0) q(\boldsymbol{x}_t|\boldsymbol{x}_0)\, d\boldsymbol{x}$$

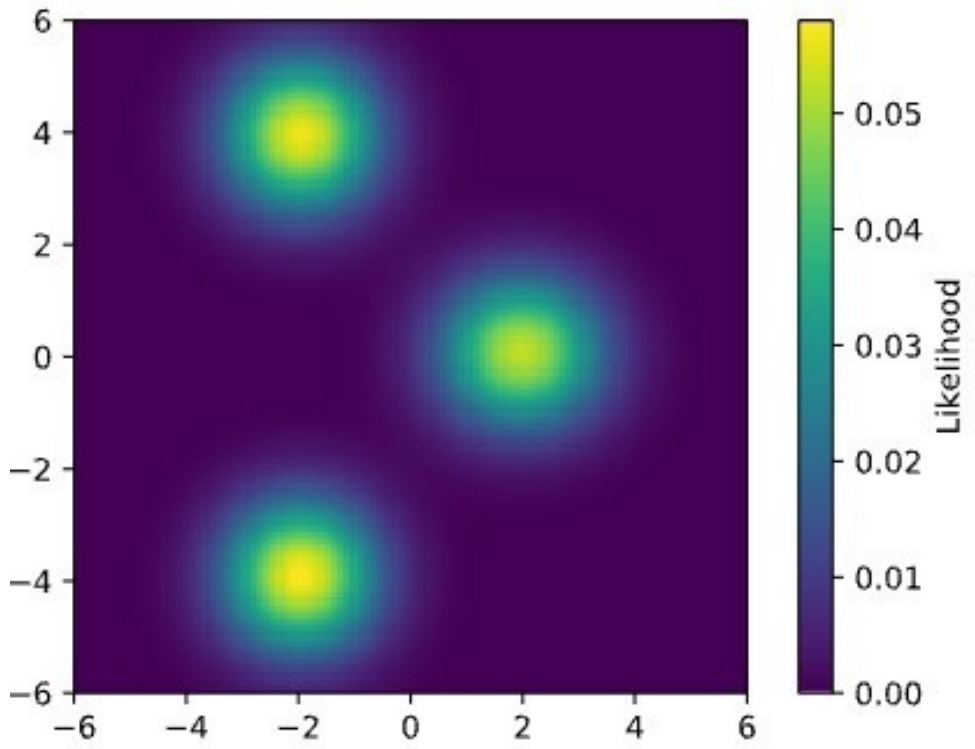
where $q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t; \boldsymbol{x}_0, \sigma_t^2 \mathbf{I})$.

通過加入不同程度的 noise，我們創建了一系列「模糊」版本的數據分佈

Noise-Conditional Score-Based Models

$$q(\boldsymbol{x}_t) = \int q(\boldsymbol{x}_0) q(\boldsymbol{x}_t|\boldsymbol{x}_0)\, d\boldsymbol{x}$$

where $q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t; \boldsymbol{x}_0, \sigma_t^2 \mathbf{I})$.



Noise-Conditional Score-Based Models

$$q(\boldsymbol{x}_t) = \int q(\boldsymbol{x}_0) q(\boldsymbol{x}_t|\boldsymbol{x}_0)\, d\boldsymbol{x}$$

Sampling from $q(\boldsymbol{x}_t)$ is the same as

1. sampling from $q(\boldsymbol{x}_0)$ (taking a random $\boldsymbol{x}_0$) and then

2. sampling from $q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t; \boldsymbol{x}_0, \sigma_t^2 \mathbf{I})$.

Noise-Conditional Score-Based Models

$$\mathbb{E}_{\boldsymbol{x}_0 \sim q(\boldsymbol{x}_0)} \left[\left\| s_\theta(\boldsymbol{x}_t) - \nabla_{\boldsymbol{x}_t} \log q(\boldsymbol{x}_t) \right\|^2 \right]$$

$$= \cdots$$

$$= \mathbb{E}_{\boldsymbol{x}_0 \sim q(\boldsymbol{x}_0), \boldsymbol{x}_t \sim q(\boldsymbol{x}_t|\boldsymbol{x}_0)} \left[\left\| s_\theta(\boldsymbol{x}_t) - \nabla_{\boldsymbol{x}_t} \log q(\boldsymbol{x}_t|\boldsymbol{x}_0) \right\|^2 \right]$$

$$= \mathbb{E}_{\boldsymbol{x}_0 \sim q(\boldsymbol{x}_0), \boldsymbol{x}_t \sim q(\boldsymbol{x}_t|\boldsymbol{x}_0)} \left[\left\| s_\theta(\boldsymbol{x}_t) + \frac{\boldsymbol{\epsilon}_t}{\sqrt{1 - \bar{\alpha}_t}} \right\|^2 \right]$$

Identical to the loss function of DDPM, up to scale!

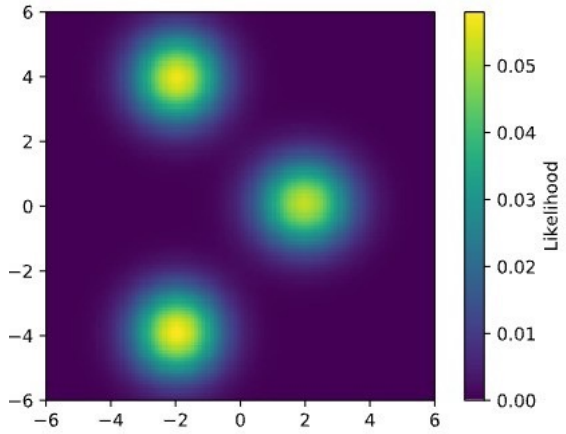
預測 score（資料分布的 gradient）和預測 noise（加在資料上的雜訊）本質上是等價的

https://yang-song.net/blog/2021/score/

Noise-Conditional Score-Based Models

$$q(\boldsymbol{x}_t) = \int q(\boldsymbol{x}_0)\mathcal{N}(\boldsymbol{x}_t; \boldsymbol{x}_0, \sigma_t^2 \mathbf{I})\, d\boldsymbol{x}$$

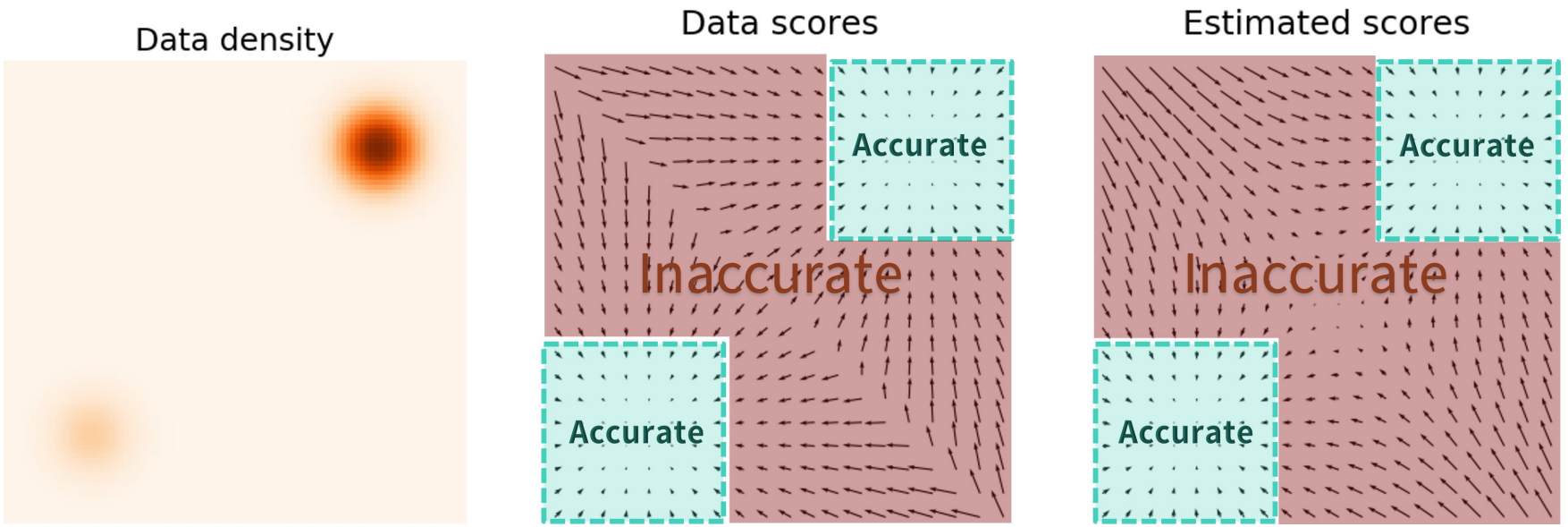
As σ_t is small,



- (+) Close to the given data samples.
- (-) May lead to low density regions where the score prediction could be inaccurate.

Noise-Conditional Score-Based Models

$$\mathbb{E}_{\boldsymbol{x}_0 \sim q(\boldsymbol{x}_0)}\left[\left\|\nabla_{\boldsymbol{x}_t} \log q(\boldsymbol{x}_0) - s_\theta(\boldsymbol{x}_t)\right\|^2\right]$$



https://yang-song.net/blog/2021/score/

Noise-Conditional Score-Based Models

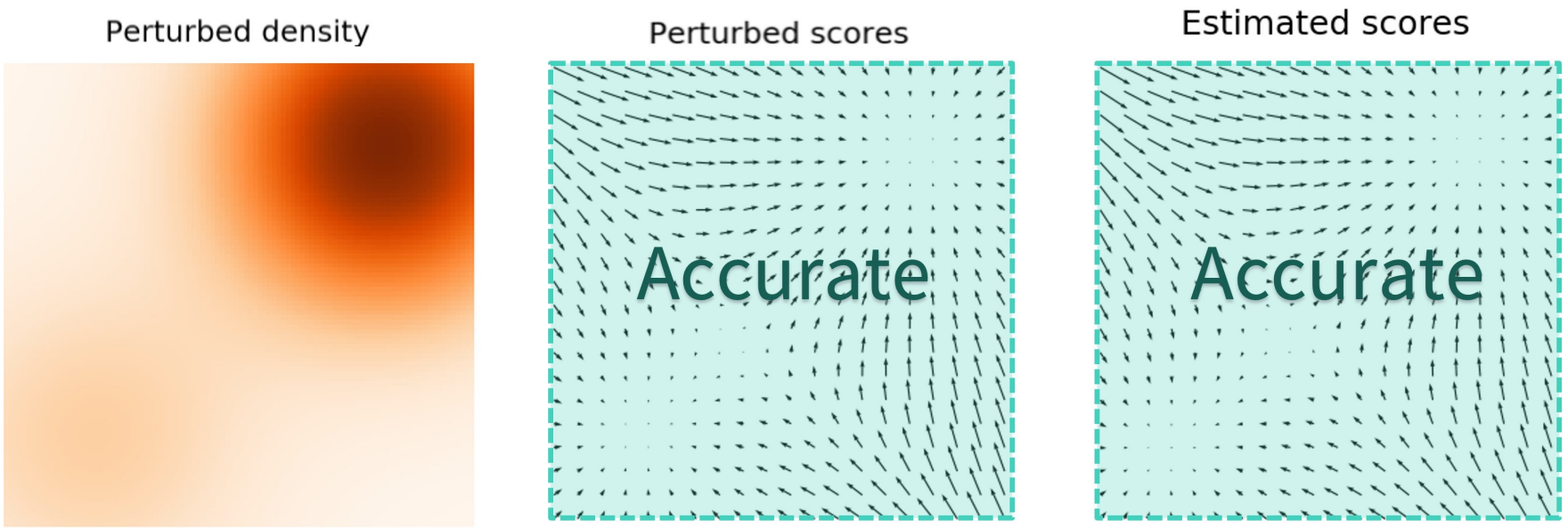
$$q(\boldsymbol{x}_t) = \int q(\boldsymbol{x}_0)\mathcal{N}(\boldsymbol{x}_t; \boldsymbol{x}_0, \sigma_t^2 \mathbf{I})\, d\boldsymbol{x}$$

As σ_t is large,

- (+) Can help avoid low-density regions.
- (-) May over-corrupt the original data distribution, also leading to noisy score predictions.

Noise-Conditional Score-Based Models

$$\mathbb{E}_{\boldsymbol{x}_0 \sim q(\boldsymbol{x}_0)}\left[\left\|\nabla_{\boldsymbol{x}_t} \log q(\boldsymbol{x}_0) - s_\theta(\boldsymbol{x}_t)\right\|^2\right]$$



https://yang-song.net/blog/2021/score/

Noise-Conditional Score-Based Models

As σ_t is small,

- (+) Score prediction become more accurate in high-density regions.
- (-) Score prediction become less accurate in low-density regions.

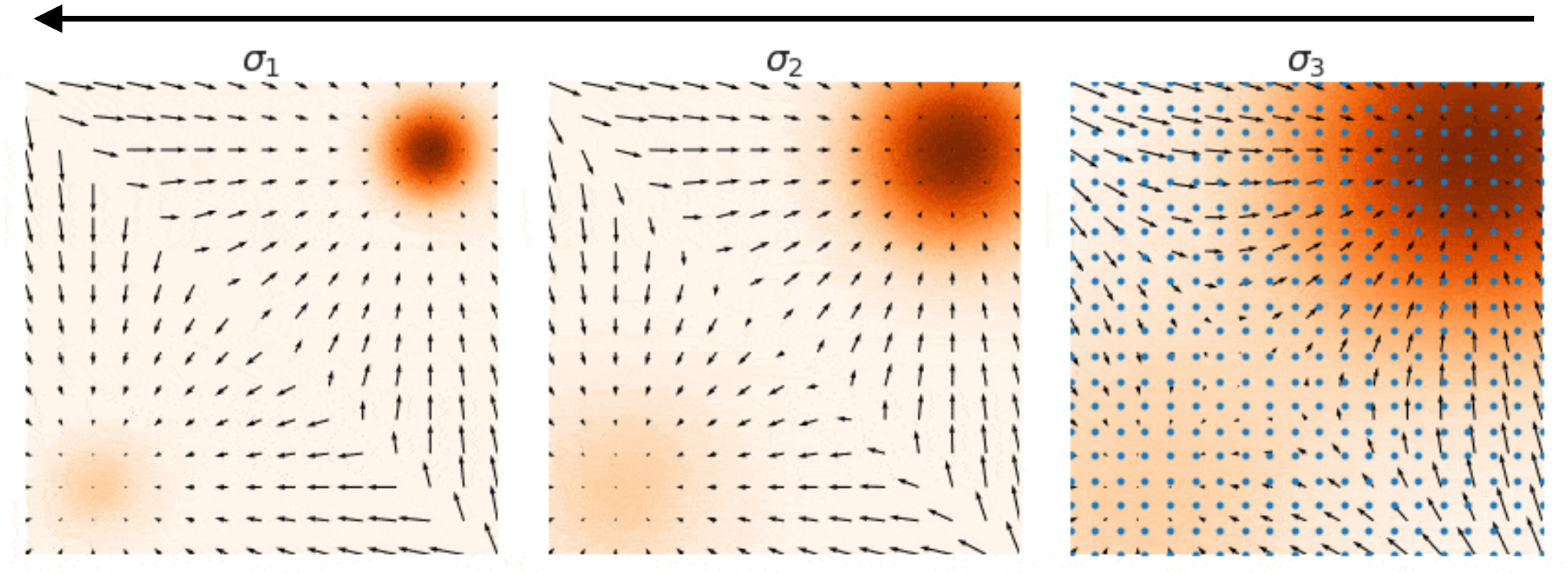
As σ_t is large,

- (+) Score prediction become relatively more accurate in low-density regions.
- (-) Score prediction become relatively less accurate in high-density regions.

Annealed Langevin Dynamics

Same as the reverse diffusion process!

Solution: Gradually decrease (anneals) σ_t over time!



https://yang-song.net/blog/2021/score/

Annealed Langevin Dynamics

- At the beginning (when σ_t is large),
  we can make good predictions of scores across the entire space,
  pushing the samples toward the high-density regions of the data distribution.

- As time progresses (and σ_t decreases),
  we achieve better predictions of scores specifically in these high-density regions, allowing for more accurate performance of Langevin dynamics.

Annealed Langevin Dynamics

Train the score prediction network $s_\theta(\boldsymbol{x}_t)$ while varying the Gaussian noise σ_t:

$$\mathbb{E}_{\boldsymbol{x}_0 \sim q(\boldsymbol{x}_0), t>1, \boldsymbol{x}_t \sim q(\boldsymbol{x}_t|\boldsymbol{x}_0)} \left[\left\| s_\theta(\boldsymbol{x}_t, t) - \nabla_{\boldsymbol{x}_t} \log q(\boldsymbol{x}_t|\boldsymbol{x}_0) \right\|^2 \right]$$

DDPM 的反向去噪過程 = Annealed Langevin Dynamics

DDPM 與 Score-based models 這兩個看似不同的方法其實是同一個框架

DDPM 提供了 discrete implementation，Score-based models 則提供了 continuous-time 的理解

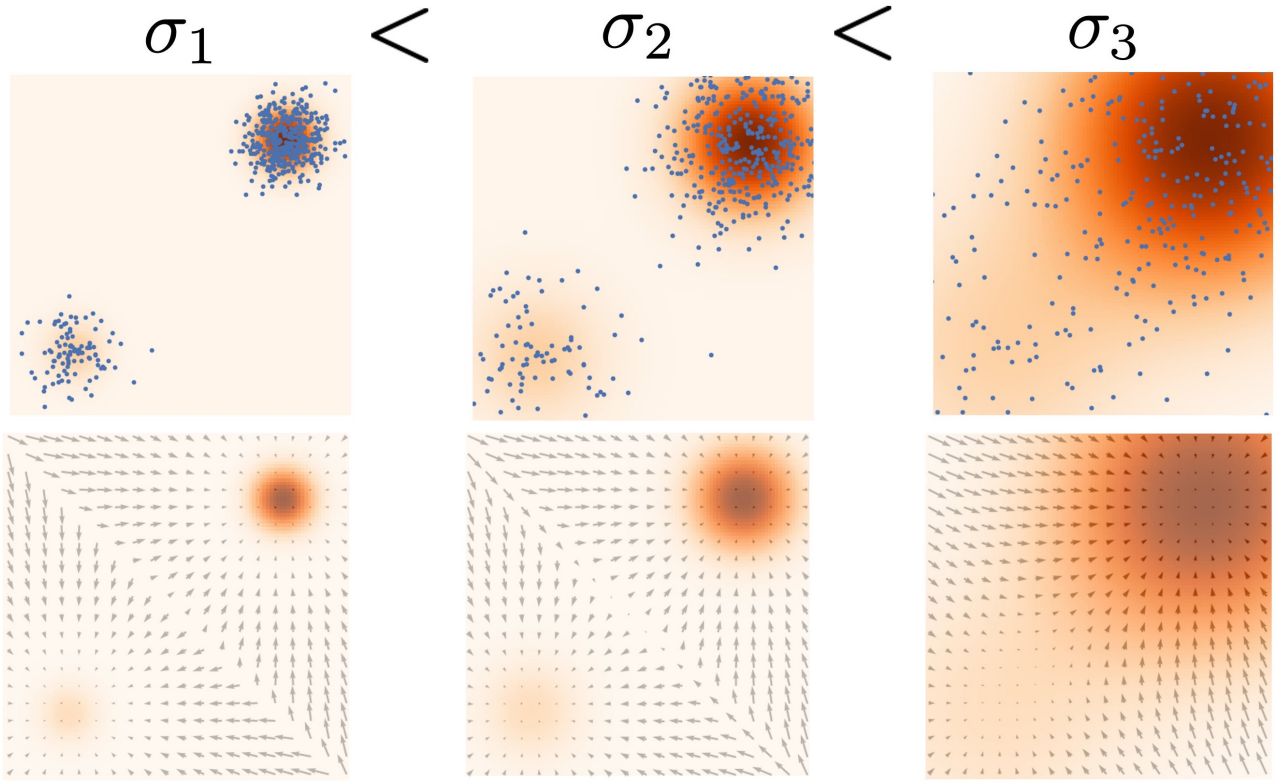
DDPM 不只是「去噪」，它實際上在學習 data distribution 的 gradient fields (score)，反向去噪過程是沿著 gradient 往上爬到高機率的區域

https://yang-song.net/blog/2021/score/

Annealed Langevin Dynamics

Same as the forward diffusion process!

The reverse of the annealed Langevin dynamics (the forward process) can be seen as a gradual data perturbation.



DDPM 的前向加噪過程和 Annealed Langevin Dynamics 的「反向」其實是同一件事

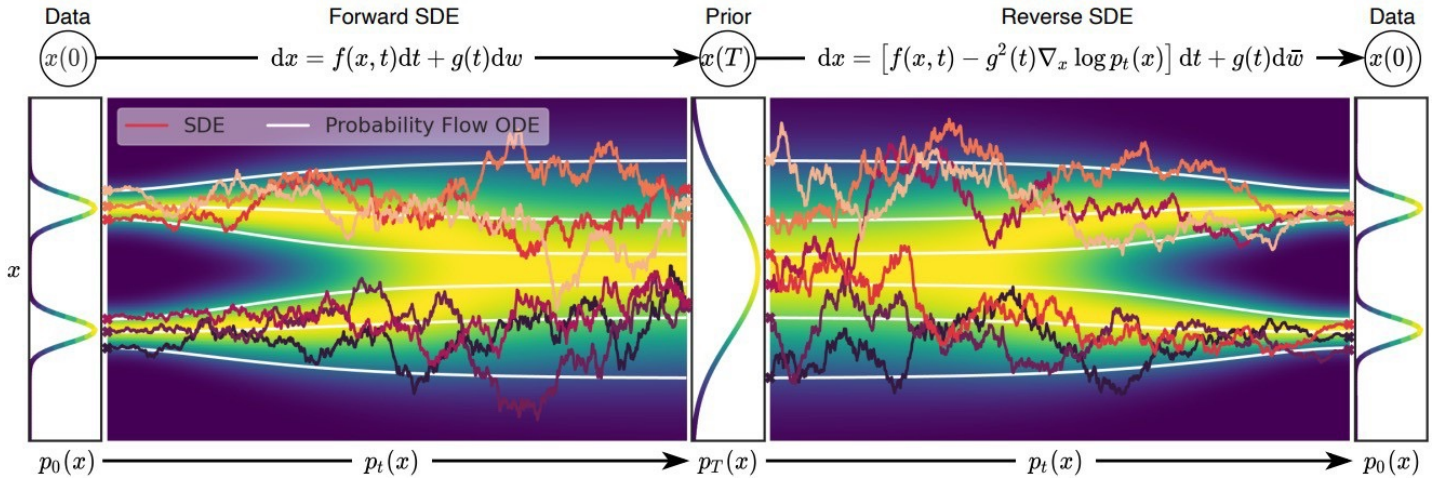
https://yang-song.net/blog/2021/score/

Stochastic Differential Equations

In a continuous time domain, the data perturbation (forward) process is described by the following stochastic differential equation (SDE):

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

- $\mathbf{f}(\boldsymbol{x}, t)$: Drift coefficient
- $g(t)$: Diffusion coefficient
- $d\boldsymbol{w}$: Infinitesimal white noise (called Brownian motion)



Stochastic Differential Equations

Its reverse process is also formulated as another stochastic differential equation:

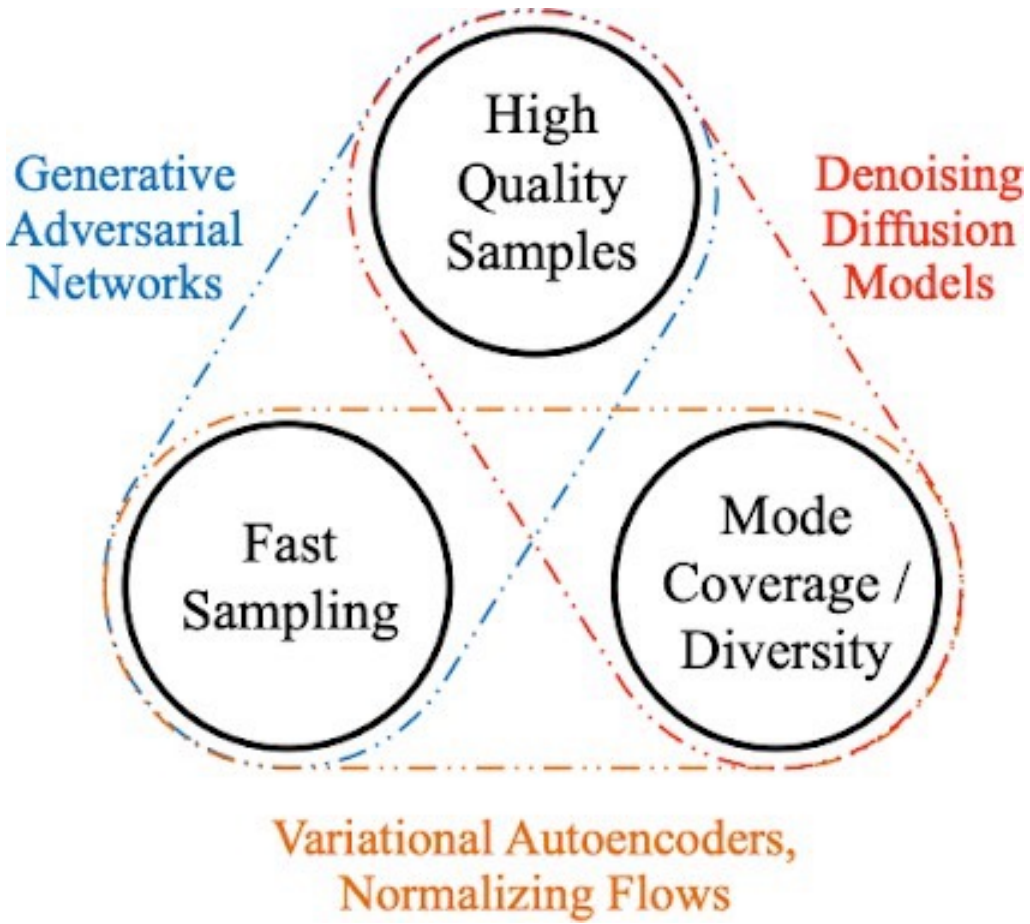
$$d\boldsymbol{x} = [\mathbf{f}(\boldsymbol{x}, t)dt - g^2(t)\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x})]dt + g(t)d\boldsymbol{w}$$

DDPM is a specific discretization of the SDE formulations.

Denoising Diffusion Implicit Models (DDIM)

Song et al., Denoising Diffusion Implicit Models, ICLR 2021.

Diffusion Models – Pros and Cons



Diffusion Models – Pros and Cons

- (+) High quality samples
- (+) Diversity
- (-) Very slow speed

How to speed up the reverse process while achieving good quality in generation?

Denoising Diffusion Implicit Models (DDIM)

Key features of DDPM:

The sequential forward and reverse processes are Markovian processes:

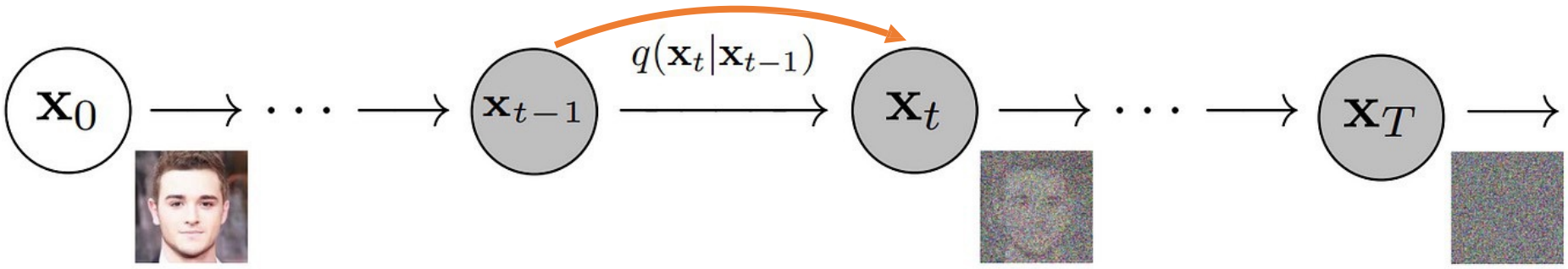
$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{x}_0)$$

Can we consider a non-Markovian reverse process?

Denoising Diffusion Implicit Models (DDIM)

DDPM's Markovian forward process:

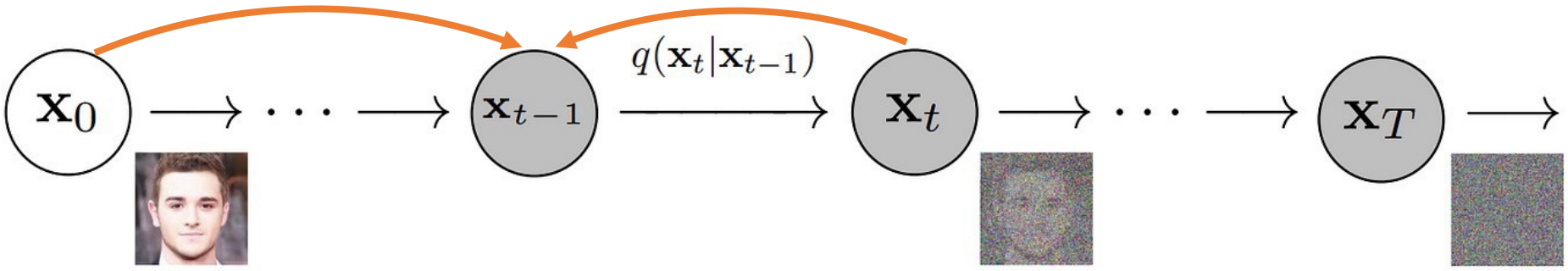
$$q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0) = \prod_{t=1}^{T} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$$



Denoising Diffusion Implicit Models (DDIM)

DDIM's non-Markovian forward process:

$$q_\sigma(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0) = q_\sigma(\boldsymbol{x}_T|\boldsymbol{x}_0)\prod_{t=2}^{T} q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$$



Denoising Diffusion Implicit Models (DDIM)

$$q_\sigma(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0) = \underbrace{q_\sigma(\boldsymbol{x}_T|\boldsymbol{x}_0)}_{(1)} \prod_{t=2}^{T} \underbrace{q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)}_{(2)}$$

In the forward process,

1. $\boldsymbol{x}_T$ is sampled from $\boldsymbol{x}_0$ first.

原本 DDPM 的正向加噪每步只看前一步
現在 DDIM 的正向加噪先得到 $\boldsymbol{x}_T$，
每步都去看原點 $\boldsymbol{x}_0$

2. Each $\boldsymbol{x}_{t-1}$ is sampled from $\boldsymbol{x}_t$ and $\boldsymbol{x}_0$ in a reverse manner!

注意這個 non-Markovian forward process 只是理論工具，用來推導新的反向去噪過程公式，實際訓練時根本**不使用**這個正向加噪過程

Denoising Diffusion Implicit Models (DDIM)

- How to define $q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$?

- Let $q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$ be a Gaussian distribution with a linear mean function and variance σ_t^2:

$$q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{N}(\omega_0 \boldsymbol{x}_0 + \omega_t \boldsymbol{x}_t + b, \sigma_t^2 \mathbf{I})$$

- How to determine ω_0, ω_t, and b?

Q. 為什麼 mean 要是 $\boldsymbol{x}_0$ 與 $\boldsymbol{x}_t$ 的線性組合加上一個 bias?

Denoising Diffusion Implicit Models (DDIM)

- How to define $q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$?

- Let $q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$ be a Gaussian distribution with a linear mean function and variance σ_t^2:

$$q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{N}(\omega_0 \boldsymbol{x}_0 + \omega_t \boldsymbol{x}_t + b, \sigma_t^2 \mathbf{I})$$

- How to determine ω_0, ω_t, and b?

A. $\boldsymbol{x}_{t-1}$ 必須能從 $\boldsymbol{x}_t$ 和 $\boldsymbol{x}_0$ 推導出來，且要保持是一個高斯分佈，就ㄈ只能是線性變換

Key Idea

How to determine ω_0, ω_t, and b?

We want to ensure that $q_\sigma(\boldsymbol{x}_t|\boldsymbol{x}_0)$ remains the same as in DDPM:

$$q_\sigma(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}\left(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)\mathbf{I}\right)$$

正向跳躍

Q. 為什麼在設計 DDIM 的時候會希望這個正向跳躍要維持與 DDPM 相同？

Key Idea

How to determine ω_0, ω_t, and b?

We want to ensure that $q_\sigma(\boldsymbol{x}_t|\boldsymbol{x}_0)$ remains the same as in DDPM:

$$q_\sigma(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}\left(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)\mathbf{I}\right)$$

正向跳躍

A. 因為訓練目標完全一樣，所以可以直接重用已訓練好的 DDPM 模型/權重，不需要重新訓練網路，讓 DDIM 成為 DDPM 的「即插即用」加速升級版本

Key Idea

Consider an induction. 歸納法

When

$$q_\sigma(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}\left(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)\mathbf{I}\right),$$

what should ω_0, ω_t, and b be in order to ensure that

$$q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_0) = \mathcal{N}\left(\sqrt{\bar{\alpha}_{t-1}}\boldsymbol{x}_0, (1-\bar{\alpha}_{t-1})\mathbf{I}\right)?$$

Key Idea

Hint 1

$$q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{N}(\omega_0 \boldsymbol{x}_0 + \omega_t \boldsymbol{x}_t + b, \sigma_t^2 \mathbf{I})$$

$$q_\sigma(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}\left(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)\mathbf{I}\right)$$

Marginalization:

$$q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_0) = \int q_\sigma(\boldsymbol{x}_t|\boldsymbol{x}_0)\, q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) d\boldsymbol{x}_t$$

透過對 $\boldsymbol{x}_t$ 積分把它邊緣化掉

Key Idea

Hint 2

When $p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{\mu}, \sigma_{\boldsymbol{x}}^2 \mathbf{I})$ and $p(\boldsymbol{y}|\boldsymbol{x}) = \mathcal{N}\left(a\boldsymbol{x} + b, \sigma_{\boldsymbol{y}}^2 \mathbf{I}\right)$

$$p(\boldsymbol{y}) = \int p(\boldsymbol{x}) p(\boldsymbol{y}|\boldsymbol{x}) d\boldsymbol{x}$$

$$= \mathcal{N}\left(a\boldsymbol{\mu} + b, \left(\sigma_{\boldsymbol{y}}^2 + a^2 \sigma_{\boldsymbol{x}}^2\right)\mathbf{I}\right)$$

Bishop, Christopher M., and Nasser M. Nasrabadi. Pattern recognition and machine learning. Vol. 4. No. 4. New York: springer, 2006.

Key Idea

Q1. What are the mean and variance $q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_0)$ with respect to ω_0, ω_t, and b?

Q2. What are ω_0, ω_t, and b?

Key Idea

A1. $q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_0) =$

$$\mathcal{N}\left(\omega_0\boldsymbol{x}_0 + \omega_t\left(\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0\right) + b, \left(\sigma_t^2 + \omega_t^2(1-\bar{\alpha}_t)\right)\mathbf{I}\right)$$

$$\mathcal{N}\left(\sqrt{\bar{\alpha}_{t-1}}\boldsymbol{x}_0, (1-\bar{\alpha}_{t-1})\mathbf{I}\right)$$

Key Idea

A2. $\omega_t = \sqrt{\frac{1-\bar{\alpha}_{t-1}-\sigma_t^2}{1-\bar{\alpha}_t}}$

$$\omega_0 = \sqrt{\bar{\alpha}_{t-1}} - \sqrt{\bar{\alpha}_t}\sqrt{\frac{1-\bar{\alpha}_{t-1}-\sigma_t^2}{1-\bar{\alpha}_t}}$$

$$b = 0$$

$$\therefore q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{N}(\omega_0 \boldsymbol{x}_0 + \omega_t \boldsymbol{x}_t + b, \sigma_t^2 \mathbf{I})$$

$$= \mathcal{N}\left(\sqrt{\bar{\alpha}_{t-1}}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2} \cdot \frac{\boldsymbol{x}_t - \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0}{\sqrt{1-\bar{\alpha}_t}}, \sigma_t^2 \mathbf{I}\right)$$

Denoising Diffusion Implicit Models (DDIM)

$$q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0)=\mathcal{N}\left(\sqrt{\bar{\alpha}_{t-1}}\boldsymbol{x}_0+\sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\cdot\frac{\boldsymbol{x}_t-\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0}{\sqrt{1-\bar{\alpha}_t}},\sigma_t^2\mathbf{I}\right)$$

with arbitrary σ_t^2 guarantees that $q_\sigma(\boldsymbol{x}_t,\boldsymbol{x}_0)$ remains the same as in DDPM!

任何 variance 都能保證前向跳躍保持與 DDPM 相同

DDPM vs. DDIM

DDPM 透過馬可夫過程定義

- $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ is defined.
- $q(\boldsymbol{x}_t|\boldsymbol{x}_0)$ and $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$ are derived from $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$.

遞迴串起來的正向跳躍　線性內插

DDIM 為了讓正向跳躍與 DDPM 相同而定義/推導出來

- $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$ is defined.
- $q(\boldsymbol{x}_t|\boldsymbol{x}_0)$ and $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ are derived from $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$.

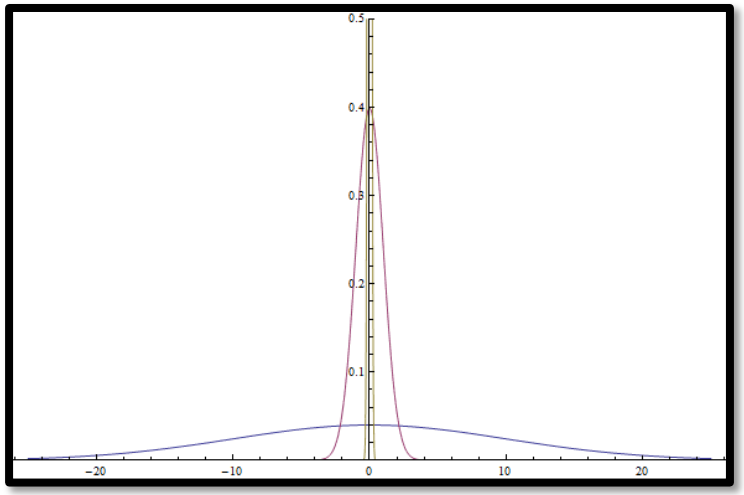
與 DDPM 相同的正向跳躍

Denoising Diffusion Implicit Models (DDIM)

$$q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{N}\left(\sqrt{\bar{\alpha}_{t-1}}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\cdot\frac{\boldsymbol{x}_t - \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0}{\sqrt{1-\bar{\alpha}_t}}, \sigma_t^2\mathbf{I}\right)$$

[Important] What if we set $\sigma_t^2 = 0$?

Then the forward and reverse processes become deterministic!



Loss Function

- The ELBO remains unchanged from DDPM, up to a constant.

- The variational likelihood distribution $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ in the reverse process is learned by minimizing the same loss function as in DDPM:

$$\mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}[D_{KL}(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0)\|p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t))]$$

Loss Function

[Important] No need to retrain the noise predictor!

- The noise predictor $\hat{\boldsymbol{\varepsilon}}_\theta(\boldsymbol{x}_t, t)$ trained for DDPM can be directly used in the DDIM reverse process!

- Using the same noise predictor $\hat{\boldsymbol{\varepsilon}}_\theta(\boldsymbol{x}_t, t)$, you can choose to perform either the DDPM or the DDIM reverse process.

DDPM Reverse Process

$$q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0)=\mathcal{N}\left(\frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{x}_t-\frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{x}_0,\left(\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t\right)\mathbf{I}\right)$$

For each time step $t=T,\dots,1$, repeat:

1. Compute $\boldsymbol{x}_{0|t}=\frac{1}{\sqrt{\bar{\alpha}_t}}\left(\boldsymbol{x}_t-\sqrt{1-\bar{\alpha}_t}\hat{\boldsymbol{\varepsilon}}_\theta(\boldsymbol{x}_t,t)\right)$
2. Compute $\tilde{\mu}=\frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{x}_t-\frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{x}_{0|t}$.
3. Sample $\boldsymbol{z}_t\sim\mathcal{N}(\mathbf{0},\mathbf{I})$.
4. Compute $\boldsymbol{x}_{t-1}=\tilde{\mu}+\sqrt{\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t}\boldsymbol{z}_t$.

Recap: mean 是線性內插
variance 是故意設計成與 posterior 一致

DDIM Reverse Process

$$q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0)=\mathcal{N}\left(\sqrt{\bar{\alpha}_{t-1}}\boldsymbol{x}_0+\sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\cdot\frac{\boldsymbol{x}_t-\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0}{\sqrt{1-\bar{\alpha}_t}},\sigma_t^2\mathbf{I}\right)$$

For each time step $t=T,\dots,1$, repeat:

1. Compute $\boldsymbol{x}_{0|t}=\frac{1}{\sqrt{\bar{\alpha}_t}}\left(\boldsymbol{x}_t-\sqrt{1-\bar{\alpha}_t}\hat{\boldsymbol{\varepsilon}}_\theta(\boldsymbol{x}_t,t)\right)$

From Tweedie's formula
並帶入 score 與 noise 的關係

2. Compute $\tilde{\mu}=\sqrt{\bar{\alpha}_{t-1}}\boldsymbol{x}_{0|t}+\sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\cdot\frac{\boldsymbol{x}_t-\sqrt{\bar{\alpha}_t}\boldsymbol{x}_{0|t}}{\sqrt{1-\bar{\alpha}_t}}$. 只要從 DDPM 改一行 code

3. Sample $\boldsymbol{z}_t\sim\mathcal{N}(\mathbf{0},\mathbf{I})$.

4. Compute $\boldsymbol{x}_{t-1}=\tilde{\mu}+\sigma_t\boldsymbol{z}_t$.

DDPM vs. DDIM

Q. What is $\tilde{\mu}$ with respect to $\boldsymbol{x}_t$ and $\boldsymbol{\varepsilon}_t$ for both DDPM and DDIM cases?

DDPM vs. DDIM

$$q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0)=\mathcal{N}\left(\frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t-\frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\varepsilon}_t\right),\left(\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t\right)\mathbf{I}\right)$$

For each time step $t=T,\ldots,1$, repeat:

1. Compute $\tilde{\mu}=\frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{x}_t-\frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\hat{\boldsymbol{\varepsilon}}_\theta(\boldsymbol{x}_t,t)\right)$.

   也可以不要外推 $\boldsymbol{x_0}$
   直接計算 mean 就好

2. Sample $\boldsymbol{z}_t\sim\mathcal{N}(\mathbf{0},\mathbf{I})$.

3. Compute $\boldsymbol{x}_{t-1}=\tilde{\mu}+\sqrt{\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t}\boldsymbol{z}_t$.

DDPM vs. DDIM

$$q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0)=\mathcal{N}\left(\sqrt{\bar{\alpha}_{t-1}}\left(\frac{\boldsymbol{x}_t-\sqrt{1-\bar{\alpha}_t}\boldsymbol{\varepsilon}_t}{\sqrt{\bar{\alpha}_t}}\right)+\sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\cdot\boldsymbol{\varepsilon}_t,\sigma_t^2\mathbf{I}\right)$$

For each time step $t=T,\ldots,1$, repeat:

也可以不要外推 $\boldsymbol{x_0}$
直接計算 mean 就好

1. Compute $\tilde{\mu}=\sqrt{\bar{\alpha}_{t-1}}\left(\frac{\boldsymbol{x}_t-\sqrt{1-\bar{\alpha}_t}\boldsymbol{\varepsilon}_t}{\sqrt{\bar{\alpha}_t}}\right)+\sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\cdot\hat{\boldsymbol{\varepsilon}}_\theta(\boldsymbol{x}_t,t)$.
2. Sample $\boldsymbol{z}_t\sim\mathcal{N}(\mathbf{0},\mathbf{I})$.
3. Compute $\boldsymbol{x}_{t-1}=\tilde{\mu}+\sigma_t\boldsymbol{z}_t$.

Denoising Diffusion Implicit Models (DDIM)

What is the meaning of DDIM?

- **It's a generalization of DDPM!** **DDIM 是更 general 的 DDPM**

- DDPM is a special case of DDIM when

$$\sigma_t^2 = \tilde{\sigma}_t^2 = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t.$$

- In this case, DDIM operates as a Markovian process.

DDIM

Q. Check out that DDPM is a special case of DDIM when

$$\sigma_t^2 = \tilde{\sigma}_t^2 = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t.$$

Controlling Stochasticity

Let's parameterize σ_t as $\sigma_t = \eta\sqrt{\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t}$.

- $\eta = 0$: Deterministic process.
  For the same $\boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, we always obtain the same $\boldsymbol{x}_0$.

也就是說，固定 initial noise $\boldsymbol{x}_T$
不管生成幾次都會得到一樣的圖

- $\eta = 1$: Same as DDPM.

就算固定 initial noise $\boldsymbol{x}_T$
每次生成都都會得到不一樣的圖

DDIM Reverse Process

$$q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0)=\mathcal{N}\left(\sqrt{\bar{\alpha}_{t-1}}\boldsymbol{x}_0+\sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\cdot\frac{\boldsymbol{x}_t-\sqrt{\bar{\alpha}_t}\boldsymbol{x}_0}{\sqrt{1-\bar{\alpha}_t}},\sigma_t^2\mathbf{I}\right)$$

For each time step $t=T,\dots,1$, repeat:

1. Compute $\boldsymbol{x}_{0|t}=\frac{1}{\sqrt{\bar{\alpha}_t}}\left(\boldsymbol{x}_t-\sqrt{1-\bar{\alpha}_t}\hat{\boldsymbol{\varepsilon}}_\theta(\boldsymbol{x}_t,t)\right)$

From Tweedie's formula
並帶入 score 與 noise 的關係

2. Compute $\tilde{\mu}=\sqrt{\bar{\alpha}_{t-1}}\boldsymbol{x}_{0|t}+\sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\cdot\frac{\boldsymbol{x}_t-\sqrt{\bar{\alpha}_t}\boldsymbol{x}_{0|t}}{\sqrt{1-\bar{\alpha}_t}}$. 只要從 DDPM 改一行 code

3. Sample $\boldsymbol{z}_t\sim\mathcal{N}(\mathbf{0},\mathbf{I})$.

4. Compute $\boldsymbol{x}_{t-1}=\tilde{\mu}+\sigma_t\boldsymbol{z}_t$.

Controlling Stochasticity

Let's parameterize σ_t as $\sigma_t = \eta\sqrt{\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t}$.

- $\eta = 0$: Deterministic process.
  For the same $\boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, we always obtain the same $\boldsymbol{x}_0$.

- $\eta = 1$: Same as DDPM.

DDIM

Q. What is the maximum σ_t?

我們知道 $\sigma_t = 0$（最小值），DDIM 會變成 deterministic
那 σ_t 的最大值可以是什麼？

DDIM

$$q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{N}\left(\sqrt{\bar{\alpha}_{t-1}}\left(\frac{\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\varepsilon}_t}{\sqrt{\bar{\alpha}_t}}\right) + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\cdot\boldsymbol{\varepsilon}_t, \sigma_t^2\mathbf{I}\right)$$

A. $1-\bar{\alpha}_{t-1}-\sigma_t^2$ must be non-negative.

The maximum σ_t is $\sqrt{1-\bar{\alpha}_{t-1}}$.

Back to the Pros and Cons…

- (+) High quality samples
- (+) Diversity
- **(–) Very slow speed**

Accelerating Sampling Process

The DDPM/DDIM reverse process with the full sequence of time steps $t \in [1, 2, \dots, T]$:

$$p_\theta(\boldsymbol{x}_{0:T}) = p_\theta(\boldsymbol{x}_T) \prod_{t=1}^{T} p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$$

Accelerating Sampling Process

Consider a sub-sequence of the time steps:
$\tau = [\tau_1, \tau_2, \ldots, \tau_S]$.

The reverse process for this sub-sequence

$$p_\theta(\boldsymbol{x}_\tau) = p_\theta(\boldsymbol{x}_T) \prod_{t=1}^{S} p_\theta\left(\boldsymbol{x}_{\tau_i - 1} | \boldsymbol{x}_{\tau_i}\right)$$

is optimized using the same objective function as in the full sequence.

Accelerating Sampling Process

- As smaller time steps are used, the quality of the generated data can worsen.

- **However, quality degradation is mitigated when the DDIM reverse process becomes more deterministic.**

DDIM Summary

- DDPM is a special case of DDIM.
- No need to retrain the noise predictor.
- The reverse process becomes deterministic when $\sigma_t = 0$.
- Quality degradation with fewer time steps is mitigated when σ_t is small.

Evaluation Metrics for Generative Models

- Inception Score (IS) ↑
  - 評分清晰度與多樣性，但不參考真實影像
- Fréchet Inception Distance (FID) ↓
  - 比較兩坨影像（真實與生成影像）的平均特徵與特徵分散程度
- Kernel Inception Distance (KID) ↓
  - 更細緻的配對比較：隨機抽取兩小坨真實和生成圖片，使用 kernel function 計算更複雜的相似度，重複多次取平均

Evaluation Metrics for Generative Models

	IS ↑	FID ↓	KID ↓
真實影像資料	不需要	需要	需要
評估重點	生成品質+多樣性	**整體**分布相似度	**局部**特徵相似度
樣本需求	少	需要大量真實影像	**較 FID 少**
穩定度	低	中	高
計算複雜度	低	中	高

Accelerating Sampling Process

FID scores (lower is better) while varying

- η: Stochasticity
- S: The number of time steps

		CIFAR10 (32×32)					CelebA (64×64)				
	S	10	20	50	100	1000	10	20	50	100	1000
η	0.0	**13.36**	**6.84**	**4.67**	**4.16**	4.04	**17.33**	**13.73**	**9.17**	**6.53**	3.51
	0.2	14.04	7.11	4.77	4.25	4.09	17.66	14.11	9.51	6.79	3.64
	0.5	16.66	8.35	5.25	4.46	4.29	19.86	16.06	11.01	8.09	4.28
	1.0	41.07	18.36	8.01	5.78	4.73	33.12	26.03	18.48	13.93	5.98

Accelerating Sampling Process

When $\eta = 1$ (DDPM), the quality gets worse quickly as S decreases from 1000 to 10.

		CIFAR10 (32×32)					CelebA (64×64)				
	S	10	20	50	100	1000	10	20	50	100	1000
η	0.0	**13.36**	**6.84**	**4.67**	**4.16**	4.04	**17.33**	**13.73**	**9.17**	**6.53**	3.51
	0.2	14.04	7.11	4.77	4.25	4.09	17.66	14.11	9.51	6.79	3.64
	0.5	16.66	8.35	5.25	4.46	4.29	19.86	16.06	11.01	8.09	4.28
	1.0	41.07	18.36	8.01	5.78	4.73	33.12	26.03	18.48	13.93	5.98

Accelerating Sampling Process

- When $\eta = 0$ (deterministic),
- The quality does not get bad too much even when $S = 10$.
- The quality when $S = 1000$ is even better than DDPM!

		CIFAR10 (32×32)					CelebA (64×64)				
	S	10	20	50	100	1000	10	20	50	100	1000
η	0.0	**13.36**	**6.84**	**4.67**	**4.16**	4.04	**17.33**	**13.73**	**9.17**	**6.53**	3.51
	0.2	14.04	7.11	4.77	4.25	4.09	17.66	14.11	9.51	6.79	3.64
	0.5	16.66	8.35	5.25	4.46	4.29	19.86	16.06	11.01	8.09	4.28
	1.0	41.07	18.36	8.01	5.78	4.73	33.12	26.03	18.48	13.93	5.98

Accelerating Sampling Process

- When $\eta = 0$ (deterministic),
- The quality does not get bad too much even when $S = 10$.
- The quality when $S = 1000$ is even better than DDPM!

		CIFAR10 (32×32)					CelebA (64×64)				
	S	10	20	50	100	1000	10	20	50	100	1000
η	0.0	**13.36**	**6.84**	**4.67**	**4.16**	4.04	**17.33**	**13.73**	**9.17**	**6.53**	3.51
	0.2	14.04	7.11	4.77	4.25	4.09	17.66	14.11	9.51	6.79	3.64
	0.5	16.66	8.35	5.25	4.46	4.29	19.86	16.06	11.01	8.09	4.28
	1.0	41.07	18.36	8.01	5.78	4.73	33.12	26.03	18.48	13.93	5.98

Q. 為什麼 DDIM 可以讓 time step 變少時的生成品質不會掉太多？

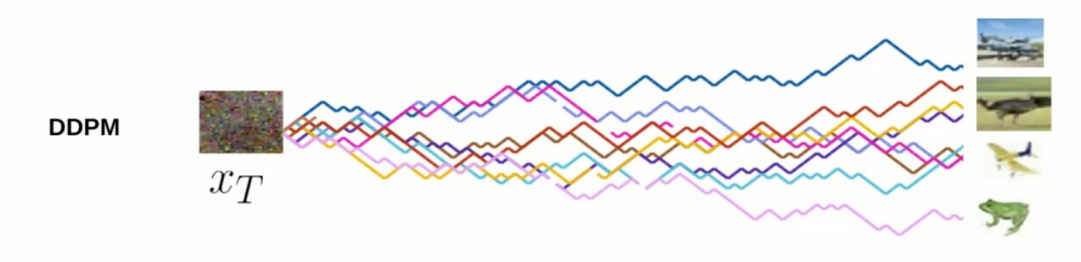
A. 關鍵差異：隨機 vs 確定性

DDPM（隨機過程）

每步都有隨機噪音，導致軌跡彎曲、迂迴，需要小步長來控制累積誤差，跳過步驟會導致分佈偏移

DDIM（確定性過程）

$\eta = 0$ 時完全確定性，軌跡是直線或平滑曲線，大步長不會引入額外隨機性，可以準確地「跳躍」



Assignment 2

- Assignment 2 is about changing DDPM to DDIM.

- It builds on your solution from Assignment 1. You can start Assignment 2 when you finish Assignment 1!