# Statistical Computing
# Chap. 5.2: Diffusion Models

LIU, Ran

April 22, 2025

# Summary

# Summary

# 基本思想概览

Diffusion Probabilistic Models（DPM）是一种基于非平衡热力学的生成模型，其基本思想如下：

- 将一个复杂的数据分布，通过一个逐步加噪的马尔可夫过程，扰动成简单的高斯分布。(变分逼近后验分布，$q_\phi(z|x) \to p_\theta(z|x)$)
- 然后学习一个反向的马尔可夫过程，从高斯分布逐步还原出原始数据，实现抽样生成。(似然函数参数学习，$p_\theta(x)$)
- 正向过程类似一个热力学扩散过程，不可逆；反向过程则是学习的近似可逆过程。

该模型为后来的 Denoising Diffusion Probabilistic Models (DDPM) 提供了理论基础。

# 正向过程（Forward Process）

正向过程通过 $T$ 步将数据 $\mathbf{x}_0$ 逐步扰动为噪声：

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

- 每步是一个高斯转移：$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$
- $\beta_t$：预先定义的噪声调度参数（通常 $0 < \beta_t < 1$）

利用重参数化技巧，单步采样可表示为：

$$\mathbf{x}_t = \sqrt{1-\beta_t}\mathbf{x}_{t-1} + \sqrt{\beta_t}\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$$

递归展开可得到从 $\mathbf{x}_0$ 直接到 $\mathbf{x}_t$ 的闭合形式。

**定义** $\alpha_t = 1 - \beta_t$，$\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$，**通过递归展开**：

$$\begin{aligned}
\mathbf{x}_t &= \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1} \\
&= \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{\alpha_t(1-\alpha_{t-1})}\boldsymbol{\epsilon}_{t-2} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1} \\
&= \cdots \\
&= \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}
\end{aligned}$$

**其中** $\epsilon \sim \mathcal{N}(0, \mathbf{I})$。**因此**：

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

这样定义的正向过程有两个非常重要的优势：

- 有 $x_0$ 到 $x_t$ 的直接表达式。
- 这个正向过程的逆分布容易表示，即 $q(x_{t-1} \mid x_t, x_0)$，算 ELBO 的时候可对应分解。

# 反向过程（Reverse Process）

反向过程定义了一个学习的马尔可夫链，用于从高斯分布逐步还原出数据：

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

初始状态为：

$$p(\mathbf{x}_T) = \mathcal{N}(\mathbf{0}, \mathbf{I})$$

每一步的反向转移分布也建模为高斯：

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t))$$

其中 $\boldsymbol{\mu}_\theta$ 和 $\Sigma_\theta$ 是通过神经网络预测的参数。

# 统计语言叙述

回顾之前统计模型定义，我们有观测数据 $x$，隐变量 $z$，待估计的模型参数 $\theta$。对应 DPM 模型的反向过程，

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

其实是在建模 $p_\theta(x, z)$，此时 $x$ 是观测数据 $x_0$，$z$ 是隐变量 $x_1, x_2, ..., x_T$。由于 $p_\theta(x_1, x_2, ..., x_T|x_0)$ 难以求得，故我们使用变分推断，即 DPM 的正向过程：

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

相当于定义了 $q_\phi(z \mid x)$。特别的是，这里跟 VAE 不同，变分分布是固定的，不含参数。

其中 diffusion 中的前向变分过程也可以是可学习的。

Kingma, D., Salimans, T., Poole, B., & Ho, J. (2021). Variational diffusion models. Neurips, 34, 21696-21707.

Bartosh, G., Vetrov, D. P., & Andersson Naesseth, C. (2024). Neural flow diffusion models: Learnable forward process for improved diffusion modelling. Neurips, 37, 73952-73985.

这是针对普通 diffusion 的弊端：

- 对所有数据统一加噪，不够灵活；
- 对某些复杂的数据分布（如多模态、高维）建模能力受限；
- 没有考虑数据的结构或语义信息。

但其实可能 latent diffusion model，即先针对 data 做 embedding 再扩散就够了。

# Summary

# 训练目标：变分下界（ELBO）

参考 VEM，我们训练目标为 ELBO：

$$\mathcal{L}(\theta, \phi; x) = \mathbb{E}_{q_\phi(z|x)} \left[ \log p_\theta(x, z) - \log q_\phi(z \mid x) \right]$$

代入 DPM 中，我们有

$$\log p_\theta(\mathbf{x}_0) \geq \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[ \log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \right]$$

记为：

$$\mathcal{L}_{\text{ELBO}} = \mathbb{E}_q \left[ \log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \right]$$

我们将逐步展开并推导该表达式。

首先，生成分布为：（黑板推导）

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

正向变分分布为：

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

代入 ELBO：

$$\mathcal{L}_{\text{ELBO}} = \mathbb{E}_q \left[ \log \frac{p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{\prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1})} \right]$$

拆分后得到：

$$\mathcal{L}_{\text{ELBO}} = \mathbb{E}_q \left[ \log p(\mathbf{x}_T) + \sum_{t=1}^{T} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1})} \right]$$

为了更好地训练模型（凑 KL 散度。正态的 KL 散度有解析表达式），我们推导正向过程的逆分布：

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1})q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)}, \quad t > 1,$$

我们有

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

提问：为什么不是 $q(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$？

## 代入到 ELBO 公式里，我们有

$$
\begin{aligned}
\mathcal{L}_{\text{ELBO}} &= \mathbb{E}_q \left[ \log p(\mathbf{x}_T) + \sum_{t=1}^{T} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1})} \right] \\
&= \mathbb{E}_q \left[ \log p(\mathbf{x}_T) + \sum_{t=2}^{T} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_t|\mathbf{x}_{t-1})} + \log \frac{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_1|\mathbf{x}_0)} \right] \\
&= \mathbb{E}_q \left[ \log p(\mathbf{x}_T) + \sum_{t=2}^{T} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)} \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)} + \log \frac{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_1|\mathbf{x}_0)} \right] \\
&= \mathbb{E}_q \left[ \log p(\mathbf{x}_T) + \sum_{t=2}^{T} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)} + \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)}{q(\mathbf{x}_T|\mathbf{x}_0)} + \log \frac{p_\theta(\mathbf{x}_0|\mathbf{x}_1)}{q(\mathbf{x}_1|\mathbf{x}_0)} \right] \\
&= \mathbb{E}_q \left[ \log \frac{p(\mathbf{x}_T)}{q(\mathbf{x}_T|\mathbf{x}_0)} + \sum_{t=2}^{T} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)} + \log p_\theta(\mathbf{x}_0|\mathbf{x}_1) \right]
\end{aligned}
$$

其中 $\mathbb{E}_q$ 指的是 $\mathbb{E}_{q(x_1,x_2,\ldots,x_T|x_0)}$.

## 又因为期望的边际化定理

$$\begin{aligned}\mathbb{E}_{q(\mathbf{x}_1,\ldots,\mathbf{x}_T|\mathbf{x}_0)}[f(\mathbf{x}_t,\mathbf{x}_{t-1})] &= \mathbb{E}_{q(\mathbf{x}_t,\mathbf{x}_{t-1}|\mathbf{x}_0)}[f(\mathbf{x}_t,\mathbf{x}_{t-1})] \\ &= \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)}\left[\mathbb{E}_{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)}[f(\mathbf{x}_t,\mathbf{x}_{t-1})]\right]\end{aligned}$$

将上述表达式重写为 KL 散度项的和:

$$
\begin{aligned}
&\mathcal{L}_{\text{ELBO}} \\
&= \mathbb{E}_{q(\mathbf{x}_T|\mathbf{x}_0)}\left[\log \frac{p(\mathbf{x}_T)}{q(\mathbf{x}_T|\mathbf{x}_0)}\right] + \mathbb{E}_{q(\mathbf{x}_t,\mathbf{x}_{t-1}|\mathbf{x}_0)}\left[\sum_{t=2}^{T} \log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)}\right] \\
&\qquad + \mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)}\left[\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)\right] \\
&= -\mathsf{KL}(q(\mathbf{x}_T|\mathbf{x}_0)\|p(\mathbf{x}_T)) - \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)} \sum_{t=2}^{T} \mathsf{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)\|p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)) \\
&\qquad + \mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)}\left[\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)\right] \\
&= \mathbb{E}_q\Bigg[ - \mathsf{KL}(q(\mathbf{x}_T|\mathbf{x}_0)\|p(\mathbf{x}_T)) - \sum_{t=2}^{T} \mathsf{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)\|p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)) \\
&\qquad + \log p_\theta(\mathbf{x}_0|\mathbf{x}_1)\Bigg]
\end{aligned}
$$

因此，训练目标可以分为三个部分，前面两项都是高斯之间的 KL 散度，可以继续化简。

训练扩散模型的目标是最小化负的 ELBO：

$$\mathcal{L}_{\text{train}} = \mathcal{L}_T + \sum_{t=2}^{T} \mathcal{L}_{t-1} + \mathcal{L}_0$$

其中：

$$\mathcal{L}_T = \mathsf{KL}(q(\mathbf{x}_T|\mathbf{x}_0) \| p(\mathbf{x}_T))$$

$$\mathcal{L}_{t-1} = \mathsf{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \| p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))$$

$$\mathcal{L}_0 = -\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)$$

要注意的是，若 diffusion 模型没有变分参数，其实有一部分可以看作常数。

# 训练目标中的 KL 项推导

我们重点关注中间 KL 项，这一项通常在主导地位：

$$\mathcal{L}_{t-1} = \mathsf{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \,\|\, p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))$$

两者都是高斯分布：

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t \mathbf{I})$$
$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \Sigma_\theta)$$

由于分布都是高斯，我们可以通过联合分布计算出后验 $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ 的参数。

我们知道：

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

$$q(\mathbf{x}_{t-1}|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0, (1-\bar{\alpha}_{t-1})\mathbf{I})$$

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

根据 Bayes 规则：

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

两个高斯分布的乘积仍然是高斯分布，可得条件分布的均值和方差。

**最终得到**:

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\boldsymbol{\mu}}_t, \tilde{\beta}_t \mathbf{I})$$

**其中均值为**:

$$\tilde{\boldsymbol{\mu}}_t(x_0, x_t) = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t$$

**方差为**:

$$\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$$

中间项 KL 散度的具体表达形式则为：

$$\mathcal{L}_{t-1} = \mathrm{KL}\left(\mathcal{N}(\tilde{\boldsymbol{\mu}}_t, \tilde{\beta}_t \mathbf{I}) \,\|\, \mathcal{N}(\boldsymbol{\mu}_\theta, \Sigma_\theta)\right)$$

训练时，常设定 $\Sigma_\theta = \sigma_\theta^2 \mathbf{I}$，那么 KL 散度具有以下解析形式：

$$\mathcal{L}_{t-1} = \frac{1}{2}\left[\frac{\tilde{\beta}_t}{\sigma_\theta^2} d + \frac{\|\tilde{\boldsymbol{\mu}}_t - \boldsymbol{\mu}_\theta\|^2}{\sigma_\theta^2} - d + d \log \frac{\sigma_\theta^2}{\tilde{\beta}_t}\right]$$

其中 $d$ 是数据维度（如图像像素数）。在训练中，常将模型预测方差固定为后验方差 $\sigma_\theta^2 = \tilde{\beta}_t$，此时，KL 简化为：

$$\mathcal{L}_{t-1} = \frac{1}{2\tilde{\beta}_t} \|\tilde{\boldsymbol{\mu}}_t - \boldsymbol{\mu}_\theta\|^2$$

# 训练目标简化

**我们用 $x_t$ 表示 $x_0$:**

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0, \mathbf{I})$$

$$x_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}\left(x_t - \sqrt{1-\bar{\alpha}_t}\epsilon\right)$$

**前向过程逆分布 $q(x_{t-1} \mid x_t, x_0)$ 的均值为:**

$$\begin{aligned}\tilde{\mu}_t(x_t, x_0) &= \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t \\ &= \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon\right)\end{aligned}$$

**将 $\tilde{\mu}_t$ 的表达式代入之后:**

$$\mathcal{L}_{t-1} = \mathbb{E}_{x_t,\epsilon}\left[\frac{1}{2\tilde{\beta}_t}\left\|\frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon\right) - \mu_\theta(x_t, t)\right\|^2\right]$$

我们令模型预测噪声 $\epsilon$（式子对应），从而构造如下参数化：

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}} \left( x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right)$$

其中 $\epsilon_\theta(x_t, t)$ 是神经网络预测的噪声。将其代入前式，有：

$$\mathcal{L}_{t-1} = \mathbb{E}_{x_t, \epsilon} \left[ \frac{\beta_t^2}{2\tilde{\beta}_t \alpha_t (1-\bar{\alpha}_t)} \left\| \epsilon - \epsilon_\theta(x_t, t) \right\|^2 \right]$$

提问：有没有可能不是用 $x_t$ 代替 $x_0$，而是用 $x_0$ 代替 $x_t$，然后用 $\mu_\theta(x_0, t)$？

令 $x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$，最终训练损失为 (随机梯度下降，所以没用所有 $x_0$ 的损失)：

$$\mathbb{E}_{x_0,\epsilon,t}\left[w_t \cdot \left\|\epsilon - \epsilon_\theta\left(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon,\, t\right)\right\|^2\right]$$

其中权重：

$$w_t = \frac{\beta_t^2}{2\tilde{\beta}_t \alpha_t (1-\bar{\alpha}_t)}$$

在实际实现中 (DDPM, Ho et al. (2020))，常将所有时间步等权处理，简化为 MSE 损失：

$$\mathbb{E}_{x_0,\epsilon,t}\left[\|\epsilon - \epsilon_\theta(x_t, t)\|^2\right]$$

其中 $\epsilon$ 是正向过程中真实加入的高斯噪声，$\epsilon_\theta(\mathbf{x}_t, t)$ 是神经网络的输出，$\mathbf{x}_t$ 是通过公式构造的：

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\epsilon$$

# Training

**Algorithm 1** *

**Algorithm 1** Training

1: **repeat**
2: $\mathbf{x}_0 \sim p(\mathbf{x}_0)$
3: $t \sim \text{Uniform}(\{1, \dots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$
5: Take gradient descent step on

$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \epsilon_\theta \left( \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \boldsymbol{\epsilon}, t \right) \right\|^2$$

6: **until** converged

# Sampling

**Algorithm 2 ***

**Algorithm 2** Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$
2: **for** $t = T, \ldots, 1$ **do**
3: $\quad \mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$ **if** $t > 1$, **else** $\mathbf{z} = 0$
4:

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left( \mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$$

5: **end for**
6: **return** $\mathbf{x}_0$

# Summary

# NLL

训练时我们最大化 ELBO，用于学习模型参数。评估时，我们希望直接衡量模型对数据的拟合能力。

负对数似然（NLL）是标准的评估指标：

$$\text{NLL}(\mathbf{x}) = -\log p_\theta(\mathbf{x})$$

NLL 越小，表示模型对观测数据的解释能力越强。

# NLL 与 ELBO 的关系

ELBO 是对数边际似然的下界：

$$\log p_\theta(\mathbf{x}) \geq \mathcal{L}_{\text{ELBO}}(\theta)$$

因此：

$$\text{NLL}(\mathbf{x}) = -\log p_\theta(\mathbf{x}) \leq -\mathcal{L}_{\text{ELBO}}(\theta)$$

训练中最大化 ELBO，相当于最小化 NLL 的上界。在扩散模型中，我们有：

$$\begin{aligned} -\mathcal{L}_{\text{ELBO}}(\theta) = \ & \text{KL}(q(\mathbf{x}_T|\mathbf{x}_0)\|p(\mathbf{x}_T)) \\ & + \sum_{t=2}^{T} \text{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)\|p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)) - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1) \end{aligned}$$

其中前两项 KL 散度因为是正态，都可以解析求得，而第三项 $\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)$，我们做近似计算：

- 图像数据 $\mathbf{x}_0$ 原本是 $\{0, \ldots, 255\}$ 的整数，为了扩散线性缩放到了 $[-1, 1]$ 区间。
- DDPM 用高斯分布来近似离散像素的概率：

$$p_\theta(\mathbf{x}_0|\mathbf{x}_1) = \prod_{i=1}^{D} \int_{\delta_-(x_0^i)}^{\delta_+(x_0^i)} \mathcal{N}(x; \mu_\theta^i(\mathbf{x}_1), \sigma_1^2) dx$$

- 每个像素值对应一个高斯分布积分区间（即一个量化 bin）。对于像素值 $x$，其积分上下界为：

$$\delta_+(x) = \begin{cases} \infty & x = 1 \\ x + \frac{1}{255} & x < 1 \end{cases}, \quad \delta_-(x) = \begin{cases} -\infty & x = -1 \\ x - \frac{1}{255} & x > -1 \end{cases}$$

# bits per dimension (bpd)

为了比较不同模型或数据集上的性能，我们通常将 NLL 标准化为 bits per dimension (bpd)：

$$\text{bpd} = \frac{-\log_2 p_\theta(\mathbf{x}_0)}{D}$$

其中 $D$ 是图像的维度（像素数 $\times$ 通道数）。bpd 是图像生成模型中常用的评估指标。

# Inception Score (IS)

- 衡量生成图像的质量，定义如下：

$$\text{IS} = \exp\left(\mathbb{E}_{x\sim p_g(x)}\left[D_{\text{KL}}\left(p(y|x)\,\|\,p(y)\right)\right]\right)$$

- 其中：
  - $x \sim p_g(x)$ 表示从生成模型采样得到的图像；
  - $p(y|x)$ 是 Inception v3 网络对图像 $x$ 的预测类别分布；
  - $p(y) = \mathbb{E}_{x\sim p_g(x)}[p(y|x)]$ 为所有生成图像的平均预测分布。
- 分数越高表示图像更清晰且更具多样性。

# 为什么 IS 能衡量生成模型好坏？

- 图像清晰度：
  - 如果 $p(y|x)$ 非常集中（低熵），说明图像易分类，图像清晰。
- 图像多样性：
  - 如果 $p(y)$ 分布接近均匀（高熵），说明生成图像覆盖了多种语义类别。
- KL 散度的作用：
  - $D_{\mathrm{KL}}(p(y|x)\|p(y))$ 在 $p(y|x)$ 低熵且 $p(y)$ 高熵时取得较大值。
  - 平均后取指数，IS 越高，说明图像既清晰又多样。

# IS 具体计算流程

1. 使用训练好的生成模型（如扩散模型）生成大量图像 $x$（例如 50,000 张）；
2. 将每张图像输入 Inception v3 网络，获得预测类别分布 $p(y|x)$；
3. 计算所有图像的平均预测类别分布：

$$p(y) = \frac{1}{N} \sum_{i=1}^{N} p(y|x_i)$$

4. 对每张图像计算 KL 散度：

$$D_{\mathrm{KL}} \left( p(y|x_i) \,\|\, p(y) \right)$$

5. 求所有图像的 KL 平均并取指数，得到最终 IS 分数：

$$\mathrm{IS} = \exp \left( \frac{1}{N} \sum_{i=1}^{N} D_{\mathrm{KL}} \left( p(y|x_i) \,\|\, p(y) \right) \right)$$

# Fréchet Inception Distance (FID)

- 衡量生成图像与真实图像在特征空间中的分布差异;
- 将图像通过 Inception 网络提取特征（如 pool3 层），假设这些特征服从多维高斯分布;
- 定义如下:

$$\text{FID} = \|\mu_r - \mu_g\|^2 + \text{Tr}(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{1/2})$$

- 其中:
  - $\mu_r, \Sigma_r$: 真实图像特征的均值和协方差矩阵;
  - $\mu_g, \Sigma_g$: 生成图像特征的均值和协方差矩阵;
  - $(\Sigma_r\Sigma_g)^{1/2}$ 表示协方差矩阵乘积的矩阵平方根。
- 分数越低表示生成图像分布越接近真实图像，更贴近人类感知。

# 为什么 FID 能反映生成质量？

- FID 比较的是生成图像和真实图像在语义特征空间中的分布差异；
- 使用 Inception 网络提取的特征具有一定的语义信息；
- 假设这些特征服从高斯分布，FID 实际计算的是两个高斯分布之间的 Fréchet 距离（也称 Wasserstein-2 距离）；
- FID 兼顾：
  - **均值差异**：衡量图像整体风格是否一致；
  - **协方差差异**：衡量图像多样性是否一致。
- 具有更强的鲁棒性，能检测到 mode collapse。

# FID 的具体计算流程

1. 使用生成模型（如扩散模型）生成 $N$ 张图像（通常 $N = 50{,}000$）；
2. 从真实数据集中选取 $N$ 张图像作为参考（要跟生成模型是同一类数据集）；
3. 使用 Inception v3 网络提取两组图像的特征（通常为 pool3 层，2048 维）；
4. 分别计算两组特征的均值和协方差：

$$\mu_r, \Sigma_r \quad \text{和} \quad \mu_g, \Sigma_g$$

5. 代入 Fréchet 距离公式计算：

$$\text{FID} = \|\mu_r - \mu_g\|^2 + \text{Tr}(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{1/2})$$

**说明**：FID 越低，表示生成图像的特征分布越接近真实图像，生成效果越好。

| 指标 | 趋势 | 意义 | 特点 |
|---|---|---|---|
| IS | 越高越好 | 图像清晰 + 多样性高 | 简单但敏感 |
| FID | 越低越好 | 分布接近真实 | 鲁棒，贴近人类感知 |
| NLL | 越低越好 | 拟合真实概率分布 | 有理论解释 |

Table 1: CIFAR10 results. NLL measured in bits/dim.

| Model | IS | FID | NLL Test (Train) |
|---|---|---|---|
| **Conditional** | | | |
| EBM [11] | 8.30 | 37.9 | |
| JEM [17] | 8.76 | 38.4 | |
| BigGAN [3] | 9.22 | 14.73 | |
| StyleGAN2 + ADA (v1) [29] | **10.06** | **2.67** | |
| **Unconditional** | | | |
| Diffusion (original) [53] | | | $\leq 5.40$ |
| Gated PixelCNN [59] | 4.60 | 65.93 | 3.03 (2.90) |
| Sparse Transformer [7] | | | **2.80** |
| PixelIQN [43] | 5.29 | 49.46 | |
| EBM [11] | 6.78 | 38.2 | |
| NCSNv2 [56] | | 31.75 | |
| NCSN [55] | $8.87 \pm 0.12$ | 25.32 | |
| SNGAN [39] | $8.22 \pm 0.05$ | 21.7 | |
| SNGAN-DDLS [4] | $9.09 \pm 0.10$ | 15.42 | |
| StyleGAN2 + ADA (v1) [29] | $\mathbf{9.74} \pm 0.05$ | 3.26 | |
| Ours ($L$, fixed isotropic $\boldsymbol{\Sigma}$) | $7.67 \pm 0.13$ | 13.51 | $\leq 3.70$ (3.69) |
| **Ours** ($L_{\text{simple}}$) | $9.46 \pm 0.11$ | **3.17** | $\leq 3.75$ (3.72) |

# Summary



Stable Diffusion

# Stable Diffusion 的动机

- 传统的扩散模型在像素空间进行建模，计算开销大，推理速度慢
- 高分辨率图像生成效率低，资源占用高
- Stable Diffusion 的核心思想是在潜空间中进行扩散过程
- 利用预训练 encoder 将图像压缩到低维潜空间，减少计算量
- 原始的 DDPM 最后一层是简单的独立高斯解码器 $p_\theta(x_0 \mid x_1)$，最终采样是用的这一层高斯的均值，而 Stable Diffuison 这里是 VAE 的 decoder 了。

# 模型结构

- 整体结构包含三个主要模块：
  1. 已经训练好的 VAE：将图像映射到潜空间
  2. 潜空间 U-Net：在潜空间中进行扩散建模
  3. 文本编码器（如 CLIP）：为图像生成提供文本条件
- 潜空间比原始图像空间小得多，效率更高

# 潜空间中的扩散过程

- 设图像经过训练好的编码器得到潜向量 $z = \text{Encoder}(x)$（其实是用的输出的均值 $\mu$）
- 在潜空间中添加高斯噪声，构造扩散过程 $z_t$
- 使用 U-Net 预测噪声 $\epsilon_\theta(z_t, t, c)$，c 是条件
- 最终通过解码器还原图像 $\hat{x} = \text{Decoder}(z_0)$

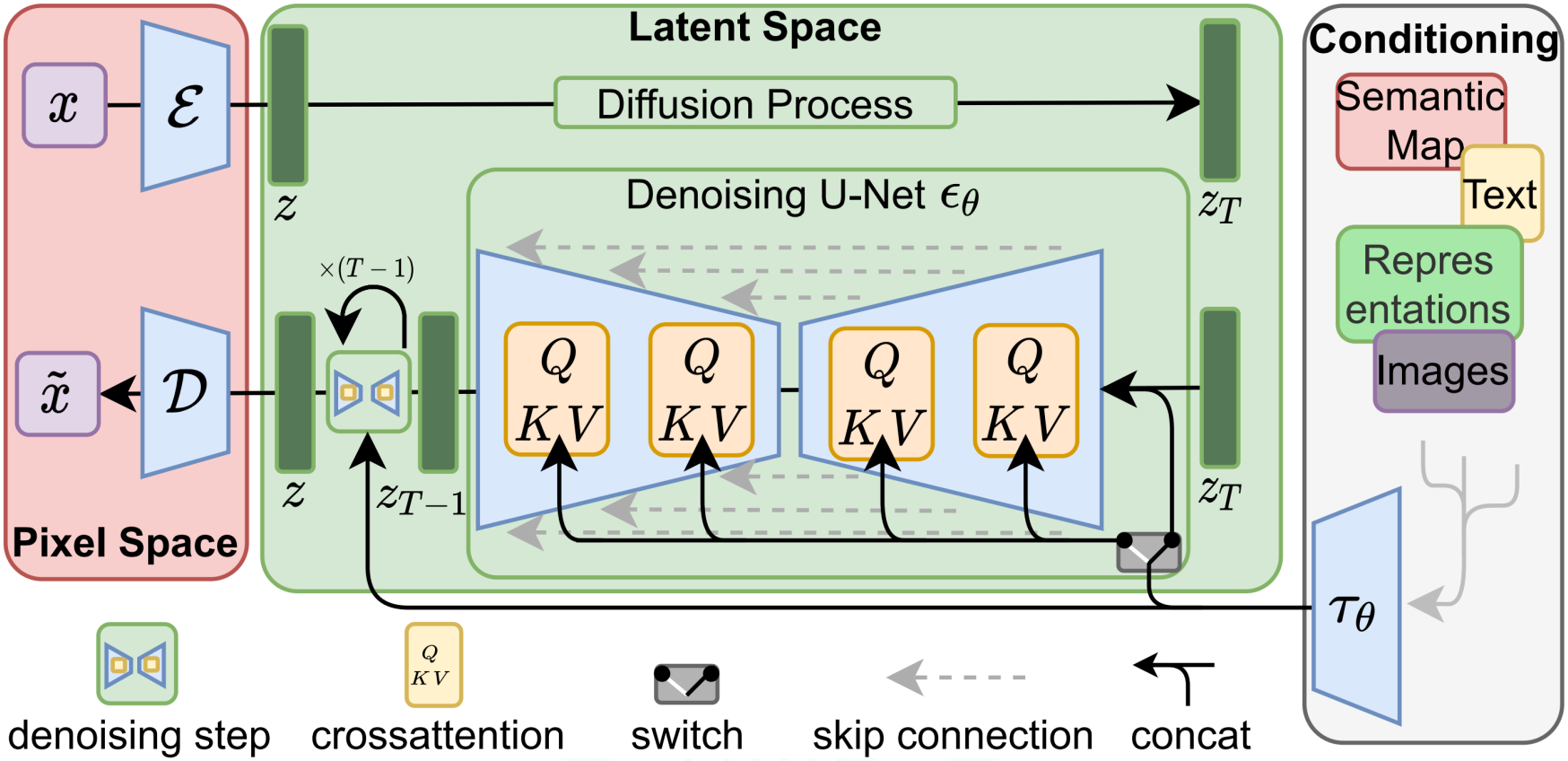
$$\mathcal{L}_{\text{simple}} = \mathbb{E}_{z,t,\epsilon}\left[\|\epsilon - \epsilon_\theta(z_t, t, c)\|^2\right]$$

# 文本引导机制

- 使用预训练的文本编码器将文本转换为条件向量 $c$
- 训练时有部分样本使用空条件（类似 dropout），支持无条件建模 (Classifier-free Guidance)
- 推理采样时引入引导策略提升文本一致性：

$$\epsilon_{\text{guidance}} = (1 + w) \cdot \epsilon_\theta(z_t, t, c) - w \cdot \epsilon_\theta(z_t, t, \varnothing)$$

- $w$ 为引导强度，调节生成图像与文本的匹配程度



# 模型优势

- 潜空间建模显著降低计算成本
- 支持高质量图像生成，速度快，内存占用小
- 结合强大的文本编码器，提升语义控制能力

Ref: https://jalammar.github.io/illustrated-stable-diffusion/