AI Applications Lecture 14
Image Generation AI 4: Goals and Scheduling of Diffusion Processes

SUZUKI, Atsushi
Jing WANG

Contents

1 Introduction

1.1 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.

1.2 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**.

2 Preparation: Mathematical Notations

- **Definition:**
  - $(\mathrm{LHS}) \coloneqq (\mathrm{RHS})$: The left-hand side is defined by the right-hand side.
- **Set:**
  - Sets are often denoted by uppercase calligraphic letters. E.g.: $\mathcal{A}$.
  - $x \in \mathcal{A}$: The element x belongs to the set $\mathcal{A}$.
  - $\{\}$: The empty set.
  - $\{a, b, c\}$: The set consisting of elements a, b, c (extensional notation).
  - $\{x \in \mathcal{A} \mid P(x)\}$: The set of elements for which the proposition $P(x)$ is true (intensional notation).
  - $|\mathcal{A}|$: The number of elements in set $\mathcal{A}$ (finite sets are assumed in this lecture).
  - $\mathbb{R}, \mathbb{R}_{>0}, \mathbb{R}_{\geq 0}, \mathbb{Z}, \mathbb{Z}_{>0}, \mathbb{Z}_{\geq 0}$ have their standard meanings.

- $[1, k]_{\mathbb{Z}}$: Integer interval for $k \in \mathbb{Z}_{>0} \cup \{+\infty\}$.

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

- **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 $\langle \boldsymbol{u}, \boldsymbol{v} \rangle \coloneqq \sum_{i=1}^{n} u_i v_i$.

- **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 $\underline{\boldsymbol{A}}$.

3 Revisiting the Goal and Steps of the Reverse Diffusion Process

3.1 Goal: Sampling from a Conditional Target Distribution

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

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

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**.

3.2 Invalidity of the Naive Method and Motivation

The most naive method considered is to **directly output** the **low-resolution latent images that appeared in the training data** corresponding to $\boldsymbol{c}$, or to add **Gaussian noise** to them. However, this is **practically meaningless**. The $\boldsymbol{c}$ used in inference is unlikely to **reappear** from training, making the direct output of training data not useful, and minor modifications with Gaussian noise tend to be **unnatural as natural images**. What we want is a **distribution** that is **continuous** with respect to $\boldsymbol{c}$ (allowing interpolation to unseen $\boldsymbol{c}$) and corresponds to **natural images**. 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**.

3.3 Review of the Role of Each Step

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. Therefore, we achieve **sampling** by providing **(pseudo-)random numbers** to the **input side** to obtain a **push-forward distribution**. In other words, it is a **trick** to "sample from a distribution using an input-output relationship."

3.4 Discrete-Time Reverse Diffusion and Stability

We take a discrete time sequence $T = t_0 > t_1 > \cdots > t_K = 0$. In a one-step reverse diffusion process, we use a function

$$g_{\boldsymbol{\theta}} : \mathbb{R}^{d_{\mathrm{Latent}}} \times \mathbb{R}^{d_{\mathrm{Latent}}} \times \mathbb{R}^{d_{\mathrm{AllText}}} \times \mathbb{R} \times \mathbb{R} \longrightarrow \mathbb{R}^{d_{\mathrm{Latent}}} \tag{2}$$

(containing a neural network internally), start from a standard multivariate normal

$$\boldsymbol{x}_{t_0} \sim \mathrm{StdNormal}_{d_{\mathrm{Latent}}} \tag{3}$$

and update as

$$\boldsymbol{x}_{t_{k+1}} = g_{\boldsymbol{\theta}}\Big(\boldsymbol{x}_{t_k},\, \boldsymbol{\epsilon}^k,\, \boldsymbol{c},\, t_k,\, t_k - t_{k+1}\Big), \qquad k = 0, 1, \ldots, K-1 \tag{4}$$

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

$$\|\boldsymbol{x}_{t_{k+1}} - \boldsymbol{x}_{t_k}\|_2 \text{ is sufficiently small.} \tag{5}$$

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

$$\mathrm{P}_{t_0} = \mathrm{StdNormal} \;\Rightarrow\; \mathrm{P}_{t_1} \Rightarrow \cdots \Rightarrow \mathrm{P}_{t_K} \approx \mathrm{P}_{\boldsymbol{c}}. \tag{6}$$

4 The Non-Triviality of Learning and Forward Noising

4.1 What is Observable and What is Missing

What we can actually obtain is only the **original data** $\boldsymbol{x}$, 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 $\mathrm{P}_{t_{K-1}}, \ldots, \mathrm{P}_{t_0}$.

4.2 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).

5 Available Probabilistic Tools and Construction of Artificial Distribution Sequence

5.1 Constraints on High-Dimensional Distributions

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**. However, the Cauchy distribution **lacks an expected value** and is difficult to handle, and the uniform distribution within a hypersphere lacks a **reproductive property with respect to sums** and is also difficult to handle. Therefore, the **isotropic Gaussian distribution** is **effectively the only tool**. An isotropic Gaussian distribution is determined by a **mean vector** and a **scalar standard deviation**.

5.2 Constructing Random Variables by Linear Combination

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

$$\boldsymbol{\zeta}_{\lambda_{\text{signal}}, \lambda_{\text{noise}}} \coloneqq \lambda_{\text{signal}} \boldsymbol{x} + \lambda_{\text{noise}} \boldsymbol{\epsilon}. \tag{7}$$

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 (7). The distribution it follows is written as

$$p_{\boldsymbol{\zeta}_{\lambda_{\text{signal}}, \lambda_{\text{noise}}}}. \tag{8}$$

5.3 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

$$0 = \overline{\alpha}_0 < \overline{\alpha}_1 < \cdots < \overline{\alpha}_{K-1} < \overline{\alpha}_K = 1 \tag{9}$$

and set

$$q_k \coloneqq p_{\zeta_{\sqrt{\overline{\alpha}_k}, \sqrt{1-\overline{\alpha}_k}}}, \tag{10}$$

$$\boldsymbol{\zeta}_k \coloneqq \sqrt{\overline{\alpha}_k}\,\boldsymbol{x} + \sqrt{1-\overline{\alpha}_k}\,\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathrm{StdNormal}_d. \tag{11}$$

Remark 5.1. Let the mean and covariance of $\boldsymbol{x}$ be $\boldsymbol{m}$ and $\boldsymbol{V}$, respectively. Then

$$\mathbb{E}[\boldsymbol{\zeta}_k] = \sqrt{\overline{\alpha}_k}\,\boldsymbol{m}, \tag{12}$$

$$\mathrm{Cov}(\boldsymbol{\zeta}_k) = \overline{\alpha}_k\,\boldsymbol{V} + (1-\overline{\alpha}_k)\boldsymbol{I}. \tag{13}$$

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

5.4 Global Schedule and Training Data Sequence

The increasing sequence $\{\overline{\alpha}_k\}_{k=0}^{K}$ used during inference may not be known during training. Therefore, to cover as wide a range as possible, we choose a sufficiently large $T \in \mathbb{Z}_{>0}$ and fix a

$$1 = \overline{a}_0 > \overline{a}_1 > \cdots > \overline{a}_T = 0 \tag{14}$$

decreasing sequence (adopting the decreasing direction by convention; in principle, increasing is also fine). If $\overline{a}_t \approx \overline{\alpha}_k$, then t roughly corresponds to k. Furthermore,

$$\boldsymbol{x}^{(i)} \text{ is assumed to be independently generated from } q_K = \mathrm{P}_{\boldsymbol{c}^{(i)}}, \tag{15}$$

$$t^{(i)} \sim \mathrm{Unif}\{0, 1, \ldots, T-1\}, \quad \boldsymbol{\epsilon}^{(i)} \sim \mathrm{StdNormal}_d, \tag{16}$$

$$\boldsymbol{\zeta}^{(i)} = \sqrt{\overline{a}_{t^{(i)}}}\,\boldsymbol{x}^{(i)} + \sqrt{1-\overline{a}_{t^{(i)}}}\,\boldsymbol{\epsilon}^{(i)}. \tag{17}$$

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

5.5 Noise Estimator Learning Objective and x Reconstruction

Using a neural network

$$\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}} : \mathbb{R}^{d_{\mathrm{Latent}}} \times \mathbb{R}^{d_{\mathrm{AllText}}} \times [0, T] \to \mathbb{R}^{d_{\mathrm{Latent}}} \tag{18}$$

we minimize the following objective function:

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{m} \left\| \boldsymbol{\epsilon}^{(i)} - \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}\left(\boldsymbol{\zeta}^{(i)}, \boldsymbol{c}^{(i)}, t^{(i)}\right) \right\|_2^2. \tag{19}$$

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

$$\boldsymbol{x} = \frac{1}{\sqrt{\overline{a}_t}}\zeta - \frac{\sqrt{1-\overline{a}_t}}{\sqrt{\overline{a}_t}}\boldsymbol{\epsilon}, \tag{20}$$

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

Remark 5.2. When t is large (strong noise), $\zeta \approx \epsilon$, and information about $\boldsymbol{x}$ is scarce. In particular, at $t = T$, $\zeta = \epsilon$. Therefore, naively generating using $\hat{\epsilon}_{\theta}(\epsilon, \boldsymbol{c}, T)$ is not appropriate.

6 Goal and Tools: Connection to Scheduler Design

6.1 Scheduler Design Goal

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

$$q_0,\ q_1,\ \ldots,\ q_{K-1},\ q_K \tag{21}$$

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

$$\zeta_{\sqrt{\overline{\alpha}_k},\sqrt{1-\overline{\alpha}_k}} = \sqrt{\overline{\alpha}_k}\boldsymbol{x} + \sqrt{1-\overline{\alpha}_k}\boldsymbol{\epsilon} \tag{22}$$

follows.

6.2 Available Tool

The main tool available is the **noise estimator** $\hat{\epsilon}_{\theta}$ ((18)), and the objective function to obtain it is (19). This is expected to **indirectly** contain information about $q_0, \ldots, q_K$. Therefore, the challenge is how to construct **sampling** from (21) using this $\hat{\epsilon}_{\theta}$.

Remark 6.1. Note that when t is large (k is small), $\hat{\epsilon}_{\theta}$ is **almost useless**. In particular, at $t = T$, $\zeta = \epsilon$ and contains no $\boldsymbol{x}$ information, so one should not naively generate using $\hat{\epsilon}_{\theta}(\epsilon, \boldsymbol{c}, T)$.

7 Overall Strategy for Denoising Scheduler Construction

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

$$z_0,\ z_1,\ z_2,\ \ldots,\ z_{K-1},\ z_K \tag{23}$$

such that the distribution of each z_k satisfies

$$\mathrm{Law}(z_k) \approx q_k \qquad (k = 0, 1, \ldots, K) \tag{24}$$

(where q_k is the artificial distribution sequence defined in (10)–(11)). To achieve this goal, we consider the following two **main strategies**:

- **Transformation by constructing deterministic functions** (**deterministic push-forward**): Construct a **deterministic** recurrence relation such as

$$z_{k+1} = \mathcal{F}_k(z_0, z_1, \ldots, z_k;\ \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}},\ \boldsymbol{c},\ t_k,\ h_k) \tag{25}$$

  and design the coefficients and step sizes so that the distribution of z_{k+1} **approaches** q_{k+1}.

  Remark 7.1. In this case, note that the origin of stochasticity comes only from the initial value $z_0 \sim \mathrm{StdNormal}_d$, and the **stochastic behavior of** $\{z_k\}$ **depends only on the behavior of** z_0.

- **Generation by constructing Markovian conditional probability densities** (**Markovian generative transitions**): Choose a **joint probability density**

$$\tilde{q}_{0:K}(\boldsymbol{\zeta}_0, \ldots, \boldsymbol{\zeta}_K) \tag{26}$$

  such that each marginal $\tilde{q}_k$ **matches** q_k, and the **conditional** $\tilde{q}_{k+1|k}$ is **easy to construct** (specifically, using a **discrete-time diffusion process**). Then, introduce a standard multivariate normal $\boldsymbol{u}_k \sim \mathrm{StdNormal}_d$,

$$z_{k+1} = \mathcal{G}_k(z_k,\ \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}},\ \boldsymbol{c},\ t_k,\ h_k,\ \boldsymbol{u}_k) \tag{27}$$

  and design the coefficients and variance terms so that the probability distribution of z_{k+1} (given z_k) **matches** the **designed** $\tilde{q}_{k+1|k}$.

These two strategies are complementary. The former provides **deterministic scheduling** based on moment matching and local error analysis from the **push-forward** perspective, while the latter systematically derives **stochastic updates** (such as ancestral steps) that match the **conditional distribution**. In the following, we first provide the concretization (coefficient design) of the deterministic update, and then proceed to the design of the Markovian update.

8 Deterministic Scheduling

8.1 Revisiting the Goal

The **goal** of this section is to use the trained noise estimator $\hat{\epsilon}_\theta$ to construct a **deterministic** recurrence relation

$$z_{k+1} = \mathcal{F}_k(z_0, \dots, z_k;\ \hat{\epsilon}_\theta,\ c,\ t_k,\ h_k) \tag{28}$$

and design the coefficients and discrete width h_k such that its push-forward distribution $\mathcal{L}(z_{k+1})$ **approaches** q_{k+1}. Here q_k is the artificial distribution sequence defined in (10)–(11).

8.2 Abstract Definition of One-Step Method

We define the **log-SNR** as

$$\lambda_k := \log\left(\frac{\sqrt{\bar{\alpha}_k}}{\sqrt{1-\bar{\alpha}_k}}\right), \qquad h_k := \lambda_{k+1} - \lambda_k \tag{29}$$

and call h_k the **discrete width**.

Definition 8.1 (One-Step Method Update $\mathrm{Upd1}_{\hat{\epsilon},\mathrm{coeff}}$)**.** Given an arbitrary sequence of coefficients $\{a_k, b_k\}_{k=0}^{K-1} \subset \mathbb{R}$. The **one-step method** is a deterministic recurrence relation determined by a linear combination of z_k and the noise estimate $\hat{\epsilon}_\theta(z_k, c, t_k)$ as

$$z_{k+1} := a_k\, z_k + b_k\, \hat{\epsilon}_\theta(z_k, c, t_k). \tag{30}$$

Remark 8.1. Through training, $\hat{\epsilon}_\theta(z, c, t_k)$ is proportional to the **score** of q_k:

$$\nabla_z \log q_k(z) \approx -\frac{1}{\sqrt{1-\bar{\alpha}_k}}\, \hat{\epsilon}_\theta(z, c, t_k) \tag{31}$$

(see [5]).

8.3 First-Order Expansion of KL Divergence for One-Step Method

First, let's set up the notation. We write (30) as

$$z_{k+1} = z_k + h_k\, v_{\mathrm{upd1}}(z_k, \lambda_k), \quad v_{\mathrm{upd1}}(z, \lambda_k) := \frac{a_k - 1}{h_k} z + \frac{b_k}{h_k} \hat{\epsilon}_\theta(z, c, t_k). \tag{32}$$

The **ideal** continuous-time connection q_λ between q_k and q_{k+1} follows the **continuity equation**:

$$\partial_\lambda q_\lambda(z) = -\nabla_z \cdot (q_\lambda(z)\, v_\star(z, \lambda)), \tag{33}$$

where $v_\star(z, \lambda) = A(\lambda) z + B(\lambda) \nabla_z \log q_\lambda(z)$ is the deterministic drift consistent with q_λ ([2, 5]).

Definition 8.2 (Updated Distribution and KL Divergence)**.** We denote the distribution of z_{k+1} resulting from the update in Definition 8.1 as $p_{k+1}^{(1)}$, and its divergence as

$$\mathrm{D_{KL}}\Big(p_{k+1}^{(1)} \,\|\, q_{k+1}\Big). \tag{34}$$

The following proposition rigorously expands (34) in the limit of small h_k.

Proposition 8.1 (First-Order Expansion of KL Divergence for One-Step Method)**.** Assumptions:

- q_λ is C^2 with respect to λ and C^3 with respect to z, and $\|\nabla_z^j \log q_\lambda(z)\|$ is $L^1(\mathbb{R}^d)$ integrable and decays sufficiently for $j \le 3$.
- $\hat{\boldsymbol{\epsilon}}_\theta(z, \boldsymbol{c}, t)$ is C^2 with respect to z and has at most polynomial growth.
- h_k is sufficiently small.

Then

$$\mathrm{D_{KL}}\Big(p_{k+1}^{(1)} \,\|\, q_{k+1}\Big) = \frac{h_k^2}{2}\, \mathbb{E}_{z\sim q_k}\Big[\big\|\boldsymbol{v}_{\mathsf{upd1}}(z,\lambda_k) - \boldsymbol{v}_\star(z,\lambda_k)\big\|_2^2\Big] + O(h_k^3). \tag{35}$$

Proof. **(1) First-order expansion of q_{k+1}:** Discretizing (33) with forward Euler in λ,

$$q_{k+1}(z) = q_k(z) - h_k\, \nabla_z \cdot (q_k(z)\, \boldsymbol{v}_\star(z,\lambda_k)) + O(h_k^2). \tag{36}$$

(2) First-order expansion of the push-forward distribution $p_{k+1}^{(1)}$: Consider the transformation $\Phi(z) = z + h_k\, \boldsymbol{v}_{\mathsf{upd1}}(z,\lambda_k)$. For small h_k, Φ is a diffeomorphism, and its inverse map Φ^{-1} can be Taylor expanded as

$$\Phi^{-1}(\boldsymbol{y}) = \boldsymbol{y} - h_k\, \boldsymbol{v}_{\mathsf{upd1}}(\boldsymbol{y},\lambda_k) + O(h_k^2). \tag{37}$$

The Jacobian is

$$\begin{aligned}\det\Big(\nabla\Phi^{-1}(\boldsymbol{y})\Big) &= \det\big(\boldsymbol{I} - h_k\, \nabla_{\boldsymbol{y}} \boldsymbol{v}_{\mathsf{upd1}}(\boldsymbol{y},\lambda_k)\big) + O(h_k^2) \\ &= 1 - h_k\, \mathrm{tr}\big(\nabla_{\boldsymbol{y}} \boldsymbol{v}_{\mathsf{upd1}}(\boldsymbol{y},\lambda_k)\big) + O(h_k^2).\end{aligned} \tag{38}$$

Therefore

$$\begin{aligned}p_{k+1}^{(1)}(\boldsymbol{y}) &= q_k\Big(\Phi^{-1}(\boldsymbol{y})\Big)\, \det\Big(\nabla\Phi^{-1}(\boldsymbol{y})\Big) \\ &= \Big(q_k(\boldsymbol{y}) - h_k\, \nabla_{\boldsymbol{y}} q_k(\boldsymbol{y}) \cdot \boldsymbol{v}_{\mathsf{upd1}}(\boldsymbol{y},\lambda_k) + O(h_k^2)\Big) \\ &\quad \times \Big(1 - h_k\, \nabla_{\boldsymbol{y}} \cdot \boldsymbol{v}_{\mathsf{upd1}}(\boldsymbol{y},\lambda_k) + O(h_k^2)\Big) \\ &= q_k(\boldsymbol{y}) - h_k\, \big(\nabla_{\boldsymbol{y}} q_k \cdot \boldsymbol{v}_{\mathsf{upd1}} + q_k\, \nabla_{\boldsymbol{y}} \cdot \boldsymbol{v}_{\mathsf{upd1}}\big) + O(h_k^2) \\ &= q_k(\boldsymbol{y}) - h_k\, \nabla_{\boldsymbol{y}} \cdot \big(q_k(\boldsymbol{y})\, \boldsymbol{v}_{\mathsf{upd1}}(\boldsymbol{y},\lambda_k)\big) + O(h_k^2),\end{aligned} \tag{39}$$

where the last equality uses the product rule for derivatives $\nabla(q_k)\cdot \boldsymbol{v} + q_k\,\nabla\cdot\boldsymbol{v} = \nabla\cdot(q_k\boldsymbol{v})$.

(3) First-order expression of the difference: The difference between (36) and (39)

$$\delta(\boldsymbol{y}) \coloneqq p^{(1)}_{k+1}(\boldsymbol{y}) - q_{k+1}(\boldsymbol{y}) = -h_k\,\nabla_{\boldsymbol{y}}\cdot\big(q_k(\boldsymbol{y})\,\big[\boldsymbol{v}_{\mathsf{upd1}}(\boldsymbol{y}) - \boldsymbol{v}_\star(\boldsymbol{y})\big]\big) + O(h_k^2). \tag{40}$$

(4) Quadratic expansion of KL: For $p = q + \delta$,

$$\begin{aligned}
\mathrm{D}_{\mathrm{KL}}(p\|q) &= \int_{\mathbb{R}^d} (q+\delta)\,\log\!\left(1+\frac{\delta}{q}\right)\mathrm{d}\boldsymbol{y} \\
&= \int (q+\delta)\left(\frac{\delta}{q} - \frac{1}{2}\frac{\delta^2}{q^2}\right)\mathrm{d}\boldsymbol{y} + O\!\left(\int \frac{|\delta|^3}{q^2}\mathrm{d}\boldsymbol{y}\right) \\
&= \int \frac{\delta^2}{2q}\,\mathrm{d}\boldsymbol{y} + O(\|\delta\|^3_{L^3}).
\end{aligned} \tag{41}$$

By assumption $\delta = O(h_k)$, so $O(\|\delta\|^3_{L^3}) = O(h_k^3)$.

(5) Evaluation of the principal term by integration by parts: Substituting (40) into (41) and sequentially applying integration by parts, using the boundary term being 0 (decay assumption),

$$\begin{aligned}
\int \frac{\delta^2}{2q_{k+1}}\,\mathrm{d}\boldsymbol{y} &= \frac{h_k^2}{2}\int \frac{\big[\nabla\cdot(q_k(\boldsymbol{v}_{\mathsf{upd1}} - \boldsymbol{v}_\star))\big]^2}{q_k}\,\mathrm{d}\boldsymbol{y} + O(h_k^3) \\
&= \frac{h_k^2}{2}\int q_k\,\big\|\boldsymbol{v}_{\mathsf{upd1}} - \boldsymbol{v}_\star\big\|_2^2\,\mathrm{d}\boldsymbol{y} + O(h_k^3),
\end{aligned} \tag{42}$$

where the last equality is obtained by successively applying integration by parts identities (Fisher information type reduction) that pair gradient and divergence terms, in line with the equality condition of the Picard inequality in the $L^2(q_k)$ inner product (intermediate calculations involve terms of the form $\int \partial_i(q_k w_i)\partial_j(q_k w_j)/q_k$, which sequentially reduce to $\int q_k w_i w_i$). Substituting this into (41) yields (35). □

8.4 Optimization of One-Step Method Coefficients (DDIM/Euler)

Theorem 8.1 (Optimal Coefficients for One-Step Method by Minimizing Divergence)**.** Under the assumptions of Proposition 8.1,

$$a_k^\star = \sqrt{\frac{\overline{\alpha}_{k+1}}{\overline{\alpha}_k}} \;+\; O(h_k^2), \tag{43}$$

$$b_k^\star = -\sqrt{\overline{\alpha}_{k+1}}\,\frac{\sqrt{1-\overline{\alpha}_k}}{\sqrt{\overline{\alpha}_k}} + \sqrt{1-\overline{\alpha}_{k+1}} \;+\; O(h_k^2), \tag{44}$$

minimize the principal term of $\mathrm{D}_{\mathrm{KL}}\Big(p^{(1)}_{k+1}\,\|\,q_{k+1}\Big)$.

Proof. **(1) Form of the ideal drift:** From the structure of (33) and (11), the drift consistent

with q_k is

$$\boldsymbol{v}_\star(z, \lambda_k) = \alpha'_k\, z + \beta'_k\, \nabla_z \log q_k(z), \tag{45}$$

where α'_k, β'_k are smooth functions of $\overline{\alpha}_k$, and to first-order accuracy

$$\alpha'_k = \frac{1}{h_k}\left(\frac{\sqrt{\overline{\alpha}_{k+1}} - \sqrt{\overline{\alpha}_k}}{\sqrt{\overline{\alpha}_k}}\right) + O(h_k), \tag{46}$$

$$\beta'_k = \sqrt{\overline{\alpha}_{k+1}} - \sqrt{\overline{\alpha}_k} + O(h_k^2), \tag{47}$$

(can be derived from the first-order difference of λ).

(2) Substitution of ϵ-approximation for the score: Substituting (31) into (45),

$$\boldsymbol{v}_\star(z, \lambda_k) = \alpha'_k\, z - \frac{\beta'_k}{\sqrt{1-\overline{\alpha}_k}}\, \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(z, \boldsymbol{c}, t_k). \tag{48}$$

(3) Separation of the minimization problem: From Proposition 8.1, the minimization of the principal term is equivalent to

$$\min_{a_k, b_k}\ \mathbb{E}_{q_k}\left[\left\|\frac{a_k - 1}{h_k} z + \frac{b_k}{h_k}\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}} - \alpha'_k\, z + \frac{\beta'_k}{\sqrt{1-\overline{\alpha}_k}}\, \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}\right\|_2^2\right]. \tag{49}$$

Although z and $\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}$ are not independent in $L^2(q_k)$, from coefficient comparison (uniqueness of least squares in any vector space), the minimum is achieved by matching the coefficients of each basis direction:

$$\frac{a_k - 1}{h_k} = \alpha'_k + O(h_k), \tag{50}$$

$$\frac{b_k}{h_k} = -\frac{\beta'_k}{\sqrt{1-\overline{\alpha}_k}} + O(h_k). \tag{51}$$

(4) Substitution and simplification of (46)**–**(47)**:** From (50)

$$\begin{aligned} a_k &= 1 + h_k \alpha'_k + O(h_k^2) = 1 + \left(\frac{\sqrt{\overline{\alpha}_{k+1}} - \sqrt{\overline{\alpha}_k}}{\sqrt{\overline{\alpha}_k}}\right) + O(h_k^2) \\ &= \frac{\sqrt{\overline{\alpha}_{k+1}}}{\sqrt{\overline{\alpha}_k}} + O(h_k^2) = \sqrt{\frac{\overline{\alpha}_{k+1}}{\overline{\alpha}_k}} + O(h_k^2), \end{aligned} \tag{52}$$

From (51)

$$
\begin{aligned}
b_k &= -h_k \frac{\beta'_k}{\sqrt{1-\overline{\alpha}_k}} + O(h_k^2) = -\frac{\sqrt{\overline{\alpha}_{k+1}} - \sqrt{\overline{\alpha}_k}}{\sqrt{1-\overline{\alpha}_k}} + O(h_k^2) \\
&= -\sqrt{\overline{\alpha}_{k+1}} \frac{1}{\sqrt{1-\overline{\alpha}_k}} + \frac{\sqrt{\overline{\alpha}_k}}{\sqrt{1-\overline{\alpha}_k}} + O(h_k^2) \\
&= -\sqrt{\overline{\alpha}_{k+1}} \frac{\sqrt{1-\overline{\alpha}_k}}{\sqrt{\overline{\alpha}_k}} \cdot \frac{1}{1-\overline{\alpha}_k} + \sqrt{1-\overline{\alpha}_k} \cdot \frac{1}{1-\overline{\alpha}_k} + O(h_k^2) \\
&= -\sqrt{\overline{\alpha}_{k+1}} \frac{\sqrt{1-\overline{\alpha}_k}}{\sqrt{\overline{\alpha}_k}} + \sqrt{1-\overline{\alpha}_k}\ \frac{1}{1-\overline{\alpha}_k} \Big(1 - \overline{\alpha}_{k+1} + \overline{\alpha}_{k+1} - (1-\overline{\alpha}_k)\Big) \\
&= -\sqrt{\overline{\alpha}_{k+1}} \frac{\sqrt{1-\overline{\alpha}_k}}{\sqrt{\overline{\alpha}_k}} + \sqrt{1-\overline{\alpha}_{k+1}} + O(h_k^2),
\end{aligned} \tag{53}
$$

where the inserted term in the fourth line is an equivalent transformation to match up to the first order of h_k using the identity $1 = (1-\overline{\alpha}_{k+1}) + (\overline{\alpha}_{k+1} - (1-\overline{\alpha}_k))$. (52) and (53) give (43)–(44). □

Remark 8.2. Substituting (43), (44) back into (30) gives

$$
z_{k+1} = \sqrt{\frac{\overline{\alpha}_{k+1}}{\overline{\alpha}_k}}\, z_k + \left(-\sqrt{\overline{\alpha}_{k+1}} \frac{\sqrt{1-\overline{\alpha}_k}}{\sqrt{\overline{\alpha}_k}} + \sqrt{1-\overline{\alpha}_{k+1}}\right) \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(z_k, \boldsymbol{c}, t_k), \tag{54}
$$

which matches the deterministic update equation of **DDIM** widely used in implementation.

Example 8.1 (Euler ODE Scheduler)**.** Discretizing (48) with forward Euler in λ gives

$$
z_{k+1} = z_k + h_k \left(\alpha'_k\, z_k - \frac{\beta'_k}{\sqrt{1-\overline{\alpha}_k}}\, \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(z_k, \boldsymbol{c}, t_k)\right). \tag{55}
$$

This matches the coefficients of (32) and satisfies the principal term minimization condition of Proposition 8.1.

8.5 Abstract Definition of Two-Step Method

Definition 8.3 (Two-Step Method Update $\mathrm{Upd2}_{\hat{\boldsymbol{\epsilon}},\mathrm{coeff}}$)**.** Given arbitrary coefficients $\{a_k, b_k^{(0)}, b_k^{(1)}\} \subset \mathbb{R}$, we define

$$
z_{k+1} \coloneqq a_k\, z_k + b_k^{(0)}\, \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(z_k, \boldsymbol{c}, t_k) + b_k^{(1)} \Big(\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(z_k, \boldsymbol{c}, t_k) - \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(z_{k-1}, \boldsymbol{c}, t_{k-1})\Big). \tag{56}
$$

Proposition 8.2 (Second-Order Accuracy of KL Divergence for Two-Step Method)**.** Under similar regularity assumptions as the one-step method,

$$
\mathrm{D}_{\mathrm{KL}}\Big(p_{k+1}^{(2)} \,\|\, q_{k+1}\Big) = \frac{h_k^3}{2}\, \mathbb{E}_{q_k}\big[\|\boldsymbol{a}_2(z)\|_2^2\big] + O(h_k^4), \tag{57}
$$

holds. Here $p_{k+1}^{(2)}$ is the push-forward distribution of (56), and $\boldsymbol{a}_2(z)$ is the difference in the **second-order local coefficients** of $\boldsymbol{v}_{\text{upd2}} - \boldsymbol{v}_\star$.

Proof. **(1) Expand the two-step method into a second-order local map:** View (56) as

$$z_{k+1} = z_k + h_k\, \boldsymbol{v}_{\text{upd2}}(z_k) + \frac{h_k^2}{2}\, \boldsymbol{w}_{\text{upd2}}(z_k) + O(h_k^3) \tag{58}$$

(approximating the second-order term of the finite difference with the difference of $\hat{\boldsymbol{\epsilon}}$), and on the q_{k+1} side, expand (33) to second order (equivalent to Heun/Adams–Bashforth)

$$q_{k+1} = q_k - h_k \nabla\cdot (q_k \boldsymbol{v}_\star) + \frac{h_k^2}{2}\, \Xi(q_k, \boldsymbol{v}_\star) + O(h_k^3), \tag{59}$$

where Ξ is a linear operator for the second-order term involving the Jacobian and density gradient.

(2) Second-order Taylor of change of variables: Let $\Phi(z) = z + h_k \boldsymbol{v}_{\text{upd2}} + \frac{h_k^2}{2}\boldsymbol{w}_{\text{upd2}}$, and expand Φ^{-1} and $\det(\nabla\Phi^{-1})$ to second order. Simplifying similarly to the one-step method,

$$p_{k+1}^{(2)} = q_k - h_k \nabla\cdot (q_k \boldsymbol{v}_{\text{upd2}}) + \frac{h_k^2}{2}\Big(\Xi(q_k, \boldsymbol{v}_{\text{upd2}}) - \nabla\cdot (q_k \boldsymbol{w}_{\text{upd2}})\Big) + O(h_k^3). \tag{60}$$

(3) Difference and KL: Write the difference between (59) and (60) as $\delta = -h_k \nabla\cdot\big(q_k(\boldsymbol{v}_{\text{upd2}} - \boldsymbol{v}_\star)\big) + \frac{h_k^2}{2}\big(\Xi(q_k, \boldsymbol{v}_{\text{upd2}}) - \Xi(q_k, \boldsymbol{v}_\star) - \nabla\cdot(q_k \boldsymbol{w}_{\text{upd2}})\big) + O(h_k^3)$, and apply a quadratic expansion similar to (41). Due to the matching of coefficients up to second order, the first-order contribution vanishes, and the principal term is of order h_k^3, $\propto \int q_k \|\boldsymbol{a}_2(z)\|^2$. This gives (57). □

Theorem 8.2 (Optimal Coefficients for Two-Step Method by Minimizing Divergence (DPM++ 2M form))**.** The coefficients that minimize D_{KL} to second-order accuracy are given by

$$a_k^\star = \sqrt{\frac{\overline{\alpha}_{k+1}}{\overline{\alpha}_k}} + O(h_k^3), \tag{61}$$

$$b_k^{(0)\star} = -\frac{\sqrt{\overline{\alpha}_{k+1}}}{\sqrt{\overline{\alpha}_k}}\, \phi_1(h_k) + \sqrt{1 - \overline{\alpha}_{k+1}}\,, \qquad \phi_1(h) \coloneqq \frac{e^h - 1}{h}, \tag{62}$$

$$b_k^{(1)\star} = -\frac{\sqrt{\overline{\alpha}_{k+1}}}{\sqrt{\overline{\alpha}_k}}\, \phi_2(h_k), \qquad \phi_2(h) \coloneqq \frac{e^h - 1 - h}{h^2}, \tag{63}$$

Proof. **(1) Exponential integrator form of the probability flow ODE:** Using (48), $\partial_\lambda z = \alpha'(\lambda) z - \gamma(\lambda)\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(z, \boldsymbol{c}, t(\lambda))$ ($\gamma(\lambda) \coloneqq \beta'(\lambda)/\sqrt{1 - \overline{\alpha}(\lambda)}$). Approximating this with a two-step exponential integrator ϕ-function, the local transition is

$$z_{k+1} = e^{h_k} z_k - e^{h_k}\phi_1(h_k)\, \hat{\gamma}_k\, \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(z_k) - e^{h_k}\phi_2(h_k)\, \hat{\gamma}_k \Big(\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\tilde{z}_{k+1}) - \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(z_k)\Big) + O(h_k^3), \tag{64}$$

where $\hat{\gamma}_k \approx \frac{\sqrt{\overline{\alpha}_{k+1}}}{\sqrt{\overline{\alpha}_k}} - 1 = O(h_k)$, $e^{h_k} = \sqrt{\overline{\alpha}_{k+1}/\overline{\alpha}_k}$ (from the definition of λ).

(2) Coefficient identification: Comparing coefficients between (64) and (56), and substituting $e^{h_k} = \sqrt{\overline{\alpha}_{k+1}/\overline{\alpha}_k}$, we obtain (61)–(63). The cancellation of the second-order error satisfies the condition $\boldsymbol{a}_2(z) \equiv 0$ of Proposition 8.2. □

Example 8.2 (Explicit formula for DPM++ 2M Karras)**.** The implementation is given in the following two stages:

$$\tilde{z}_{k+1} = \sqrt{\frac{\overline{\alpha}_{k+1}}{\overline{\alpha}_k}}\, z_k - \frac{\sqrt{\overline{\alpha}_{k+1}}}{\sqrt{\overline{\alpha}_k}}\, \phi_1(h_k)\, \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(z_k, \boldsymbol{c}, t_k) + \sqrt{1-\overline{\alpha}_{k+1}}\, \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(z_k, \boldsymbol{c}, t_k), \tag{65}$$

$$\begin{aligned} z_{k+1} = \sqrt{\frac{\overline{\alpha}_{k+1}}{\overline{\alpha}_k}}\, z_k - \frac{\sqrt{\overline{\alpha}_{k+1}}}{\sqrt{\overline{\alpha}_k}} \Big(\phi_1(h_k)\, \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(z_k, \boldsymbol{c}, t_k) + \phi_2(h_k)\, [\hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\tilde{z}_{k+1}, \boldsymbol{c}, t_{k+1}) - \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(z_k, \boldsymbol{c}, t_k)] \Big) \\ + \sqrt{1-\overline{\alpha}_{k+1}}\, \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\tilde{z}_{k+1}, \boldsymbol{c}, t_{k+1}), \end{aligned} \tag{66}$$

$h_k = \lambda_{k+1} - \lambda_k$. This matches the coefficients of Theorem 8.2 and achieves a local error of $O(h_k^3)$ [2, 3].

9 Markovian Scheduling

9.1 Revisiting the Goal

The **goal** of this section is to choose a **joint probability density** $\tilde{q}_{0:K}$ such that each marginal $\tilde{q}_k$ matches q_k, and the conditional $\tilde{q}_{k+1|k}$ is easy to construct (discrete-time diffusion process), and to design

$$z_{k+1} = \mathcal{G}_k(z_k,\ \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}},\ \boldsymbol{c},\ t_k,\ h_k,\ \boldsymbol{u}_k), \qquad \boldsymbol{u}_k \sim \text{StdNormal}_d, \tag{67}$$

such that the distribution of z_{k+1} given z_k **matches** $\tilde{q}_{k+1|k}$.

9.2 Local Linear Noising and Gaussian Approximation of Reverse Conditional

Proposition 9.1 (Gaussian Approximation of Reverse Conditional in Local Linear Noising)**.** Define a sequence of random variables $\{\boldsymbol{\xi}_k\}$ by

$$\boldsymbol{\xi}_k = \lambda_{\text{signal}}\, \boldsymbol{\xi}_{k+1} + \delta\, \boldsymbol{\epsilon}_k, \qquad \boldsymbol{\epsilon}_k \sim \text{StdNormal}_d,\ \text{independent}. \tag{68}$$

When $\delta > 0$ is sufficiently small, even if $\boldsymbol{\xi}_k$ is non-Gaussian, the reverse conditional distribution $p(\boldsymbol{\xi}_{k+1} \mid \boldsymbol{\xi}_k)$ is

$$p(\boldsymbol{\xi}_{k+1} \mid \boldsymbol{\xi}_k) = \mathcal{N}\big(\boldsymbol{M}\boldsymbol{\xi}_k,\ \boldsymbol{\Sigma}\big) + O(\delta^2), \tag{69}$$

(where $\boldsymbol{M}, \boldsymbol{\Sigma}$ are first-order functions of $\lambda_{\text{signal}}, \delta$).

Proof. **(1) Application of Bayes' rule:** $p(\boldsymbol{\xi}_{k+1} \mid \boldsymbol{\xi}_k) \propto p(\boldsymbol{\xi}_k \mid \boldsymbol{\xi}_{k+1})\, p(\boldsymbol{\xi}_{k+1})$. Here $p(\boldsymbol{\xi}_k \mid \boldsymbol{\xi}_{k+1}) = \mathcal{N}(\lambda_{\text{signal}}\boldsymbol{\xi}_{k+1}, \delta^2 \boldsymbol{I})$ is exactly Gaussian.

(2) Laplace approximation: The log-posterior $\ell(\boldsymbol{u}) \coloneqq \log p(\boldsymbol{u} \mid \boldsymbol{\xi}_k)$ with $\boldsymbol{u} = \boldsymbol{\xi}_{k+1}$ is

$$\ell(\boldsymbol{u}) = -\frac{1}{2\delta^2} \left\| \boldsymbol{\xi}_k - \lambda_{\text{signal}} \boldsymbol{u} \right\|_2^2 + \log p(\boldsymbol{u}) + \text{const.} \tag{70}$$

The gradient and Hessian are

$$\nabla_{\boldsymbol{u}} \ell(\boldsymbol{u}) = \frac{\lambda_{\text{signal}}}{\delta^2} \left(\boldsymbol{\xi}_k - \lambda_{\text{signal}} \boldsymbol{u} \right) + \nabla_{\boldsymbol{u}} \log p(\boldsymbol{u}), \tag{71}$$

$$\nabla_{\boldsymbol{u}}^2 \ell(\boldsymbol{u}) = -\frac{\lambda_{\text{signal}}^2}{\delta^2} \boldsymbol{I} + \nabla_{\boldsymbol{u}}^2 \log p(\boldsymbol{u}). \tag{72}$$

As $\delta \to 0$, the first term dominates.

(3) First-order approximation of the mode: From $\nabla \ell(\boldsymbol{u}^\star) = 0$, $\frac{\lambda_{\text{signal}}}{\delta^2}(\boldsymbol{\xi}_k - \lambda_{\text{signal}} \boldsymbol{u}^\star) = -\nabla \log p(\boldsymbol{u}^\star)$. The right side is $O(1)$, the left side is $O(\delta^{-2})$, so $\boldsymbol{u}^\star = \lambda_{\text{signal}}^{-1} \boldsymbol{\xi}_k + O(\delta^2)$.

(4) Quadratic approximation and covariance: The value of (72) at the mode is $-\nabla^2 \ell(\boldsymbol{u}^\star) = \frac{\lambda_{\text{signal}}^2}{\delta^2} \boldsymbol{I} - \nabla^2 \log p(\boldsymbol{u}^\star)$. The principal term is $\frac{\lambda_{\text{signal}}^2}{\delta^2} \boldsymbol{I}$, so $\boldsymbol{\Sigma} = \left(\frac{\lambda_{\text{signal}}^2}{\delta^2} \boldsymbol{I} \right)^{-1} + O(\delta^2) = \frac{\delta^2}{\lambda_{\text{signal}}^2} \boldsymbol{I} + O(\delta^4)$. The mean is $\boldsymbol{M}\boldsymbol{\xi}_k = \boldsymbol{u}^\star + O(\delta^2) = \lambda_{\text{signal}}^{-1} \boldsymbol{\xi}_k + O(\delta^2)$. □

9.3 $\tilde{q}$ as a Discrete-Time Diffusion Process

Definition 9.1 (Joint Distribution of Forward Discrete Diffusion)**.** In d dimensions, let

$$\boldsymbol{\xi}_K \sim q_K, \qquad \boldsymbol{\epsilon}_k \sim \text{StdNormal}_d \text{ independent}, \tag{73}$$

$$\alpha_k \coloneqq \frac{\overline{\alpha}_{k-1}}{\overline{\alpha}_k}, \qquad \boldsymbol{\xi}_{k-1} = \sqrt{\alpha_k}\, \boldsymbol{\xi}_k + \sqrt{1 - \alpha_k}\, \boldsymbol{\epsilon}_k, \quad (k = 1, \ldots, K), \tag{74}$$

define the joint distribution $\tilde{q}_{0:K}$.

Theorem 9.1 (Identity $\tilde{q}_k = q_k$)**.** Under Definition 9.1, for all k,

$$\tilde{q}_k = q_k \tag{75}$$

holds.

Proof. **(1)** $k = K$**:** By definition $\tilde{q}_K = q_K$.

(2) Induction: Assume $\tilde{q}_k = q_k$ and show $\tilde{q}_{k-1} = q_{k-1}$. From (74) $\sqrt{\overline{\alpha}_{k-1}}\, \boldsymbol{\xi}_{k-1} = \sqrt{\overline{\alpha}_k}\, \boldsymbol{\xi}_k + \sqrt{\overline{\alpha}_{k-1} - \overline{\alpha}_k}\, \boldsymbol{\epsilon}_k$. By assumption $\boldsymbol{\xi}_k \overset{d}{=} \sqrt{\overline{\alpha}_k} \boldsymbol{x} + \sqrt{1 - \overline{\alpha}_k} \boldsymbol{\epsilon}$. By addition of independent normals $\sqrt{\overline{\alpha}_k}\, \boldsymbol{\xi}_k + \sqrt{\overline{\alpha}_{k-1} - \overline{\alpha}_k}\, \boldsymbol{\epsilon}_k \overset{d}{=} \sqrt{\overline{\alpha}_k}\, \boldsymbol{x} + \sqrt{1 - \overline{\alpha}_{k-1}}\, \boldsymbol{\epsilon}'$ (additivity of variance and independence). Thus $\boldsymbol{\xi}_{k-1} \overset{d}{=} \sqrt{\overline{\alpha}_{k-1}} \boldsymbol{x} + \sqrt{1 - \overline{\alpha}_{k-1}} \boldsymbol{\epsilon}'$, i.e., $\tilde{q}_{k-1} = q_{k-1}$. □

9.4 Abstract Definition of Markovian Update and Divergence Expansion

Definition 9.2 (Markovian Update $\mathrm{MUpd}_{\hat{\epsilon},\mathrm{coeff}}$)**.** Given arbitrary coefficients $\{A_k, B_k, C_k\} \subset \mathbb{R}$, define

$$z_{k+1} \coloneqq A_k\, z_k + B_k\, \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(z_k, \boldsymbol{c}, t_k) + C_k\, \boldsymbol{u}_k, \qquad \boldsymbol{u}_k \sim \mathrm{StdNormal}_d. \tag{76}$$

Proposition 9.2 (First-Order Expansion of Conditional Divergence)**.** Let $\tilde{q}_{k+1|k}$ be the exact posterior (Gaussian) from Definition 9.1. Let the distribution of z_{k+1} conditioned on $z_k = z$ be $p^{\mathrm{cond}}_{k+1|k}(\cdot \mid z)$. Then

$$\mathbb{E}_{z\sim q_k}\Big[\mathrm{D_{KL}}\big(p^{\mathrm{cond}}_{k+1|k}(\cdot \mid z) \,\|\, \tilde{q}_{k+1|k}(\cdot \mid z)\big)\Big] = \frac{1}{2}\,\mathbb{E}_{q_k}\left[\frac{\|\mu_{\mathrm{mupd}}(z) - \mu_*(z)\|_2^2}{\sigma_*^2} + \frac{(\sigma_{\mathrm{mupd}} - \sigma_*)^2}{\sigma_*^2}\right] + O(h_k^2), \tag{77}$$

where

$$\mu_{\mathrm{mupd}}(z) = A_k\, z + B_k\, \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(z, \boldsymbol{c}, t_k), \quad \sigma^2_{\mathrm{mupd}} = C_k^2, \tag{78}$$

$$\mu_*(z) = \sqrt{\tfrac{\overline{\alpha}_{k+1}}{\overline{\alpha}_k}}\, z - \sqrt{\overline{\alpha}_{k+1}}\,\tfrac{\sqrt{1-\overline{\alpha}_k}}{\sqrt{\overline{\alpha}_k}}\, \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(z, \boldsymbol{c}, t_k), \tag{79}$$

$$\sigma_*^2 = 1 - \frac{\overline{\alpha}_{k+1}}{\overline{\alpha}_k}\,. \tag{80}$$

Proof. **(1) Mean and variance of the posterior Gaussian:** In the linear Gaussian process of Definition 9.1, from standard conditional Gaussian formulas $\tilde{q}_{k+1|k} = \mathcal{N}(\mu_*(z), \sigma_*^2 \boldsymbol{I})$, and μ_*, σ_*^2 are given by (79)–(80) (identical to the derivation in [1]).

(2) Conditional of the generative side: From (76) $p^{\mathrm{cond}}_{k+1|k}(\cdot \mid z) = \mathcal{N}(\mu_{\mathrm{mupd}}(z), \sigma^2_{\mathrm{mupd}} \boldsymbol{I})$.

(3) Exact formula for KL between isotropic Gaussians: For isotropic Gaussians with means μ_1, μ_2 and variances σ_1^2, σ_2^2,

$$\mathrm{D_{KL}}(\mathcal{N}(\mu_1, \sigma_1^2 \boldsymbol{I}) \| \mathcal{N}(\mu_2, \sigma_2^2 \boldsymbol{I})) = \frac{1}{2}\left(\frac{\|\mu_1 - \mu_2\|_2^2}{\sigma_2^2} + d\left(\frac{\sigma_1^2}{\sigma_2^2} - 1 - \log\frac{\sigma_1^2}{\sigma_2^2}\right)\right). \tag{81}$$

Here d is the dimension.

(4) Small-step approximation: Since $\overline{\alpha}_{k+1}/\overline{\alpha}_k = 1 - O(h_k)$, $\sigma_*^2 = 1 - \overline{\alpha}_{k+1}/\overline{\alpha}_k = O(h_k)$. Thus $\frac{\sigma^2_{\mathrm{mupd}}}{\sigma_*^2} - 1 - \log\frac{\sigma^2_{\mathrm{mupd}}}{\sigma_*^2} = \frac{(\sigma_{\mathrm{mupd}} - \sigma_*)^2}{\sigma_*^2} + O(h_k^2)$ quadratic expansion applies. Substitute $\mu_1 = \mu_{\mathrm{mupd}}$, $\mu_2 = \mu_*$, $\sigma_1 = \sigma_{\mathrm{mupd}}$, $\sigma_2 = \sigma_*$ into (81) for this problem, and take the q_k expectation to obtain (77). □

9.5 Optimal Coefficients and DDPM/Euler a

Theorem 9.2 (Optimal Coefficients by Minimizing Divergence)**.** The coefficients that minimize the principal term on the right-hand side of Proposition 9.2 are

$$A_k^\star = \sqrt{\frac{\overline{\alpha}_{k+1}}{\overline{\alpha}_k}} + O(h_k^2), \tag{82}$$

$$B_k^\star = -\sqrt{\overline{\alpha}_{k+1}}\frac{\sqrt{1-\overline{\alpha}_k}}{\sqrt{\overline{\alpha}_k}} + O(h_k^2), \tag{83}$$

$$C_k^\star = \sqrt{1 - \frac{\overline{\alpha}_{k+1}}{\overline{\alpha}_k}} \; + \; O(h_k^2). \tag{84}$$

Proof. **(1) Point-wise minimization:** The integrand of (77) for each z is $\frac{\|\mu_{\mathsf{mupd}}(z)-\mu_*(z)\|^2}{2\sigma_*^2} + \frac{(\sigma_{\mathsf{mupd}}-\sigma_*)^2}{2\sigma_*^2}$. To minimize this, it suffices to satisfy $\mu_{\mathsf{mupd}}(z) = \mu_*(z),\ \sigma_{\mathsf{mupd}} = \sigma_*$.

(2) Coefficient matching: Comparing (78) and (79), $A_k^\star = \sqrt{\overline{\alpha}_{k+1}/\overline{\alpha}_k}$, $B_k^\star = -\sqrt{\overline{\alpha}_{k+1}}\frac{\sqrt{1-\overline{\alpha}_k}}{\sqrt{\overline{\alpha}_k}}$, $C_k^\star = \sqrt{1-\overline{\alpha}_{k+1}/\overline{\alpha}_k}$ are obtained. The $O(h_k^2)$ residuals are due to the first-order smoothness of $\overline{\alpha}$. □

Example 9.1 (DDPM and Euler a)**.** Substituting Theorem 9.2 into (76)

$$z_{k+1} = \sqrt{\frac{\overline{\alpha}_{k+1}}{\overline{\alpha}_k}}\, z_k - \sqrt{\overline{\alpha}_{k+1}}\frac{\sqrt{1-\overline{\alpha}_k}}{\sqrt{\overline{\alpha}_k}}\, \hat{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(z_k, \boldsymbol{c}, t_k) + \sqrt{1 - \frac{\overline{\alpha}_{k+1}}{\overline{\alpha}_k}}\, \boldsymbol{u}_k, \tag{85}$$

is obtained. This matches the **DDPM ancestral sample (Euler a)** [1], and makes the principal term of Proposition 9.2 zero point-wise (i.e., first-order consistency).

10 Summary and Next Time

10.1 Summary Corresponding 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.

10.2 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 [4]).

References

[1] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In NeurIPS, 2020.

[2] Tero Karras, Miika Aittala, Samuli Laine, Ari Herva, and Jaakko Lehtinen. Elucidating the design space of diffusion-based generative models. arXiv:2206.00364, 2022.

[3] Cheng Lu and Yuhao Zhou. Dpm-solver++: Fast solver for diffusion odes with optimal error bounds. arXiv preprint arXiv:2211.01095, 2022.

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

[5] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. arXiv preprint arXiv:2011.13456, 2021.