

# Step-by-Step Notes

## Fundamentals of Diffusion

### 生成模型的目标与扩散模型的基本思想

#### 生成模型的目标

生成模型的目的是给定一组来自某个未知分布 $p^*(x)$ 的独立同分布(i.i.d.)样本，构建一个采样器，能够近似地从相同的分布中生成新的样本。例如，假设我们有一组狗的图像训练集，这些图像来自某个潜在分布 $p_{\text{dog}}$ ，我们的目标是生成新的狗的图像，这些图像看起来像是从 $p_{\text{dog}}$ 中抽取的。数学上，这意味着我们希望构造一个采样器，使得其输出的样本分布尽可能接近 $p^*(x)$ 。

#### 扩散模型的基本思想

直接从复杂的 $p^*(x)$ 中采样通常是困难的。扩散模型提供了一种解决方案：通过学习从一个易于采样的分布（例如高斯噪声分布）到目标分布 $p^*(x)$ 的转换，间接实现采样。扩散模型的巧妙之处在于，它将从 $p^*(x)$ 采样的复杂问题分解成一系列更简单的子问题。具体来说，它通过一个逐步添加噪声的前向过程将数据转化为噪声，再通过一个逐步去除噪声的反向过程从噪声恢复到数据。

### 高斯扩散示例

为了直观理解扩散模型的工作原理，我们以高斯扩散为例进行讲解。这一示例将展示前向过程如何将数据变为噪声，以及反向过程如何从噪声恢复数据。

#### 前向过程

假设我们从目标分布 $p^*(x)$ 中抽取初始样本 $x_0$ ，即 $x_0 \sim p^*(x)$ 。前向过程的目标是通过逐步添加高斯噪声，将 $x_0$ 转化为一个接近标准高斯分布的样本。定义一个时间序列 $x_0, x_1, \dots, x_T$ ，其中每一步都添加一个小的独立高斯噪声：

$$x_{t+1} = x_t + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, \sigma^2 I)$$

- $x_t$ 表示时间步 $t$ 的样本。
- $\epsilon_t$ 是高斯噪声，均值为0，方差为 $\sigma^2 I$ （ $I$ 是单位矩阵）。
- $\sigma^2$ 是每步添加噪声的方差，通常取小值。

推导前向过程的结果：经过 $T$ 步后， $x_T$ 是 $x_0$ 加上 $T$ 个独立高斯噪声之和，即：

$$x_T = x_0 + \epsilon_0 + \epsilon_1 + \dots + \epsilon_{T-1}$$

由于 $\epsilon_t$ 是独立的，且每个 $\epsilon_t \sim \mathcal{N}(0, \sigma^2 I)$ ，因此总噪声的分布是：

$$\sum_{t=0}^{T-1} \epsilon_t \sim \mathcal{N}(0, T\sigma^2 I)$$

因此， $x_T$ 的分布为：

$$x_T = x_0 + \mathcal{N}(0, T\sigma^2 I)$$

当 $T$ 足够大时， $T\sigma^2$ 变得很大， $x_T$ 的分布将趋近于一个均值为0、方差为 $T\sigma^2 I$ 的高斯分布 $\mathcal{N}(0, T\sigma^2 I)$ ，几乎完全覆盖了 $x_0$ 的信息。此时， $x_T$ 可以近似看作从纯噪声分布 $p_T \approx \mathcal{N}(0, T\sigma^2 I)$ 中采样得到的。

## 反向过程

前向过程将数据 $x_0$ 转化为噪声 $x_T$ ，而反向过程的目标是逆转这一过程，从 $x_T$ 开始逐步去除噪声，恢复到 $x_0$ 。具体来说，我们需要构造一个反向采样器，能够从时间步 $t + 1$ 的分布 $p_{t+1}$ 采样得到时间步 $t$ 的分布 $p_t$ 。如果我们能解决以下子问题：

给定来自 $p_{t+1}$ 的样本，生成来自 $p_t$ 的样本。

那么我们就可以从 $p_T \approx \mathcal{N}(0, T\sigma^2 I)$ 开始，逐步应用反向采样器，生成样本序列 $x_T, x_{T-1}, \dots, x_1, x_0$ ，最终得到 $x_0 \sim p^*(x)$ 。

## 反向采样器的构建

在高斯扩散的设置中，反向过程的条件分布 $p(x_{t-\Delta t}|x_t)$ 可以被近似为一个高斯分布，这为反向采样器的设计提供了便利。

### 反向条件分布的高斯近似

一个关键事实是，当噪声水平较小时（即 $\Delta t$ 或 $\sigma^2$ 较小），反向条件分布 $p(x_{t-\Delta t}|x_t)$ 接近于高斯分布。这个事实将在后续章节中详细推导，这里我们先接受其结论，并基于此构建反向采样器。

假设 $p(x_{t-\Delta t}|x_t) \approx \mathcal{N}(\mu, \sigma^2 I)$ ，我们需要确定均值 $\mu$ 和方差 $\sigma^2$ 。

### 反向采样器的参数化

基于高斯近似，我们可以将反向采样器定义为：

$$x_{t-\Delta t} \sim \mathcal{N}(\mu_\theta(x, t), \sigma^2 I)$$

- 均值 $\mu_\theta(x, t)$ ：由神经网络参数化，输入当前样本 $x_t$ 和时间步 $t$ ，输出预测的均值。
- 方差 $\sigma^2 I$ ：通常固定为一个小的常数，表示反向步骤的噪声水平。

这种参数化简化了问题：我们只需要学习均值函数 $\mu_\theta$ ，而不需要建模复杂的分布。

## 训练反向采样器

为了使反向采样器有效，我们需要训练一个模型来预测反向条件分布的参数。在高斯扩散中，这通常通过一个去噪模型实现。

### 去噪模型

一个常见的策略是训练一个神经网络 $f_\theta(x_t, t)$ 来预测原始无噪声样本 $x_0$ 。训练过程如下：

数据生成：

- 从目标分布采样 $x_0 \sim p^*(x)$
- 通过前向过程生成 $x_t$ ： $x_t = x_0 + \sum_{i=0}^{t-1} \epsilon_i$ ， $\sum_{i=0}^{t-1} \epsilon_i \sim \mathcal{N}(0, t\sigma^2 I)$ ，或者更简洁的：

$$x_t = x_0 + \sqrt{t}\sigma_q z, \quad z \sim \mathcal{N}(0, I)$$

$$x_t \sim \mathcal{N}(x_0, \sigma_t^2), \quad \sigma_t = \sigma_q \sqrt{t}$$

损失函数：训练模型最小化预测值与真实 $x_0$ 之间的均方误差：

$$\mathcal{L} = \|f_\theta(x_t, t) - x_0\|^2$$

通过优化这个损失， $f_\theta$ 学会从带噪样本 $x_t$ 预测 $x_0$ 。

## 反向过程的采样

在采样阶段，我们从纯噪声 $x_T \sim \mathcal{N}(0, \sigma_T^2 I)$ 开始，逐步应用反向采样器：

$$x_{t-\Delta t} = \mu_\theta(x_t, t) + \sigma \cdot z, \quad z \sim \mathcal{N}(0, I)$$

步骤推导：

- 假设 $\mu_\theta(x_t, t)$ 近似于从 $x_t$ 到 $x_{t-\Delta t}$ 的均值移动。
- 添加噪声 $\sigma \cdot z$ 模拟反向过程中的随机性。
- 迭代应用此公式，从 $t = T$ 到 $t = 0$ ，最终得到 $x_0$

# Stochastic Sampling: DDPM

在本节中，我们将深入探讨 **Denoising Diffusion Probabilistic Models (DDPM)** 的随机采样机制。DDPM 是一种生成模型，通过从高斯噪声中逐步去噪生成高质量样本。

## DDPM 概述

DDPM 由 Ho 等人在 2020 年提出，是一种基于扩散过程的生成模型。其核心思想是通过以下两个过程实现数据生成：

- 前向过程**：从真实数据逐步添加高斯噪声，直到数据变成纯噪声。
- 反向过程**：从纯噪声开始，通过学习去噪模型逐步恢复到真实数据。

DDPM 的随机采样过程是其反向过程的关键，依赖于反向条件分布的高斯近似和神经网络的去噪能力。

## DDPM 的正确性

在介绍采样机制之前，我们需要证明一个关键事实：当噪声水平较小时，反向条件分布 $p(x_{t-\Delta t}|x_t)$ 可以近似为高斯分布。

### 反向条件分布的高斯近似

#### 问题重述

假设 $p_{t-\Delta t}(x)$ 是 $\mathbb{R}^d$ 上一个充分光滑的密度函数。考虑联合分布 $(x_{t-\Delta t}, x_t)$ ，其中：

$$x_t = x_{t-\Delta t} + \eta_t, \quad \eta_t \sim \mathcal{N}(0, \sigma^2 I)$$

当 $\Delta t$ 足够小（即噪声水平低）时，对于任意条件 $x_t = z \in \mathbb{R}^d$ ，反向条件分布满足：

$$p(x_{t-\Delta t}|x_t = z) \approx \mathcal{N}(\mu_z, \sigma^2 I)$$

其中  $\mu_z$  是依赖于  $z$  的均值，方差为  $\sigma^2 I$ 。

## 推导过程

我们通过贝叶斯定理推导这一近似：

1. **前向过程**：根据定义， $x_t = x_{t-\Delta t} + \eta_t$ ，其中  $\eta_t \sim \mathcal{N}(0, \sigma^2 I)$ 。这意味着前向条件分布为：

$$p(x_t|x_{t-\Delta t}) = \mathcal{N}(x_t; x_{t-\Delta t}, \sigma^2 I)$$

即  $x_t$  是以  $x_{t-\Delta t}$  为均值、方差为  $\sigma^2 I$  的高斯分布。

2. **反向条件分布**：我们需要计算  $p(x_{t-\Delta t}|x_t = z)$ 。根据贝叶斯定理：

$$p(x_{t-\Delta t}|x_t = z) = \frac{p(x_t = z|x_{t-\Delta t})p(x_{t-\Delta t})}{p(x_t = z)}$$

- **似然项**： $p(x_t = z|x_{t-\Delta t}) = \mathcal{N}(z; x_{t-\Delta t}, \sigma^2 I)$ ，因为  $x_t = x_{t-\Delta t} + \eta_t$ 。
  - **先验项**： $p(x_{t-\Delta t})$  是  $x_{t-\Delta t}$  的分布，通常假设为光滑的密度函数。
  - **归一化项**： $p(x_t = z) = \int p(x_t = z|x_{t-\Delta t})p(x_{t-\Delta t})dx_{t-\Delta t}$ 。
3. **小噪声近似**：当  $\sigma^2$  很小时，似然  $p(x_t = z|x_{t-\Delta t})$  在  $x_{t-\Delta t} = z$  附近高度集中。我们可以通过对  $\log p(x_{t-\Delta t})$  进行泰勒展开来近似整个分布：
    - **取对数**：

$$\log p(x_{t-\Delta t}|x_t = z) = \log p(x_t = z|x_{t-\Delta t}) + \log p(x_{t-\Delta t}) - \log p(x_t = z)$$

- 因为  $\log p(x_t = z|x_{t-\Delta t}) = -\frac{1}{2\sigma^2}|z - x_{t-\Delta t}|^2 + C$ （忽略常数项），当  $\sigma^2$  小且  $p(x_{t-\Delta t})$  变化平缓时，反向分布近似为高斯形式。
4. **结果**：通过拉普拉斯近似，反向分布可近似为：

$$p(x_{t-\Delta t}|x_t = z) \approx \mathcal{N}(\mu_z, \sigma^2 I)$$

其中  $\mu_z$  依赖于  $z$  和  $p(x_{t-\Delta t})$  的梯度（具体形式需进一步计算，但在 DDPM 中由神经网络估计）。

## 反向采样器的构建

基于此近似，反向采样器定义为：

$$x_{t-\Delta t} \sim \mathcal{N}(\mu_\theta(x_t, t), \sigma^2 I)$$

其中  $\mu_\theta(x_t, t)$  是神经网络参数化的均值函数，目标是学习这一函数以逆转前向噪声过程。

# DDPM 的算法

## 训练目标

DDPM 的训练目标是学习去噪模型  $f_\theta(x_t, t)$ ，使其能够预测无噪声样本  $x_0$ 。训练损失定义为：

$$L = \mathbb{E}_{t \sim \mathcal{U}[0,1], x_0 \sim p^*, \epsilon \sim \mathcal{N}(0, I)} [|f_\theta(x_t, t) - x_0|^2]$$

其中：

- $x_0 \sim p^*$ ：真实数据分布。

- $t \sim \mathcal{U}[0, 1]$ : 均匀采样的时间步。
- $\epsilon \sim \mathcal{N}(0, I)$ : 标准高斯噪声。
- $x_t = x_0 + \sqrt{t}\sigma\epsilon$ : 加噪样本。

## 训练伪代码

1. 采样时间  $t \sim \mathcal{U}[0, 1]$ 。
2. 采样数据  $x_0 \sim p^*$ 。
3. 采样噪声  $\epsilon \sim \mathcal{N}(0, I)$ 。
4. 计算加噪样本  $x_t = x_0 + \sqrt{t}\sigma\epsilon$ 。
5. 计算损失  $L = |f_\theta(x_t, t) - x_0|^2$ 。

## 3.2 采样过程

采样从纯噪声  $x_T \sim \mathcal{N}(0, \sigma_T^2 I)$  开始，逐步应用反向采样器：

$$x_{t-\Delta t} = \mu_\theta(x_t, t) + \sigma \cdot z, \quad z \sim \mathcal{N}(0, I)$$

其中  $\mu_\theta(x_t, t)$  可通过训练好的  $f_\theta$  推导。

## 采样伪代码

1. 采样初始噪声  $x_T \sim \mathcal{N}(0, \sigma_T^2 I)$ 。
2. 对于  $t = T$  到 1:
  - 计算  $\mu_\theta(x_t, t)$ 。
  - 采样  $z \sim \mathcal{N}(0, I)$ 。
  - 更新  $x_{t-\Delta t} = \mu_\theta(x_t, t) + \sigma \cdot z$ 。
3. 返回生成的  $x_0$ 。

## DDPM 与 SDE 的联系

DDPM 的离散时间过程可以看作随机微分方程（SDE）的离散化形式。

## 前向 SDE

连续时间下的扩散过程表示为：

$$dx_t = f(x_t, t)dt + g(t)dW_t$$

其中  $f(x_t, t)$  是漂移项， $g(t)$  是扩散系数，与时间  $t$  有关， $W_t$  是布朗运动。

## 反向 SDE

反向过程对应：

$$dx_t = [f(x_t, t) - g(t)^2 \nabla_x \log p_t(x_t)]dt + g(t)dW_t$$

其中  $\nabla_x \log p_t(x_t)$  是得分函数，DDPM 的  $f_\theta(x_t, t)$  实际上在近似这一项。

# Deterministic Sampling: DDIM

DDIM 是一种高效的扩散模型采样方法，通过确定性流逆转前向扩散过程，从高斯噪声生成高质量样本。与 DDPM 的随机性不同，DDIM 的确定性使其在生成过程中更可控且计算效率更高。其核心是通过学习一个速度场（velocity field），从噪声分布（如高斯分布）逐步恢复到目标数据分布（如狗的图像分布）。它利用了前向扩散过程中条件分布的高斯近似，并通过确定性更新规则避免了随机噪声的引入。

## DDIM 概述

DDIM（去噪扩散隐式模型）由 Song 等人在 2021 年提出，是一种确定性采样方法，用于从高斯噪声生成目标分布的样本。与 DDPM 的随机采样不同，DDIM 通过确定性流逆转扩散过程，减少了采样步骤并提高了计算效率。DDIM 被视为流匹配的一种特殊形式，并与概率流常微分方程（ODE）密切相关。

## 核心概念

### 确定性采样

DDIM 的核心在于其确定性逆转过程。给定一个初始噪声样本  $x_T \sim \mathcal{N}(0, \sigma_T^2 I)$ ，DDIM 通过一系列确定性步骤生成  $x_0$ ，确保每次运行结果相同。这种确定性提高了可重复性和计算效率。

### 流匹配

DDIM 被描述为流匹配的一种特殊形式。流匹配的目标是学习一个速度场  $v(x, t)$ ，将基分布  $q$ （如高斯分布）映射到目标分布  $p$ （如数据分布）。在 DDIM 中，速度场是确定性的，定义了从噪声到数据的轨迹。

### 概率流 ODE

DDIM 可以看作是对概率流 ODE 的离散化。概率流 ODE 描述了从噪声到数据的确定性路径，去除了随机扩散项：

$$\frac{dx_t}{dt} = f(x_t, t) - \frac{1}{2}g(t)^2 \nabla_x \log p_t(x_t)$$

其中：

- $f(x_t, t)$  是漂移项，描述确定性趋势。
- $g(t)$  是扩散系数。
- $\nabla_x \log p_t(x_t)$  是得分函数，指示分布的梯度方向。

**线性流：**DDIM 的轨迹可以表示为线性点流（pointwise flows）的组合，通过时间缩放调整。这些线性流简化了计算，使得 DDIM 在实践中更易实现。

**效率：**DDIM 通常需要较少的采样步骤（例如 50-100 步，而 DDPM 可能需要 1000 步），显著降低了计算成本。这得益于其确定性更新规则和对 ODE 的数值求解。

## 数学推导

### 前向扩散过程

前向扩散过程从目标分布 $p^*(x)$ 开始，逐步添加高斯噪声：

$$x_t = x_{t-1} + \eta_t, \quad \eta_t \sim \mathcal{N}(0, \sigma^2 I)$$

经过 $T$ 步， $x_T$ 接近高斯分布 $\mathcal{N}(0, T\sigma^2 I)$ 。在连续时间下，这对应于随机微分方程（SDE）：

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

其中 $dW$ 是布朗运动。

## DDIM 反向过程

DDIM 的反向过程通过确定性更新规则逆转前向扩散。关键假设是反向条件分布 $p(x_{t-1}|x_t)$ 在小噪声水平下近似为高斯分布。DDIM 的更新规则为：

$$x_{t-1} = \alpha_t x_t + \beta_t \epsilon_\theta(x_t, t)$$

- 符号说明：
  - $\alpha_t, \beta_t$ ：时间依赖的缩放系数，基于噪声调度。
  - $\epsilon_\theta(x_t, t)$ ：神经网络预测的噪声或相关量（如得分函数）。

此更新规则是确定性的，不引入随机噪声。

## 速度场

DDIM 使用速度场 $v(x, t)$ 定义点在时间上的移动方向：

$$v(x, t) \approx -\frac{1}{2}g(t)^2 \nabla_x \log p_t(x)$$

此速度场指向逆转扩散的方向，指导从 $(x_t)$ 到 $(x_{t-1})$ 的移动。离散更新为：

$$x_{t-1} = x_t + v(x_t, t)\Delta t$$

## 概率流 ODE

DDIM 的连续形式是概率流 ODE：

$$\frac{dx_t}{dt} = f(x_t, t) - \frac{1}{2}g(t)^2 \nabla_x \log p_t(x_t)$$

通过数值求解器（如 Euler 或 Runge-Kutta）离散化此 ODE，得到 DDIM 的更新规则。

## 点流与线性流

DDIM 的轨迹可以通过点流表示。对于目标分布 $p_0$ 是 Dirac-delta 分布的情况，点流定义为：

$$G_t(x_1) = x_1 - \Delta$$

其中 $\Delta$ 是从 $x_1$ 到 $x_0$ 的位移。DDIM 将这些点流组合为线性流，通过时间缩放调整轨迹。

## 推导 DDIM 更新规则

DDIM 的更新规则可以通过流匹配推导。假设目标分布 $p_0$ 是 Dirac-delta 分布，流 $v$ 将基分布 $p_1$ 映射到 $p_0$ 。离散更新为：

$$x_{t-1} = x_t - \alpha_t v(x_t, t)$$

其中 $v(x_t, t)$ 由神经网络预测， $\alpha_t$ 调整步长。

## 实际意义

**高效采样：**DDIM 的确定性使其采样步骤少于 DDPM，适合实时应用。例如，在图像生成中，DDIM 可以在 50-100 步内生成高质量图像，而 DDPM 可能需要 1000 步。

**可重复性：**由于没有随机噪声，DDIM 确保相同输入产生相同输出，便于调试和验证。

**数值求解器：**DDIM 使用 Euler 或 Runge-Kutta 等求解器离散化 ODE，提高采样效率。

## 挑战与局限性

**学习误差：**DDIM 的性能依赖于精确学习速度场 $v(x_t, t)$ 。时间依赖性 or 回归误差可能导致生成样本偏离目标分布。

**数据集规模：**在小数据集上，DDIM 可能过拟合，导致轨迹坍缩到训练点，影响泛化能力。

## 与 DDPM 的对比

特性	DDPM	DDIM
采样性质	随机	确定性
步骤数	较多 (1000+)	较少 (50-100)
计算成本	高	低
可重复性	低	高

# Flow Matching

Flow Matching 的核心目标是通过学习一个流 $v$ ，将初始分布 $q$ （通常是易于采样的，如高斯分布）运输到目标分布 $p$ （如狗的图像分布），从而生成目标分布的样本。

## 主要概念

**流与 ODE:**

- 流 $v$ 被定义为一个速度场，通过普通微分方程（ODE）描述点随时间移动。
- 例如，给定初始点 $x_1$ 在时间 $t = 1$ ，流 $v$ 将其运输到时间 $t = 0$ 的最终点 $x_f$ ，用公式表示为 $x_f = \text{RunFlow}(x_1, v, t)$
- 这个过程不仅适用于单个点，还能通过“向前推送”(pushing forward)处理整个分布的转换。

**点流(Pointwise Flows):**

- 点流是 Flow Matching 的基本构建块，描述单个点 $x_1$ 到另一个点 $x_0$ 的轨迹，通过在路径上每个点的速度 $v_t(x)$ 定义。
- 这可以直观地理解为粒子在空间中的移动，比如气体分子从一个位置移动到另一个位置。
- 一个简单的例子是线性流，其中点沿直线从 $x_1$ 移动到 $x_0$

**分布运输:**



- 流不仅能映射单个点，还能运输整个分布。如果初始分布  $p_1$  在  $t = 1$ ，应用流  $v$  后得到最终分布  $p_0$ ，用符号表示为  $p_1 \xrightarrow{v} p_0$
- Flow Matching 的目标是找到一个流  $v$ ，使基础分布  $q$ （如高斯）被运输到目标分布  $p$ 。

基础分布与目标分布:

- 基础分布  $q$  通常是易于采样的分布，比如高斯分布或环形分布。
- 目标分布  $p$  是我们希望生成样本的分布，比如狗的图像分布。
- 使用耦合  $\Pi_{q,p}$  来联合采样来自  $q$  和  $p$  的点，最简单的选择是独立耦合，即从每个分布独立采样点。

与扩散模型的关系

- Flow Matching 与扩散模型（如 DDPM 和 DDIM）有密切联系，尤其是与 DDIM 共享许多思想。
- 它被描述为 DDIM 的推广，DDIM 算法是 Flow Matching 的一种特殊情况，特别是在特定流  $v$  的选择下。
- 与扩散模型不同，Flow Matching 避免了显式的向前加噪过程，专注于学习确定性的速度场，简化了生成建模的学习问题。

## 实现细节

Flow Matching 的实现需要以下关键选择：

1. **基础分布  $q$** : 通常选择易于采样的分布，如高斯分布。
2. **耦合  $\Pi_{q,p}$** : 定义点对  $(x_1, x_0)$  的联合分布，其中  $x_1 \sim q$ ,  $x_0 \sim p$ 。常见选择是独立耦合，即  $x_1$  和  $x_0$  独立采样。
3. **点流**: 定义从  $x_1$  到  $x_0$  的轨迹，例如线性流，点沿直线移动。

## 训练

训练目标是学习一个神经网络  $f_\theta$ ，逼近边缘流  $v_t^*(x_t)$ 。

伪代码4: Flow Matching 训练损失

- 输入: 神经网络  $f_\theta$
- 数据: 耦合  $\Pi_{q,p}$  的采样访问
- 步骤

:

1. 从  $\Pi_{q,p}$  采样  $(x_1, x_0)$
2. 选择点流  $v^{[x_1, x_0]}$
3. 对于时间  $t \in [0, 1]$ ，计算  $x_t = \text{RunFlow}(v^{[x_1, x_0]}, x_1, t)$
4. 计算速度  $v_t^{[x_1, x_0]}(x_t)$
5. 计算损失:
 
$$\mathbb{E} * t \sim \text{Uniform}[0, 1] \mathbb{E} * (x_1, x_0) \left[ |f_\theta(x_t, t) - v_t^{[x_1, x_0]}(x_t)|^2 \right]$$

此损失函数促使  $f_\theta$  在轨迹上匹配点流的速度。

## 采样

训练完成后，通过求解  $f_\theta$  定义的常微分方程（ODE）从目标分布  $p$  采样。

## 伪代码5：采样

- 输入：训练好的神经网络  $f_\theta$
- 步骤：
  1. 从  $q$  采样  $x_1$
  2. 初始化  $x = x_1$
  3. 对于从  $t = 1$  到  $t = 0$ ，以小步长  $\Delta t$  迭代：
    - 更新  $x \leftarrow x + f_\theta(x, t) \cdot \Delta t$
  4. 在  $t = 0$  时的  $x$  为  $p$  的样本

## 与DDIM的关系

Flow Matching是DDIM（去噪扩散隐式模型）的推广。DDIM可以看作Flow Matching在特定点流和耦合选择下的特例。

**（DDIM等价于线性流）**：DDIM采样器在时间重参数化下等价于使用线性点流和扩散耦合产生的边缘流。

具体地，使用线性点流  $v_t^{[x_1, x_0]}(x_t) = \frac{1}{2t}(x_t - x_0)$  或  $v_t^{[x_1, x_0]}(x_t) = \frac{1}{2\sqrt{t}}(x_0 - x_1)$ ，结合扩散耦合  $x_1 = x_0 + \mathcal{N}(0, \sigma_q^2)$ ，可重现DDIM轨迹。

## 额外备注

- **无需显式前向过程**：与传统扩散模型不同，Flow Matching直接学习逆向过程，通过流定义。
- **点流别名**：点流也称为双侧条件流，见Albergo和Vanden-Eijnden [2022]等。
- **进一步阅读**：Fjelde等人 [2024] 提供了详细的可视化和数学阐述。
- **扩展**：随机插值（Albergo等人 [2023]）和非标准流（Stark等人 [2024]）将框架扩展到离散空间等场景。

## 实践中的扩散模型

### 实用采样器

- **DDPM和DDIM**：Ho等人 [2020] 和Song等人 [2021] 提出的标准采样算法。
- **减少步数生成**：使用高级数值求解器（如Euler、Heun、Runge-Kutta）减少采样步数。
- **时间蒸馏**：如一致性模型（Song等人 [2023b]）和对抗性蒸馏（Lin等人 [2024]）等方法提高效率。

### 噪声调度

噪声调度  $\sigma_t$  决定每一步添加的噪声量。

- **简单扩散**：  $p(x_t) \sim \mathcal{N}(x_0, \sigma_t^2)$ ，通常  $\sigma_t \propto \sqrt{t}$ 。
- **变异保持调度**：如Ho等人 [2020]，更新公式为：

$$x_t = \sqrt{1 - \beta(t)}x_{t-1} + \sqrt{\beta(t)}\varepsilon_t, \quad 0 < \beta(t) < 1$$

选择合适的噪声调度对模型性能至关重要。

# 似然解释与变分自编码器

扩散模型可通过变分自编码器（VAE）的视角解释：

- **前向过程**：作为编码器，将数据映射到潜在空间。
- **逆向过程**：作为解码器，重建数据。
- **训练目标**：可从VAE的证据下界（ELBO）推导，简化为带时间加权的L2回归损失，支持最大似然训练。

## 参数化方法：e-预测与 $\theta$ -预测

扩散模型使用不同的参数化方式预测不同量：

- **e-预测**：通常预测添加到数据的噪声分量。
- **$\theta$ -预测**：模型预测向量  $v = \alpha_v \cdot c_{x_v}$ ，具体含义需进一步上下文。

这些参数化方式影响训练稳定性和生成样本质量。