AI Applications Lecture 14

Image Generation AI 4: Goals and Scheduling of Diffusion Processes

SUZUKI, Atsushi
Jing WANG

Outline

Introduction

Recap

Last time, we learned that we can generate low-resolution **latent images** through continuous update steps using **pseudo-random numbers** and a **denoising scheduler**.

Recap

Last time, we learned that we can generate low-resolution **latent images** through continuous update steps using **pseudo-random numbers** and a **denoising scheduler**. However, what each update step **aims for** was largely based on intuition.

Recap

Last time, we learned that we can generate low-resolution **latent images** through continuous update steps using **pseudo-random numbers** and a **denoising scheduler.** However, what each update step **aims for** was largely based on intuition. In this lecture, we will explain the **meaning of the scheduler's update equations** and the **design of convergence to the target distribution** by formulating them mathematically and rigorously.

Learning Outcomes

By the end of this lecture, students should be able to:

Learning Outcomes

By the end of this lecture, students should be able to:

- Explain how to obtain **data points** using a **diffusion process** for **implicit distribution learning** in the context of **sampling** from a **target distribution.**

Learning Outcomes

By the end of this lecture, students should be able to:

- Explain how to obtain **data points** using a **diffusion process** for **implicit distribution learning** in the context of **sampling** from a **target distribution.**
- Explain how training a **noise estimator** using **data point pairs generated by adding noise** simultaneously solves **two difficulties**: (i) learning from **realistically obtainable data** and (ii) **realizing the target distribution** through a **reverse diffusion process.**

Learning Outcomes

By the end of this lecture, students should be able to:

- Explain how to obtain **data points** using a **diffusion process** for **implicit distribution learning** in the context of **sampling** from a **target distribution.**
- Explain how training a **noise estimator** using **data point pairs generated by adding noise** simultaneously solves **two difficulties**: (i) learning from **realistically obtainable data** and (ii) **realizing the target distribution** through a **reverse diffusion process.**
- Explain, through theorems and calculations, in what sense a **denoising scheduler approaches the target distribution with each update**.

Preparation: Mathematical Notations

Notations (1/2) i

- **Definition:** $(\mathrm{LHS}) \coloneqq (\mathrm{RHS})$: The left-hand side is defined by the right-hand side.

Notations (1/2) ii

- **Function:** Notation $f : \mathcal{X} \to \mathcal{Y}$, $y = f(x)$.

Notations (1/2) iii

- **Vector:**
  - Vectors are column vectors, denoted by bold italic lowercase $\boldsymbol{v}$.
  - $\boldsymbol{v} = \begin{bmatrix} v_1 & \cdots & v_n \end{bmatrix}^\top \in \mathbb{R}^n$.
  - Standard inner product $\langle \boldsymbol{u}, \boldsymbol{v} \rangle := \sum_{i=1}^{n} u_i v_i$.
- **Sequence:** $\boldsymbol{a} : [1, n]_{\mathbb{Z}} \to \mathcal{A}$ is called a sequence of length n.
- **Matrix:** Matrices are bold italic uppercase $\boldsymbol{A} \in \mathbb{R}^{m,n}$. Transpose $\boldsymbol{A}^\top \in \mathbb{R}^{n,m}$.
- **Tensor:** Tensors as multidimensional arrays are written as $\underline{\boldsymbol{A}}$.

Revisiting the Goal and Steps of the Reverse Diffusion Process

Goal: Conditional Target Distribution

The **goal** is to achieve **sampling** from a **target distribution** $\mathrm{P}_{\boldsymbol{c}}$ that depends on a text condition $\boldsymbol{c} \in \mathbb{R}^{d_{\mathrm{AllText}}}$.

Goal: Conditional Target Distribution

The **goal** is to achieve **sampling** from a **target distribution** $\mathrm{P}_{\boldsymbol{c}}$ that depends on a text condition $\boldsymbol{c} \in \mathbb{R}^{d_{\text{AllText}}}$. Here,

$$\boldsymbol{c} = \left(\boldsymbol{c}^{[j]}\right)_{j=1}^{n}$$

is an output vector sequence from **text encoders**, and although it generally consists of multiple vectors, in this lecture, we treat it as **concatenated into a single vector**.

Invalidity of the Naive Method and Motivation i

The most naive method considered is to **directly output** the **low-resolution latent images that appeared in the training data** corresponding to c, or to add **Gaussian noise** to them.

Invalidity of the Naive Method and Motivation ii

For this reason, we employ the **reverse diffusion process**, which uses the **continuity** and **nonlinearity** of **neural networks** to achieve sampling from a **non-Gaussian** distribution by **push-forward** from a **simple base distribution.**

In general, the **composition of functions**, such as neural networks, only provides a **point-to-point correspondence between input and output**, not a **distribution** directly.

Discrete-Time Reverse Diffusion and Stability i

We take a discrete time sequence $T = t_0 > t_1 > \cdots > t_K = 0$.

Discrete-Time Reverse Diffusion and Stability ii

For **inference stability**, it is designed such that at each step

$$\|\boldsymbol{x}_{t_{k+1}} - \boldsymbol{x}_{t_k}\|_2 \text{ is sufficiently small.} \tag{4}$$

If we write the distribution of $\boldsymbol{x}_{t_k}$ as P_{t_k}, the ideal is

$$\mathrm{P}_{t_0} = \text{StdNormal} \;\Rightarrow\; \mathrm{P}_{t_1} \Rightarrow \cdots \Rightarrow \mathrm{P}_{t_K} \approx \mathrm{P}_{\boldsymbol{c}}. \tag{5}$$

The Non-Triviality of Learning and Forward Noising

What is Observable and What is Missing

What we can actually obtain is only the **original data** x, which can be regarded as generated from $\mathrm{P}_{t_K} = \mathrm{P}_{\boldsymbol{c}}$.

What is Observable and What is Missing

What we can actually obtain is only the **original data** x, which can be regarded as generated from $\mathrm{P}_{t_K} = \mathrm{P}_{\boldsymbol{c}}$. The corresponding $\boldsymbol{x}_{t_{K-1}}, \ldots, \boldsymbol{x}_{t_0}$ are **not uniquely given**.

What is Observable and What is Missing

What we can actually obtain is only the **original data** x, which can be regarded as generated from $\mathrm{P}_{t_K} = \mathrm{P}_{\boldsymbol{c}}$. The corresponding $\boldsymbol{x}_{t_{K-1}}, \ldots, \boldsymbol{x}_{t_0}$ are **not uniquely given.** Therefore, it is necessary to **artificially** construct $\boldsymbol{x}_{t_{K-1}}, \ldots, \boldsymbol{x}_{t_0}$. The distributions these follow correspond to $\mathrm{P}_{t_{K-1}}, \ldots, \mathrm{P}_{t_0}$.

Two Simultaneous Challenges

Surveying the situation, we need to satisfy the following **two challenges simultaneously**:

Two Simultaneous Challenges

Surveying the situation, we need to satisfy the following **two challenges simultaneously**:

- Making the final distribution P_{t_0} match (or sufficiently approximate) the **target distribution $\mathrm{P}_{\boldsymbol{c}}$.**

Two Simultaneous Challenges

Surveying the situation, we need to satisfy the following **two challenges simultaneously**:

- Making the final distribution P_{t_0} match (or sufficiently approximate) the **target distribution** $\mathrm{P}_{\boldsymbol{c}}$.
- Being **learnable** through **realistic operations** (possible with available computational resources and data).

Available Probabilistic Tools and Construction of Artificial Distribution Sequence

Constraints on High-Dimensional Distributions i

The only practical distributions that are **directly realizable** in high dimensions and defined by a **small number of parameters** are the **isotropic Gaussian distribution**, the **Cauchy distribution**, and the **uniform distribution within a hypersphere.**

Constructing Random Variables by Linear Combination

Using a sample $\boldsymbol{x}$ from the original data distribution $\mathrm{P}_{\boldsymbol{c}}$ and an independent $\boldsymbol{\epsilon} \sim \mathsf{StdNormal}_d$, we construct

$$\zeta_{\lambda_{\mathsf{signal}}, \lambda_{\mathsf{noise}}} := \lambda_{\mathsf{signal}} \boldsymbol{x} + \lambda_{\mathsf{noise}} \boldsymbol{\epsilon}. \tag{6}$$

From a computational cost perspective, if we limit the use to one random vector per data point, the attainable random variables are effectively limited to the form above.

Constructing Random Variables by Linear Combination

Using a sample x from the original data distribution $\mathrm{P}_{\boldsymbol{c}}$ and an independent $\boldsymbol{\epsilon} \sim \mathsf{StdNormal}_d$, we construct

$$\zeta_{\lambda_{\mathsf{signal}}, \lambda_{\mathsf{noise}}} := \lambda_{\mathsf{signal}} \boldsymbol{x} + \lambda_{\mathsf{noise}} \boldsymbol{\epsilon}. \tag{6}$$

From a computational cost perspective, if we limit the use to one random vector per data point, the attainable random variables are effectively limited to the form above. The distribution it follows is written as

$$p_{\zeta_{\lambda_{\mathsf{signal}}, \lambda_{\mathsf{noise}}}}. \tag{7}$$

Designing a Smooth Distribution Sequence q_k

To transition **smoothly** from q_0 to q_K, so that the mean and variance move smoothly, we define

$$0 = \overline{\alpha}_0 < \overline{\alpha}_1 < \cdots < \overline{\alpha}_{K-1} < \overline{\alpha}_K = 1 \tag{8}$$

Designing a Smooth Distribution Sequence q_k

To transition **smoothly** from q_0 to q_K, so that the mean and variance move smoothly, we define

$$0 = \overline{\alpha}_0 < \overline{\alpha}_1 < \cdots < \overline{\alpha}_{K-1} < \overline{\alpha}_K = 1 \tag{8}$$

and set

$$q_k := p_{\boldsymbol{\zeta}_{\sqrt{\overline{\alpha}_k},\sqrt{1-\overline{\alpha}_k}}}, \tag{9}$$

$$\boldsymbol{\zeta}_k := \sqrt{\overline{\alpha}_k}\,\boldsymbol{x} + \sqrt{1-\overline{\alpha}_k}\,\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathsf{StdNormal}_d. \tag{10}$$

Remark

Let the mean and covariance of $\boldsymbol{x}$ be $\boldsymbol{m}$ and $\boldsymbol{V}$, respectively. Then

$$\mathbb{E}[\boldsymbol{\zeta}_k] = \sqrt{\overline{\alpha}_k}\,\boldsymbol{m}, \tag{11}$$

$$\mathrm{Cov}(\boldsymbol{\zeta}_k) = \overline{\alpha}_k\,\boldsymbol{V} + (1 - \overline{\alpha}_k)\boldsymbol{I}. \tag{12}$$

Thus, as k increases, the mean **monotonically** approaches $\boldsymbol{m}$ from 0, and the covariance **monotonically** approaches $\boldsymbol{V}$ from $\boldsymbol{I}$.

Global Schedule and Training Data Sequence i

The increasing sequence $\{\overline{\alpha}_k\}_{k=0}^{K}$ used during inference may not be known during training.

Global Schedule and Training Data Sequence ii

Thus, a **sequence of points** $\left(\zeta^{(i)}, t^{(i)}, \boldsymbol{x}^{(i)}\right)_{i=1}^{m}$ is obtained.

Using a neural network

$$\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}} : \mathbb{R}^{d_{\text{Latent}}} \times \mathbb{R}^{d_{\text{AllText}}} \times [0, T] \to \mathbb{R}^{d_{\text{Latent}}} \tag{17}$$

we minimize the following objective function:

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{m} \left\| \boldsymbol{\epsilon}^{(i)} - \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}\left(\boldsymbol{\zeta}^{(i)}, \boldsymbol{c}^{(i)}, t^{(i)}\right) \right\|_2^2 . \tag{18}$$

Here, in implementation, $\hat{\epsilon}$ is estimating the **noise**, but due to the **linear relationship**

$$\boldsymbol{x} = \frac{1}{\sqrt{\overline{a}_t}} \boldsymbol{\zeta} - \frac{\sqrt{1 - \overline{a}_t}}{\sqrt{\overline{a}_t}} \boldsymbol{\epsilon}, \tag{19}$$

if $\hat{\epsilon}$ is obtained, an estimate of x is also **immediately** obtained.

Goal and Tools: Connection to Scheduler Design

Scheduler Design Goal i

The **goal** is to construct an appropriate function (update equation) and provide a **scheduler** such that its **push-forward distribution**

$$q_0,\ q_1,\ \ldots,\ q_{K-1},\ q_K$$

gradually approaches the target distribution q_K (i.e., the ideal distribution matching $\mathrm{P}_{\boldsymbol{c}}$) at each step.

The main tool available is the **noise estimator** $\hat{\epsilon}_{\boldsymbol{\theta}}$, and the objective function to obtain it is the squared loss above.

Overall Strategy for Denoising Scheduler Construction

Two Main Strategies i

Finally, we construct the **denoising scheduler**. The **goal** is to use the trained noise estimator $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}$ to construct a random variable sequence

$$\boldsymbol{z}_0,\ \boldsymbol{z}_1,\ \boldsymbol{z}_2,\ \ldots,\ \boldsymbol{z}_{K-1},\ \boldsymbol{z}_K$$

such that the distribution of each $\boldsymbol{z}_k$ satisfies $\mathrm{Law}(\boldsymbol{z}_k) \approx q_k$.

Two Main Strategies ii

- **Markovian generative transitions**: Choose a joint density $\tilde{q}_{0:K}$ with marginals $\tilde{q}_k = q_k$ and easy conditionals $\tilde{q}_{k+1|k}$. Introduce $\boldsymbol{u}_k \sim \mathsf{StdNormal}_d$ and set

$$\boldsymbol{z}_{k+1} = \mathcal{G}_k(\boldsymbol{z}_k,\ \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}},\ \boldsymbol{c},\ t_k,\ h_k,\ \boldsymbol{u}_k)$$

  so that $\mathsf{Law}(\boldsymbol{z}_{k+1} \mid \boldsymbol{z}_k)$ matches $\tilde{q}_{k+1|k}$.

Deterministic Scheduling

Revisiting the Goal (Deterministic)

Use $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}$ to construct a deterministic recurrence

$$\boldsymbol{z}_{k+1} = \mathcal{F}_k(\boldsymbol{z}_0, \ldots, \boldsymbol{z}_k;\ \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}},\ \boldsymbol{c},\ t_k,\ h_k)$$

whose push-forward distribution approaches q_{k+1}. Here h_k is the discrete width in the log-SNR grid.

Log-SNR and Discrete Width

We define the **log-SNR** and width

$$\lambda_k := \log\left(\frac{\sqrt{\overline{\alpha}_k}}{\sqrt{1-\overline{\alpha}_k}}\right), \qquad h_k := \lambda_{k+1} - \lambda_k.$$

Definition: One-Step Method

Definition (One-Step Method Update $\texttt{Upd1}_{\hat{\epsilon},\textbf{coeff}}$)

Given coefficients $\{a_k, b_k\}_{k=0}^{K-1} \subset \mathbb{R}$,

$$\boldsymbol{z}_{k+1} \coloneqq a_k\, \boldsymbol{z}_k + b_k\, \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_k, \boldsymbol{c}, t_k).$$

Definition: One-Step Method

Definition (One-Step Method Update $\mathtt{Upd1}_{\hat{\epsilon},\textbf{coeff}}$)

Given coefficients $\{a_k, b_k\}_{k=0}^{K-1} \subset \mathbb{R}$,

$$\boldsymbol{z}_{k+1} := a_k\, \boldsymbol{z}_k + b_k\, \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_k, \boldsymbol{c}, t_k).$$

Remark

Through training,

$$\nabla_{\boldsymbol{z}} \log q_k(\boldsymbol{z}) \approx -\frac{1}{\sqrt{1-\overline{\alpha}_k}}\, \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{c}, t_k)$$

(see cited score-based references).

First-Order KL Expansion for One-Step i

We rewrite the update as

$$\boldsymbol{z}_{k+1} = \boldsymbol{z}_k + h_k \, \boldsymbol{v}_{\mathsf{upd1}}(\boldsymbol{z}_k, \lambda_k), \quad \boldsymbol{v}_{\mathsf{upd1}}(\boldsymbol{z}, \lambda_k) := \frac{a_k - 1}{h_k}\boldsymbol{z} + \frac{b_k}{h_k}\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}, \boldsymbol{c}, t_k).$$

The ideal connection q_λ satisfies the continuity equation

$$\partial_\lambda q_\lambda(\boldsymbol{z}) = -\nabla_{\boldsymbol{z}} \cdot \left(q_\lambda(\boldsymbol{z}) \, \boldsymbol{v}_\star(\boldsymbol{z}, \lambda) \right).$$

First-Order KL Expansion for One-Step ii

Definition (Updated Distribution and KL)

Let $p^{(1)}_{k+1}$ be the distribution of $\boldsymbol{z}_{k+1}$ from the one-step method, and consider $\mathrm{D}_{\mathrm{KL}}\left(p^{(1)}_{k+1} \,\|\, q_{k+1}\right)$.

Proposition (First-Order Expansion)

Under regularity and small h_k,

$$\mathrm{D}_{\mathrm{KL}}\left(p^{(1)}_{k+1} \,\|\, q_{k+1}\right) = \frac{h_k^2}{2}\, \mathbb{E}_{\boldsymbol{z}\sim q_k}\left[\left\|\boldsymbol{v}_{\textit{upd1}}(\boldsymbol{z}, \lambda_k) - \boldsymbol{v}_\star(\boldsymbol{z}, \lambda_k)\right\|_2^2\right] + \mathcal{O}(h_k^3).$$

Optimal Coefficients (DDIM/Euler)

Theorem (Optimal Coefficients)

$$a_k^\star = \sqrt{\frac{\overline{\alpha}_{k+1}}{\overline{\alpha}_k}} \;+\; \mathcal{O}(h_k^2), \quad b_k^\star = -\sqrt{\overline{\alpha}_{k+1}}\,\frac{\sqrt{1-\overline{\alpha}_k}}{\sqrt{\overline{\alpha}_k}} + \sqrt{1-\overline{\alpha}_{k+1}} \;+\; \mathcal{O}(h_k^2).$$

Optimal Coefficients (DDIM/Euler)

Theorem (Optimal Coefficients)

$$a_k^\star = \sqrt{\frac{\overline{\alpha}_{k+1}}{\overline{\alpha}_k}} \;+\; \mathcal{O}(h_k^2), \quad b_k^\star = -\sqrt{\overline{\alpha}_{k+1}}\,\frac{\sqrt{1-\overline{\alpha}_k}}{\sqrt{\overline{\alpha}_k}} + \sqrt{1-\overline{\alpha}_{k+1}} \;+\; \mathcal{O}(h_k^2).$$

Remark

Substituting yields the DDIM deterministic update

$$\boldsymbol{z}_{k+1} = \sqrt{\tfrac{\overline{\alpha}_{k+1}}{\overline{\alpha}_k}}\,\boldsymbol{z}_k + \Big(-\sqrt{\overline{\alpha}_{k+1}}\tfrac{\sqrt{1-\overline{\alpha}_k}}{\sqrt{\overline{\alpha}_k}} + \sqrt{1-\overline{\alpha}_{k+1}}\Big)\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_k, \boldsymbol{c}, t_k).$$

Example: Euler ODE Scheduler

Discretizing the ideal drift with forward Euler in λ gives

$$\boldsymbol{z}_{k+1} = \boldsymbol{z}_k + h_k \left(\alpha_k' \, \boldsymbol{z}_k - \frac{\beta_k'}{\sqrt{1 - \overline{\alpha}_k}} \, \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_k, \boldsymbol{c}, t_k) \right).$$

This matches the one-step coefficients and principal minimization condition.

Two-Step Method

Definition (Two-Step Method $\mathsf{Upd2}_{\hat{\epsilon},\mathbf{coeff}}$)

Given coefficients $\{a_k, b_k^{(0)}, b_k^{(1)}\}$,

$$\boldsymbol{z}_{k+1} = a_k\, \boldsymbol{z}_k + b_k^{(0)}\, \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_k, \boldsymbol{c}, t_k) + b_k^{(1)} \Big(\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_k, \boldsymbol{c}, t_k) - \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_{k-1}, \boldsymbol{c}, t_{k-1}) \Big).$$

Two-Step Method

Definition (Two-Step Method $\mathrm{Upd2}_{\hat{\epsilon},\textbf{coeff}}$**)**

Given coefficients $\{a_k, b_k^{(0)}, b_k^{(1)}\}$,

$$\boldsymbol{z}_{k+1} = a_k\, \boldsymbol{z}_k + b_k^{(0)}\, \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_k, \boldsymbol{c}, t_k) + b_k^{(1)} \Big(\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_k, \boldsymbol{c}, t_k) - \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_{k-1}, \boldsymbol{c}, t_{k-1}) \Big).$$

Proposition (Second-Order Accuracy)

$$\mathrm{D}_{\mathrm{KL}}\Big(p_{k+1}^{(2)} \,\|\, q_{k+1} \Big) = \frac{h_k^3}{2}\, \mathbb{E}_{q_k}\Big[\|\boldsymbol{a}_2(\boldsymbol{z})\|_2^2 \Big] + \mathcal{O}(h_k^4).$$

Optimal Two-Step Coefficients (DPM++ 2M)

Theorem (Optimal Coefficients)

$$a_k^\star = \sqrt{\frac{\overline{\alpha}_{k+1}}{\overline{\alpha}_k}} + \mathcal{O}(h_k^3), \quad b_k^{(0)\star} = -\frac{\sqrt{\overline{\alpha}_{k+1}}}{\sqrt{\overline{\alpha}_k}}\,\phi_1(h_k) + \sqrt{1-\overline{\alpha}_{k+1}}, \quad b_k^{(1)\star} = -\frac{\sqrt{\overline{\alpha}_{k+1}}}{\sqrt{\overline{\alpha}_k}}\,\phi_2(h_k)$$

with $\phi_1(h) = \frac{e^h-1}{h}$, $\phi_2(h) = \frac{e^h-1-h}{h^2}$.

Stage 1:

$$\tilde{\boldsymbol{z}}_{k+1} = \sqrt{\frac{\overline{\alpha}_{k+1}}{\overline{\alpha}_k}}\,\boldsymbol{z}_k - \frac{\sqrt{\overline{\alpha}_{k+1}}}{\sqrt{\overline{\alpha}_k}}\,\phi_1(h_k)\,\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_k, \boldsymbol{c}, t_k) + \sqrt{1-\overline{\alpha}_{k+1}}\,\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_k, \boldsymbol{c}, t_k).$$

Markovian Scheduling

Goal (Markovian) i

Choose a joint density $\tilde{q}_{0:K}$ with $\tilde{q}_k = q_k$ and tractable $\tilde{q}_{k+1|k}$, and design

$$\boldsymbol{z}_{k+1} = \mathcal{G}_k(\boldsymbol{z}_k,\ \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}},\ \boldsymbol{c},\ t_k,\ h_k,\ \boldsymbol{u}_k), \quad \boldsymbol{u}_k \sim \mathsf{StdNormal}_d,$$

such that $\mathsf{Law}(\boldsymbol{z}_{k+1} \mid \boldsymbol{z}_k)$ matches $\tilde{q}_{k+1|k}$.

Local Linear Noising and Gaussian Approximation

Proposition (Gaussian Approximation of Reverse Conditional)

Define

$$\boldsymbol{\xi}_k = \lambda_{\mathsf{signal}}\, \boldsymbol{\xi}_{k+1} + \delta\, \boldsymbol{\epsilon}_k, \qquad \boldsymbol{\epsilon}_k \sim \mathsf{StdNormal}_d.$$

For sufficiently small $\delta > 0$,

$$p(\boldsymbol{\xi}_{k+1} \mid \boldsymbol{\xi}_k) = \mathcal{N}(\boldsymbol{M}\boldsymbol{\xi}_k,\ \boldsymbol{\Sigma}) + \mathcal{O}(\delta^2).$$

Forward Discrete Diffusion Joint

Definition (Joint of Forward Discrete Diffusion)

In d dimensions, let

$$\boldsymbol{\xi}_K \sim q_K, \quad \boldsymbol{\epsilon}_k \sim \mathsf{StdNormal}_d, \quad \alpha_k := \frac{\overline{\alpha}_{k-1}}{\overline{\alpha}_k},$$

$$\boldsymbol{\xi}_{k-1} = \sqrt{\alpha_k}\,\boldsymbol{\xi}_k + \sqrt{1-\alpha_k}\,\boldsymbol{\epsilon}_k, \ (k = 1, \ldots, K),$$

defining $\tilde{q}_{0:K}$.

Marginals Match the Artificial Path

Theorem (Identity of Marginals)

Under the forward construction, for all k, $\tilde{q}_k = q_k$.

Markovian Update: Definition and Conditional KL

Definition (Markovian Update $\mathsf{MUpd}_{\hat{\epsilon},\mathbf{coeff}}$)

Given coefficients $\{A_k, B_k, C_k\} \subset \mathbb{R}$,

$$\boldsymbol{z}_{k+1} := A_k\, \boldsymbol{z}_k + B_k\, \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_k, \boldsymbol{c}, t_k) + C_k\, \boldsymbol{u}_k, \qquad \boldsymbol{u}_k \sim \mathsf{StdNormal}_d.$$

Markovian Update: Definition and Conditional KL

Definition (Markovian Update $\mathsf{MUpd}_{\hat{\epsilon},\textbf{coeff}}$)

Given coefficients $\{A_k, B_k, C_k\} \subset \mathbb{R}$,

$$\boldsymbol{z}_{k+1} := A_k\, \boldsymbol{z}_k + B_k\, \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_k, \boldsymbol{c}, t_k) + C_k\, \boldsymbol{u}_k, \qquad \boldsymbol{u}_k \sim \mathsf{StdNormal}_d.$$

Proposition (First-Order Conditional KL)

Let $\tilde{q}_{k+1|k} = \mathcal{N}(\mu_*(\boldsymbol{z}), \sigma_*^2 \boldsymbol{I})$. *Then*

$$\mathbb{E}_{q_k}\Big[\mathrm{D}_{\mathrm{KL}}\big(p^{\textit{cond}}_{k+1|k}(\cdot \mid \boldsymbol{z}) \,\|\, \tilde{q}_{k+1|k}(\cdot \mid \boldsymbol{z})\big)\Big] = \frac{1}{2}\,\mathbb{E}_{q_k}\left[\frac{\|\mu_{\textit{mupd}} - \mu_*\|_2^2}{\sigma_*^2} + \frac{(\sigma_{\textit{mupd}} - \sigma_*)^2}{\sigma_*^2}\right] + \mathcal{O}(h_k^2).$$

Optimal Markovian Coefficients (DDPM/Euler a)

Theorem (Optimal Coefficients)

$$A_k^\star = \sqrt{\frac{\overline{\alpha}_{k+1}}{\overline{\alpha}_k}} + \mathcal{O}(h_k^2), \quad B_k^\star = -\sqrt{\overline{\alpha}_{k+1}}\frac{\sqrt{1-\overline{\alpha}_k}}{\sqrt{\overline{\alpha}_k}} + \mathcal{O}(h_k^2), \quad C_k^\star = \sqrt{1 - \frac{\overline{\alpha}_{k+1}}{\overline{\alpha}_k}} + \mathcal{O}(h_k^2).$$

Theorem (Optimal Coefficients)

$$A_k^\star = \sqrt{\frac{\overline{\alpha}_{k+1}}{\overline{\alpha}_k}} + \mathcal{O}(h_k^2), \quad B_k^\star = -\sqrt{\overline{\alpha}_{k+1}}\,\frac{\sqrt{1-\overline{\alpha}_k}}{\sqrt{\overline{\alpha}_k}} + \mathcal{O}(h_k^2), \quad C_k^\star = \sqrt{1 - \frac{\overline{\alpha}_{k+1}}{\overline{\alpha}_k}} + \mathcal{O}(h_k^2).$$

Example (DDPM / Euler a)

$$\boldsymbol{z}_{k+1} = \sqrt{\frac{\overline{\alpha}_{k+1}}{\overline{\alpha}_k}}\,\boldsymbol{z}_k - \sqrt{\overline{\alpha}_{k+1}}\,\frac{\sqrt{1-\overline{\alpha}_k}}{\sqrt{\overline{\alpha}_k}}\,\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_k, \boldsymbol{c}, t_k) + \sqrt{1 - \frac{\overline{\alpha}_{k+1}}{\overline{\alpha}_k}}\,\boldsymbol{u}_k.$$

Summary and Next Time

Summary (tied to Learning Outcomes)

- **Data Acquisition for Implicit Distribution Learning**: We constructed an **artificial distribution sequence** $q_0, \ldots, q_K$ and corresponding data through forward noising.

Summary (tied to Learning Outcomes)

- **Data Acquisition for Implicit Distribution Learning**: We constructed an **artificial distribution sequence** $q_0, \ldots, q_K$ and corresponding data through forward noising.
- **Role of Noise Estimator (Reconciliation)**: Learning via the objective function achieved the **reconciliation** of two difficult conditions: being learnable from real data and realizing the target distribution via reverse diffusion.

Summary (tied to Learning Outcomes)

- **Data Acquisition for Implicit Distribution Learning**: We constructed an **artificial distribution sequence** $q_0, \ldots, q_K$ and corresponding data through forward noising.
- **Role of Noise Estimator (Reconciliation)**: Learning via the objective function achieved the **reconciliation** of two difficult conditions: being learnable from real data and realizing the target distribution via reverse diffusion.
- **Proximity of Scheduler**: We showed that by determining the coefficients of deterministic (DDIM/Euler) and stochastic (DDPM/Euler a) updates through moment matching, we **approach** q_{k+1} at each update.

Next Time

Next time, we will discuss **convolutional neural networks** used in image generation AI from an implementation and design perspective (centering on U-Net [1]).

References i

[1] Olaf Ronneberger, Philipp Fischer, and Thomas Brox.
U-net: Convolutional networks for biomedical image segmentation.
In MICCAI, 2015.