目录

第 1 章 能量模型

1.1 定义

能量模型（Energy-Based Models）是一种概率模型，为了简单起见，这里只介绍单个因变量 x 上的无条件能量模型。对于某个 x，其概率密度可以写为：

$$p_\theta(x) = \frac{\exp\left(-E_\theta(x)\right)}{Z_\theta}, \tag{1.1}$$

其中 $E_\theta(x)$ 是 x 的能量，$E_\theta(\cdot)$ 是参数为 θ 的非线性回归方程，$Z_\theta = \int \exp\left(-E_\theta(x)\right) dx$ 是归一化常数，其与 x 无关但与 θ 相关。与其它大部分概率模型不同的是，能量模型将归一化因子独立出来，对 Z_θ 的可处理性没有限制，因此能量方程具有更大的灵活性，建模能力也更强。目前主要有三种训练能量模型的方法，分别是基于马尔可夫链蒙特卡洛采样（MCMC sampling）的极大似然法、分数匹配法（Score Matching）和噪声对比估计法（Noise Constrastive Estimation），本章只涉及前两种。

1.2 极大似然法

极大似然法的学习目标为最大化能量模型在数据分布 $p_{\text{data}}(x)$ 上的期望对数似然：

$$\mathbb{E}_{x\sim p_{\text{data}}(x)}\left[\log p_\theta(x)\right], \tag{1.2}$$

该目标等价于最小化 $p_\theta(x)$ 和 $p_{\text{data}}(x)$ 之间的 KL-散度（即相对熵，用于衡量两个分布之间的差异程度）：

$$\begin{aligned} -\mathbb{E}_{x\sim p_{\text{data}}(x)}\left[\log p_\theta(x)\right] &= \mathbb{E}_{x\sim p_{\text{data}}(x)}\left[\log \frac{p_{\text{data}}(x)}{p_\theta(x)}\right] - \mathbb{E}_{x\sim p_{\text{data}}(x)}\left[\log p_{\text{data}}(x)\right] \\ &= D_{KL}\left(p_{\text{data}}(x) \,\|\, p_\theta(x)\right) - \text{constant}. \end{aligned} \tag{1.3}$$

我们一般使用梯度下降法优化 θ，对于式 1.2中的期望，我们可以使用蒙特卡洛法进行模拟，因此，我们只需要求得 $\nabla_\theta \log p_\theta(x)$：

$$\nabla_\theta \log p_\theta(x) = -\nabla_\theta E_\theta(x) - \nabla_\theta \log Z_\theta, \tag{1.4}$$

其中第一项为可以由自动微分计算，主要困难在于第二项的计算：

$$\begin{aligned}
\nabla_\theta \log Z_\theta &= \nabla_\theta \log \int \exp\left(-E_\theta\left(x\right)\right) dx \\
&= \left(\int \exp\left(-E_\theta\left(x\right)\right) dx\right)^{-1} \nabla_\theta \int \exp\left(-E_\theta\left(x\right)\right) dx \\
&= \left(\int \exp\left(-E_\theta\left(x\right)\right) dx\right)^{-1} \int \nabla_\theta \exp\left(-E_\theta\left(x\right)\right) dx \\
&= \left(\int \exp\left(-E_\theta\left(x\right)\right) dx\right)^{-1} \int \exp\left(-E_\theta\left(x\right)\right) \left(-\nabla_\theta E_\theta\left(x\right)\right) dx \\
&= \int \left(\int \exp\left(-E_\theta\left(x\right)\right) dx\right)^{-1} \exp\left(-E_\theta\left(x\right)\right) \left(-\nabla_\theta E_\theta\left(x\right)\right) dx \\
&= \int \frac{\exp\left(-E_\theta\left(x\right)\right)}{Z_\theta} \left(-\nabla_\theta E_\theta\left(x\right)\right) dx \\
&= \int p_\theta\left(x\right) \left(-\nabla_\theta E_\theta\left(x\right)\right) dx \\
&= \mathbb{E}_{x \sim p_\theta(x)} \left[-\nabla_\theta E_\theta\left(x\right)\right] ,
\end{aligned} \tag{1.5}$$

这需要我们从 $p_\theta\left(x\right)$ 中进行采样，所以很多能量模型的研究都聚焦在如何从能量模型中采样，MCMC 采样法较为常用，其中最著名的是郎之万马尔可夫链蒙特卡洛（Langevin MCMC）采样法[1-2]，其首先从一个简单的先验分布中采样一个 x_0，之后模拟郎之万扩散过程（Langevin Diffusion Process），以 $\epsilon > 0$ 的步长迭代采样 K 步：

$$x_{k+1} = x_k + \frac{\epsilon^2}{2} \nabla_{x_k} \log p_\theta\left(x_k\right) + \epsilon z_k \quad k = 0, 1, \cdots, K-1 , \tag{1.6}$$

其中 $z_k \sim \mathcal{N}\left(0, 1\right)$，式中的 $\nabla_x \log p_\theta\left(x\right)$ 被称为 $p_\theta\left(x\right)$ 的分数（score，即概率分布的对数关于变量的梯度），它是容易计算的：

$$\nabla_x \log p_\theta\left(x\right) = -\nabla_x E_\theta\left(x\right) - \nabla_x \log Z_\theta = -\nabla_x E_\theta\left(x\right) , \tag{1.7}$$

其中 $\nabla_x E_\theta\left(x\right)$ 也可以由自动微分计算。当 $\epsilon \to 0$ 且 $K \to \infty$ 时，采样得到的 x_K 服从 $p_\theta\left(x\right)$ 的分布。最终的训练过程需要我们先从数据分布中采样一批数据，再从 $p_\theta\left(x\right)$ 中采样一批样本，用它们分别计算它们的 $\nabla_\theta E_\theta\left(x\right)$，从而计算出 θ 的梯度进行优化。然而，MCMC 采样过程的计算量较大，且每步训练都需要从更新过的能量模型中重新采样，所以实践中需要做一些近似，但这也会引入误差。

1.3 分数匹配法

另一种训练能量模型的方法称为分数匹配法，其优化 $p_\theta(x)$ 与 $p_{\text{data}}(x)$ 之间的 Fisher 散度（Fisher Divergence）：

$$D_F\left(p_{\text{data}}(x) \parallel p_\theta(x)\right) = \mathbb{E}_{p_{\text{data}}(x)}\left[\frac{1}{2}\|\nabla_x \log p_{\text{data}}(x) - \nabla_x \log p_\theta(x)\|^2\right], \tag{1.8}$$

即最小化 $p_\theta(x)$ 与 $p_{\text{data}}(x)$ 之间的分数。理论上，如果 $\nabla_x \log p_\theta(x) = \nabla_x \log p_{\text{data}}(x)$，那么根据式1.6采样出的样本就和数据同分布，上述极大似然法计算出来的 θ 的梯度就为0，所以该方法与极大似然法有相同的最优解。然而，虽然式1.8中的 $\nabla_x \log p_\theta(x) = -\nabla_x E_\theta(x)$ 很容易计算，但 $\nabla_x \log p_{\text{data}}(x)$ 却无法计算。Hyvarinen 等人[3]证明，在满足一定的条件时，Fisher 散度可以被重写为：

$$D_F\left(p_{\text{data}}(x) \parallel p_\theta(x)\right) = \mathbb{E}_{p_{\text{data}}(x)}\left[\frac{1}{2}\sum_{i=1}^{d}\left(\frac{\partial E_\theta(x)}{\partial x_i}\right)^2 + \frac{\partial^2 E_\theta(x)}{(\partial x_i)^2}\right] + \text{constant}, \tag{1.9}$$

其中 d 为 x 的维度。然而，该优化目标要求 $\log p_\theta(x)$ 满足很多条件，如处处有界且连续可导，现实中的数据分布很难满足该条件，如图像像素的值就是离散分布的。一个解决方案是给每个数据加少量的噪声，即 $\tilde{x} = x + \epsilon$，只要噪声分布是光滑的，带噪数据的分布 $q(\tilde{x}) = \int_x q(\tilde{x}|x)\, p_{\text{data}}(x)\, dx$ 就是光滑的，我们可以转而优化 $D_F\left(q(\tilde{x}) \parallel p_\theta(\tilde{x})\right)$。然而，该优化目标需要求二阶导数，尤其是当 d 较大时，其计算量是无法接受的。

为了克服这一困难，Vincent[4]在加噪思想的基础上，将其与去噪自编码器（Denoising Autoencoders）联系起来，提出了去噪分数匹配法（Denoising Score Matching，DSM），它将 Fisher 散度重写为：

$$\begin{aligned} D_F\left(q(\tilde{x}) \parallel p_\theta(\tilde{x})\right) &= \mathbb{E}_{q(\tilde{x})}\left[\frac{1}{2}\|\nabla_{\tilde{x}} \log q(\tilde{x}) - \nabla_{\tilde{x}} \log p_\theta(\tilde{x})\|^2\right] \\ &= \mathbb{E}_{q(x,\tilde{x})}\left[\frac{1}{2}\|\nabla_{\tilde{x}} \log q(\tilde{x}|x) - \nabla_{\tilde{x}} \log p_\theta(\tilde{x})\|^2\right] + \text{constant}. \end{aligned} \tag{1.10}$$

为了证明该公式，我们首先将公式右边展开：

$$\begin{aligned} &\mathbb{E}_{q(x,\tilde{x})}\left[\frac{1}{2}\|\nabla_{\tilde{x}} \log q(\tilde{x}|x) - \nabla_{\tilde{x}} \log p_\theta(\tilde{x})\|^2\right] \\ &= \mathbb{E}_{q(x,\tilde{x})}\left[\frac{1}{2}\|\nabla_{\tilde{x}} \log p_\theta(\tilde{x})\|^2\right] - \mathbb{E}_{q(x,\tilde{x})}\left[\left\langle \nabla_{\tilde{x}} \log p_\theta(\tilde{x}), \nabla_{\tilde{x}} \log q(\tilde{x}|x) \right\rangle\right] \\ &+ \mathbb{E}_{q(x,\tilde{x})}\left[\frac{1}{2}\|\nabla_{\tilde{x}} \log q(\tilde{x}|x)\|^2\right] \\ &= \mathbb{E}_{q(\tilde{x})}\left[\frac{1}{2}\|\nabla_{\tilde{x}} \log p_\theta(\tilde{x})\|^2\right] - \mathbb{E}_{q(x,\tilde{x})}\left[\left\langle \nabla_{\tilde{x}} \log p_\theta(\tilde{x}), \nabla_{\tilde{x}} \log q(\tilde{x}|x) \right\rangle\right] + C_1, \end{aligned} \tag{1.11}$$

其中 C_1 是与 θ 无关的常数。再将公式左边展开：

$$
\begin{aligned}
&\mathbb{E}_{q(\tilde{x})}\left[\frac{1}{2}\|\nabla_{\tilde{x}}\log q\left(\tilde{x}\right)-\nabla_{\tilde{x}}\log p_\theta\left(\tilde{x}\right)\|^2\right] \\
&=\mathbb{E}_{q(\tilde{x})}\left[\frac{1}{2}\|\nabla_{\tilde{x}}\log p_\theta\left(\tilde{x}\right)\|^2\right]-\mathbb{E}_{q(\tilde{x})}\left[\left\langle\nabla_{\tilde{x}}\log p_\theta\left(\tilde{x}\right),\nabla_{\tilde{x}}\log q\left(\tilde{x}\right)\right\rangle\right]+\mathbb{E}_{q(\tilde{x})}\left[\frac{1}{2}\|\nabla_{\tilde{x}}\log q\left(\tilde{x}\right)\|^2\right] \\
&=\mathbb{E}_{q(\tilde{x})}\left[\frac{1}{2}\|\nabla_{\tilde{x}}\log p_\theta\left(\tilde{x}\right)\|^2\right]-\int q\left(\tilde{x}\right)\left\langle\nabla_{\tilde{x}}\log p_\theta\left(\tilde{x}\right),\nabla_{\tilde{x}}\log q\left(\tilde{x}\right)\right\rangle d\tilde{x}+C_2 \\
&=\mathbb{E}_{q(\tilde{x})}\left[\frac{1}{2}\|\nabla_{\tilde{x}}\log p_\theta\left(\tilde{x}\right)\|^2\right]-\int q\left(\tilde{x}\right)\left\langle\nabla_{\tilde{x}}\log p_\theta\left(\tilde{x}\right),\frac{\nabla_{\tilde{x}}q\left(\tilde{x}\right)}{q\left(\tilde{x}\right)}\right\rangle d\tilde{x}+C_2 \\
&=\mathbb{E}_{q(\tilde{x})}\left[\frac{1}{2}\|\nabla_{\tilde{x}}\log p_\theta\left(\tilde{x}\right)\|^2\right]-\int\left\langle\nabla_{\tilde{x}}\log p_\theta\left(\tilde{x}\right),\nabla_{\tilde{x}}q\left(\tilde{x}\right)\right\rangle d\tilde{x}+C_2 \\
&=\mathbb{E}_{q(\tilde{x})}\left[\frac{1}{2}\|\nabla_{\tilde{x}}\log p_\theta\left(\tilde{x}\right)\|^2\right]-\int\left\langle\nabla_{\tilde{x}}\log p_\theta\left(\tilde{x}\right),\nabla_{\tilde{x}}\int q\left(x\right)q\left(\tilde{x}|x\right)dx\right\rangle d\tilde{x}+C_2 \\
&=\mathbb{E}_{q(\tilde{x})}\left[\frac{1}{2}\|\nabla_{\tilde{x}}\log p_\theta\left(\tilde{x}\right)\|^2\right]-\int\left\langle\nabla_{\tilde{x}}\log p_\theta\left(\tilde{x}\right),\int q\left(x\right)\nabla_{\tilde{x}}q\left(\tilde{x}|x\right)dx\right\rangle d\tilde{x}+C_2 \\
&=\mathbb{E}_{q(\tilde{x})}\left[\frac{1}{2}\|\nabla_{\tilde{x}}\log p_\theta\left(\tilde{x}\right)\|^2\right]-\int\left\langle\nabla_{\tilde{x}}\log p_\theta\left(\tilde{x}\right),\int q\left(x\right)q\left(\tilde{x}|x\right)\nabla_{\tilde{x}}\log q\left(\tilde{x}|x\right)dx\right\rangle d\tilde{x}+C_2 \\
&=\mathbb{E}_{q(\tilde{x})}\left[\frac{1}{2}\|\nabla_{\tilde{x}}\log p_\theta\left(\tilde{x}\right)\|^2\right]-\int\int q\left(x\right)q\left(\tilde{x}|x\right)\left\langle\nabla_{\tilde{x}}\log p_\theta\left(\tilde{x}\right),\nabla_{\tilde{x}}\log q\left(\tilde{x}|x\right)\right\rangle dxd\tilde{x}+C_2 \\
&=\mathbb{E}_{q(\tilde{x})}\left[\frac{1}{2}\|\nabla_{\tilde{x}}\log p_\theta\left(\tilde{x}\right)\|^2\right]-\int\int q\left(x,\tilde{x}\right)\left\langle\nabla_{\tilde{x}}\log p_\theta\left(\tilde{x}\right),\nabla_{\tilde{x}}\log q\left(\tilde{x}|x\right)\right\rangle dxd\tilde{x}+C_2 \\
&=\mathbb{E}_{q(\tilde{x})}\left[\frac{1}{2}\|\nabla_{\tilde{x}}\log p_\theta\left(\tilde{x}\right)\|^2\right]-\mathbb{E}_{q(x,\tilde{x})}\left[\left\langle\nabla_{\tilde{x}}\log p_\theta\left(\tilde{x}\right),\nabla_{\tilde{x}}\log q\left(\tilde{x}|x\right)\right\rangle\right]+C_2,
\end{aligned}
\tag{1.12}
$$

其中 C_2 是与 θ 无关的常数。等式两边只有常数项不同，所以等式成立。实践中，我们只给数据加少量的噪声以保证 $q\left(\tilde{x}\right)\approx p_{\text{data}}\left(x\right)$，如 $q\left(\tilde{x}|x\right)=\mathcal{N}\left(x,\sigma^2 I\right)$，其中 $\sigma\approx 0$，这样式 1.10变为：

$$
\begin{aligned}
D_F\left(q\left(\tilde{x}\right)\parallel p_\theta\left(\tilde{x}\right)\right)&=\mathbb{E}_{q(x,\tilde{x})}\left[\frac{1}{2}\|\nabla_{\tilde{x}}\log q\left(\tilde{x}|x\right)-\nabla_{\tilde{x}}\log p_\theta\left(\tilde{x}\right)\|^2\right] \\
&=\mathbb{E}_{p_{\text{data}}(x)}\mathbb{E}_{z\sim\mathcal{N}(0,\sigma^2 I)}\left[\frac{1}{2}\|\frac{z}{\sigma^2}+\nabla_{\tilde{x}}\log p_\theta\left(\tilde{x}\right)\|^2\right] \\
&=\mathbb{E}_{p_{\text{data}}(x)}\mathbb{E}_{z\sim\mathcal{N}(0,\sigma^2 I)}\left[\frac{1}{2}\|\frac{z}{\sigma^2}-\nabla_{\tilde{x}}E_\theta\left(\tilde{x}\right)\|^2\right],
\end{aligned}
\tag{1.13}
$$

其中 $\tilde{x}=x+z$。去噪分数匹配法使训练变得简洁优雅。然而，使用去噪分数匹配法训练得到的能量模型是对加噪数据分布 $q\left(\tilde{x}\right)$ 的拟合，而非真正的数据分布 $p_{\text{data}}\left(x\right)$，这也是该算法的一种取舍。

第 2 章 基于分数的生成模型

2.1 动机

值得注意的是，如果我们从生成模型的视角来看能量模型，如果使用分数匹配训练法，那么训练中只涉及 $\nabla_x \log p_\theta(x)$，如果使用 MCMC 采样法，那么生成也只涉及 $\nabla_x \log p_\theta(x)$，所以我们可以将 $\nabla_x \log p_\theta(x)$ 视为一个整体，使用一个分数模型 $s_\theta(x)$ 直接拟合它，从而脱离能量这个概念。Song 等人[5]基于此提出了基于分数的生成模型（Score-based Generative Models）。

从生成模型的视角来看，分数匹配法的一个问题在于，如果数据分布由两个几乎互不重叠的分布构成，即两个分布之间有大量低密度区域，那么分数匹配法无法估计这个两个分布之间的相对权重。考虑数据分布 $p_{\text{data}}(x) = \pi p_0(x) + (1-\pi) p_1(x)$，其中 $p_0(x)$ 和 $p_1(x)$ 是两个几乎互不重叠的分布，π 是权重，假设 $S_0 = \{x | p_0(x) > 0\}$ 和 $S_1 = \{x | p_1(x) > 0\}$，则数据分布的分数为：

$$\nabla_x \log p_{\text{data}}(x) = \begin{cases} \nabla_x \log p_0(x) & \text{if } x \in S_0 \\ \nabla_x \log p_1(x) & \text{if } x \in S_1, \end{cases} \tag{2.1}$$

其与权重 π 是无关的，因此学习得到的分数模型中并不包含 $p_0(x)$ 和 $p_1(x)$ 之间的权重信息，导致生成样本中属于 $p_0(x)$ 和 $p_1(x)$ 的样本的比例与 π 无关。现实中，很多数据分布都存在这种现象，尤其是高维数据，这限制了模型的应用。

为了解决这个问题，基于去噪分数匹配法，Song 等人[5]提出了 SMLD（Score Matching with Langevin Dynamics），即通过对数据进行不同级别的加噪并为每个级别的加噪分布学习一个分数模型的方法：当噪声级别较大时，互不重叠的分布会融合在一起，从而可以准确建模不同分布之间的权重；当噪声级别较小时，加噪分布接近于原分布，可以复原原分布，采样时按噪声级别从大到小的顺序依次采样。

2.2 训练与采样算法

我们先定义一个代表噪声级别的序列 $\{\sigma_i\}_{i=1}^L$，其满足 $\frac{\sigma_1}{\sigma_2} = \frac{\sigma_2}{\sigma_3} = \cdots = \frac{\sigma_{L-1}}{\sigma_L} > 1$，加噪后的数据分布为 $q_{\sigma_i}(\tilde{x}) = \int_x p_{\text{data}}(x) q_{\sigma_i}(\tilde{x}|x)\, dx$，其中 $q_{\sigma_i}(\tilde{x}|x) = \mathcal{N}(x, \sigma_i^2 I)$，$\sigma_1$ 要足

够大以让 $q_{\sigma_1}(\tilde{x})$ 尽可能融合，σ_L 要足够小以让 $q_{\sigma_L}(\tilde{x})$ 足够接近 $p_{\text{data}}(x)$。理论上，我们需要学习 L 个分数估计网络 $s_\theta^i(\tilde{x})$ 去分别拟合 $\nabla_{\tilde{x}} \log q_{\sigma_i}(\tilde{x})$，但这样会导致计算开销过大，我们可以只学习一个模型 $s_\theta(\tilde{x}, \sigma_i)$，此时模型需要识别不同的噪声级别 σ_i，所以我们将 σ_i 也作为模型的输入。根据式 1.13，其损失函数可以写为：

$$\mathcal{L}_i = \frac{1}{2}\mathbb{E}_{p_{\text{data}}(x)}\mathbb{E}_{z\sim\mathcal{N}(0,\sigma^2 I)}\left[\left\|s_\theta(x+z,\sigma_i) + \frac{z}{\sigma_i^2}\right\|^2\right]. \tag{2.2}$$

训练时，我们优化 $\frac{1}{L}\sum_{i=1}^{L}\lambda_i\mathcal{L}_i$，其中 λ_i 为 $\mathcal{L}_i$ 的权重。

与 Langevin MCMC 采样中只根据一个分数进行采样不同的是，我们现在有 L 个不同的噪声级别的分数，采样时，我们需要按噪声级别从大到小的顺序依次采样。我们使用退火郎之万马尔可夫链蒙特卡洛（Annealed Langevin MCMC）采样法，其将上一个噪声级别的 Langevin MCMC 采样结果作为下一个噪声级别的 Langevin MCMC 采样的初始化，并不断减小 Langevin MCMC 采样的步长。算法 2.1展示了具体算法。

算法 2.1 Annealed Langevin MCMC 采样算法

Prepare: well-trained score network s_θ.

Input: base step size s, Langevin MCMC sampling steps T

Run:

$\tilde{x}_0 \sim \mathcal{N}(0, \sigma_1^2 I)$

for $i = 1$ **to** L **do**

  $\epsilon_i = \sqrt{s}\cdot\frac{\sigma_i}{\sigma_L}$

  for $t = 0$ **to** $T-1$ **do**

    $z_t \sim \mathcal{N}(0, I)$

    $\tilde{x}_{t+1} = \tilde{x}_t + \frac{\epsilon_i^2}{2}s_\theta(\tilde{x}_t, \sigma_i) + \epsilon_i z_t$

  $\tilde{x}_0 = \tilde{x}_T$

return $\tilde{x}_0$

第 3 章 扩散模型

3.1 基本思想

对数据集的分布进行概率模型建模是机器学习的核心问题之一。然而现有数据集的分布都较为复杂，这使得概率模型的假设和参数拟合都较为困难，基于这一痛点，在非平衡态热力学的启发下，Jascha 等人[6]提出了扩散模型。非平衡热力学研究不处于热力学平衡中的物理系统，如热传导、物质的扩散等。

在扩散模型中，我们模拟非平衡热力学中的扩散过程，通过迭代的方法，有规律地、缓慢地摧毁原始数据的分布，最终将其转化一个已知的、简单的、容易采样的分布（通常为标准高斯分布），理论上任何复杂的原始分布都可以通过这一过程被转化为这种简单的分布。以图像为例，通过不断对图像的每个像素做扩散操作，最终这个图像就变会成了一张高斯噪声图。扩散模型学习这个扩散过程的逆向过程，学习完成后，任何从简单分布中采样的样本都可被这个逆向过程还原到原始分布中，即生成了一个服从原始分布的样本，这一过程就是生成模型进行生成的过程，这比直接显式地建模原始数据分布要容易很多。

该思想类似由粗到细的方法（coarse-to-fine），比如从低分辨率图像开始逐步生成更高分辨率的图像，但不同的是，扩散模型使用数学工具准确地描述出了由粗到细之间的状态以及它们之间是如何进行转换的。

3.2 马尔可夫性质

在扩散模型中，扩散过程被定义为一个马尔可夫过程。马尔可夫过程是一个具备了马尔可夫性质的随机过程。马尔可夫性质指随机过程在给定现在状态及所有过去状态的情况下，其未来状态的条件概率分布仅依赖于当前状态；换句话说，在给定现在状态时，它与过去状态（即该过程的历史路径）是条件独立的，那么此随机过程即具有马尔可夫性质。即：

$$Pr\left(X\left(t+h\right)=y|X\left(s\right)=x\left(s\right),s\leq t\right)=Pr\left(X\left(t+h\right)=y|X\left(t\right)=x\left(t\right)\right),\forall h>0\,, \tag{3.1}$$

其中，$X(t), t>0$ 为一个随机过程。

扩散模型要学习的是和扩散过程相同的路径，但方向不同。如果将扩散过程看为一个有向无环概率图（Directed Acyclic Graphs，DAG）模型，由于扩散过程的马尔可夫性质，这个有向无环概率图即为一条链（Chain/Path Graph），那么其逆向过程就是将概率图中的有向边调转方向，形成另一条链，因此其逆向过程也具有马尔可夫性质，所以扩散模型使用另一个可学习的马尔可夫过程拟合扩散过程的逆向过程。为了对应逆向过程的表述，我们之后将扩散过程称为前向过程。

3.3 同函数形式性质

对于连续时间的前向过程，Feller[7]证明了，在无穷小的时间间隔 dt 内，逆向过程的概率密度函数 $p(x_t|x_{t+dt})$ 与前向过程的概率密度函数 $p(x_{t+dt}|x_t)$ 具有相同的函数形式，但由于计算机只能建模离散的过程，为了利用这一性质，我们必须提高整个离散扩散过程的状态数量，使得相邻状态之间只做尽可能小的扰动，从而近似模拟连续的扩散过程。

3.4 前向过程与逆向过程

我们分别用 q 和 p 表示前向过程和逆向过程的概率分布，特别地，$q(x_0)$ 表示原始数据分布，$p(x_T)$ 代表我们想要的已知的、简单的、容易采样的分布。T 为扩散过程的总状态数量。我们的目标为在已知 q 的前提下通过机器学习的方法学习 p。

我们定义一步扩散为：

$$q(x_t|x_{t-1}) = T_\pi(x_t|x_{t-1};\beta_t) , \tag{3.2}$$

其中，$T_\pi(x_t|x_{t-1};\beta_t)$ 为马尔可夫扩散核，β_t 为扩散率。前向过程从数据分布 $q(x_0)$ 出发并进行 T 步扩散，由此，基于马尔可夫性质，前向过程的联合概率分布可以表示为：

$$\begin{aligned} q(x_{1\cdots T}|x_0) &= q(x_1|x_0)\, q(x_2|x_1,x_0)\cdots q(x_T|x_{T-1},x_{T-2},\cdots,x_0) \\ &= q(x_1|x_0)\, q(x_2|x_1)\cdots q(x_t|x_{t-1}) \\ &= \prod_{t=1}^{T} q(x_t|x_{t-1}) . \end{aligned} \tag{3.3}$$

逆向过程描述前向过程的逆，基于马尔可夫性质，逆向过程的联合概率分布可以表示为：

$$\begin{aligned} p\left(x_{0 \cdots T}\right) &= p\left(x_T\right) p\left(x_{T-1} | x_T\right) p\left(x_{T-2} | x_{T-1}, x_T\right) \cdots p\left(x_0 | x_1, x_2, \cdots, x_T\right) \\ &= p\left(x_T\right) p\left(x_{T-1} | x_T\right) p\left(x_{T-2} | x_{T-1}\right) \cdots p\left(x_0 | x_1\right) \\ &= p\left(x_T\right) \prod_{t=1}^{T} p\left(x_{t-1} | x_t\right) . \end{aligned} \tag{3.4}$$

这里我们使用 $p\left(x_{t-1} | x_t\right)$ 去拟合 $q\left(x_{t-1} | x_t\right)$，根据节 3.3，当 β_t 足够小时，$q\left(x_{t-1} | x_t\right)$ 和 $q\left(x_t | x_{t-1}\right)$ 具有相同的函数形式，故我们可以将 $p\left(x_{t-1} | x_t\right)$ 预设为和 $q\left(x_t | x_{t-1}\right)$ 相同的函数形式。注意，虽然我们的本意是使用 $p\left(x_{t-1} | x_t\right)$ 去拟合 $q\left(x_{t-1} | x_t\right)$，但在实践中，虽然 $q\left(x_t | x_{t-1}\right)$ 是已知的，但却无法求得 $q\left(x_{t-1} | x_t\right)$ 的闭式解，但如果引入 x_0 作为条件，就可以求得 $q\left(x_{t-1} | x_t, x_0\right)$ 的闭式解，并且这种引入并不影响公式推导，这在之后的推导中会展现出来。

3.5 优化目标推导

由 Chapman-Kolmogorov 公式，逆向过程生成的数据的概率分布为：

$$p\left(x_0\right) = \int d x_{1:T} p\left(x_{0:T}\right) . \tag{3.5}$$

由于我们希望对微观层面每一步的小扰动进行建模，所以我们需要引入前向过程并把公式分解成前向过程和逆向过程的每一步的组合：

$$\begin{aligned} p\left(x_0\right) &= \int d x_{1:T} p\left(x_{0:T}\right) \\ &= \int d x_{1:T} p\left(x_{0:T}\right) \frac{q\left(x_{1:T} | x_0\right)}{q\left(x_{1:T} | x_0\right)} \\ &= \int d x_{1:T} q\left(x_{1:T} | x_0\right) \frac{p\left(x_{0:T}\right)}{q\left(x_{1:T} | x_0\right)} \\ &= \int d x_{1:T} q\left(x_{1:T} | x_0\right) p\left(x_T\right) \prod_{t=1}^{T} \frac{p\left(x_{t-1} | x_t\right)}{q\left(x_t | x_{t-1}\right)} , \end{aligned} \tag{3.6}$$

这里最后一步的推导代入了式 3.3和式 3.4。

之后，我们希望极大化模型在数据分布 $q\left(x_0\right)$ 上的期望对数似然：

$$\begin{aligned} L &= \int d x_0 q\left(x_0\right) \log \left(p(x_0)\right) \\ &= \int d x_0 q\left(x_0\right) \log \left[\int d x_{1:T} q\left(x_{1:T} | x_0\right) p\left(x_T\right) \prod_{t=1}^{T} \frac{p\left(x_{t-1} | x_t\right)}{q\left(x_t | x_{t-1}\right)}\right] . \end{aligned} \tag{3.7}$$

使用 Jensen 不等式（概率论的版本），我们可以得到 L 的一个下界 K：

$$\begin{aligned} L &\geq \int dx_0 q\left(x_0\right) \int dx_{1:T} q\left(x_{1:T}|x_0\right) \log \left[p\left(x_T\right) \prod_{t=1}^{T} \frac{p\left(x_{t-1}|x_t\right)}{q\left(x_t|x_{t-1}\right)}\right] \\ &= \int dx_{0:T} q\left(x_{0:T}\right) \log \left[p\left(x_T\right) \prod_{t=1}^{T} \frac{p\left(x_{t-1}|x_t\right)}{q\left(x_t|x_{t-1}\right)}\right] \\ &= K\,. \end{aligned} \tag{3.8}$$

可以看到，上述推导将所有时间步 t 的前向过程和逆向过程联系了起来，体现在连乘符号后，但由于前向过程和逆向过程的随机变量不同，我们进一步使用贝叶斯公式改写 $q\left(x_t|x_{t-1}\right)$：

$$\begin{aligned} K &= \int dx_{0:T} q\left(x_{0:T}\right) \left[\log p\left(x_T\right) + \sum_{t>1} \log \frac{p\left(x_{t-1}|x_t\right)}{q\left(x_t|x_{t-1}\right)} + \log \frac{p\left(x_0|x_1\right)}{q\left(x_1|x_0\right)}\right] \\ &= \int dx_{0:T} q\left(x_{0:T}\right) \left[\log p\left(x_T\right) + \sum_{t>1} \log \frac{p\left(x_{t-1}|x_t\right)}{q\left(x_t|x_{t-1}, x_0\right)} + \log \frac{p\left(x_0|x_1\right)}{q\left(x_1|x_0\right)}\right] \\ &= \int dx_{0:T} q\left(x_{0:T}\right) \left[\log p\left(x_T\right) + \sum_{t>1} \log \frac{p\left(x_{t-1}|x_t\right)}{q\left(x_{t-1}|x_t, x_0\right)} \cdot \frac{q\left(x_{t-1}|x_0\right)}{q\left(x_t|x_0\right)} + \log \frac{p\left(x_0|x_1\right)}{q\left(x_1|x_0\right)}\right] \\ &= \int dx_{0:T} q\left(x_{0:T}\right) \left[\log \frac{p\left(x_T\right)}{q\left(x_T|x_0\right)} + \sum_{t>1} \log \frac{p\left(x_{t-1}|x_t\right)}{q\left(x_{t-1}|x_t, x_0\right)} + \log p\left(x_0|x_1\right)\right]\,, \end{aligned} \tag{3.9}$$

其中第二行的推导用到了节 3.2的马尔可夫性质，即 $q\left(x_t|x_{t-1}\right) = q\left(x_t|x_{t-1}, x_0\right)$，第三行的推导用到了贝叶斯公式，即 $q\left(x_t|x_{t-1}, x_0\right) = q\left(x_{t-1}|x_t, x_0\right) \frac{q(x_t|x_0)}{q(x_{t-1}|x_0)}$。

我们分析 K 的中括号内的每一项，以帮助我们理解它的含义。具体来说，第一项为：

$$\begin{aligned} &\int dx_{0:T} q\left(x_{0:T}\right) \log \frac{p\left(x_T\right)}{q\left(x_T|x_0\right)} \\ =& \int dx_{0:T} q\left(x_0\right) q\left(x_T|x_0\right) q\left(x_{1:T-1}|x_0, x_T\right) \log \frac{p\left(x_T\right)}{q\left(x_T|x_0\right)} \\ =& \int dx_0 dx_T q\left(x_0\right) q\left(x_T|x_0\right) \log \frac{p\left(x_T\right)}{q\left(x_T|x_0\right)} \int dx_{1:T-1} q\left(x_{1:T-1}|x_0, x_T\right) \\ =& \int dx_0 dx_T q\left(x_0\right) q\left(x_T|x_0\right) \log \frac{p\left(x_T\right)}{q\left(x_T|x_0\right)} \\ =& \int dx_0 q\left(x_0\right) \int dx_T q\left(x_T|x_0\right) \log \frac{p\left(x_T\right)}{q\left(x_T|x_0\right)} \\ =& \int dx_0 q\left(x_0\right) \left[-D_{KL}\left(q\left(x_T|x_0\right) \parallel p\left(x_T\right)\right)\right]\,. \end{aligned} \tag{3.10}$$

最后一项为：

$$\begin{aligned}
&\int dx_{0:T} q\left(x_{0:T}\right) \log p\left(x_0 | x_1\right) \\
=&\int dx_{0:T} q\left(x_0\right) q\left(x_{1:T} | x_0\right) \log p\left(x_0 | x_1\right) \\
=&\int dx_0 q\left(x_0\right) \log p\left(x_0 | x_1\right) \int dx_{1:T} q\left(x_{1:T} | x_0\right) \\
=&\int dx_0 q\left(x_0\right) \log p\left(x_0 | x_1\right) .
\end{aligned} \tag{3.11}$$

中间的连加项为：

$$\begin{aligned}
&\int dx_{0:T} q\left(x_{0:T}\right) \log \frac{p\left(x_{t-1} | x_t\right)}{q\left(x_t | x_{t-1}, x_0\right)} \\
=&\int dx_{0:T} q\left(x_{1:t-2, t+1:T} | x_0, x_{t-1}, x_t\right) q\left(x_t, x_0\right) q\left(x_{t-1} | x_t, x_0\right) \log \frac{p\left(x_{t-1} | x_t\right)}{q\left(x_{t-1} | x_t, x_0\right)} \\
=&\int dx_0 dx_{t-1} dx_t q\left(x_t, x_0\right) q\left(x_{t-1} | x_t, x_0\right) \log \frac{p\left(x_{t-1} | x_t\right)}{q\left(x_{t-1} | x_t, x_0\right)} \\
&\int dx_{1:t-2, t+1:T} q\left(x_{1:t-2, t+1:T} | x_0, x_{t-1}, x_t\right) \\
=&\int dx_0 dx_{t-1} dx_t q\left(x_t, x_0\right) q\left(x_{t-1} | x_t, x_0\right) \log \frac{p\left(x_{t-1} | x_t\right)}{q\left(x_{t-1} | x_t, x_0\right)} \\
=&\int dx_0 dx_t q\left(x_t, x_0\right) \int dx_{t-1} q\left(x_{t-1} | x_t, x_0\right) \log \frac{p\left(x_{t-1} | x_t\right)}{q\left(x_{t-1} | x_t, x_0\right)} \\
=&\int dx_0 dx_t q\left(x_0, x_t\right)\left[-D_{KL}\left(q\left(x_{t-1} | x_t, x_0\right) \| p\left(x_{t-1} | x_t\right)\right)\right] .
\end{aligned} \tag{3.12}$$

最终结果为：

$$\begin{aligned}
K=&\int dx_0 q\left(x_0\right)\left[-D_{KL}\left(q\left(x_T | x_0\right) \| p\left(x_T\right)\right)\right] \\
&+\sum_{t>1} \int dx_0 dx_t q\left(x_0, x_t\right)\left[-D_{KL}\left(q\left(x_{t-1} | x_t, x_0\right) \| p\left(x_{t-1} | x_t\right)\right)\right] \\
&+\int dx_0 q\left(x_0\right)\left[\log p\left(x_0 | x_1\right)\right] \\
=&\mathbb{E}_q\left[-D_{KL}\left(q\left(x_T | x_0\right) \| p\left(x_T\right)\right)+\sum_{t>1}-D_{KL}\left(q\left(x_{t-1} | x_t, x_0\right) \| p\left(x_{t-1} | x_t\right)\right)+\log p\left(x_0 | x_1\right)\right] ,
\end{aligned} \tag{3.13}$$

其中，D_{KL} 为 KL-散度。注意，我们的目标是极大化对数似然的下界 K，为了方便，我们取其相反数为损失函数 $\mathcal{L}$：

$$\mathcal{L}=\mathbb{E}_q\left[\underbrace{-\log p\left(x_0 | x_1\right)}_{\mathcal{L}_0}+\underbrace{D_{KL}\left(q\left(x_T | x_0\right) \| p\left(x_T\right)\right)}_{\mathcal{L}_T}+\sum_{t>1} \underbrace{D_{KL}\left(q\left(x_{t-1} | x_t, x_0\right) \| p\left(x_{t-1} | x_t\right)\right)}_{\mathcal{L}_{t-1}}\right] . \tag{3.14}$$

直观上，$\mathcal{L}_0$ 使得逆向过程的终点与前向过程的起点一致，$\mathcal{L}_T$ 使得逆向过程的起点与前向过程的终点一致，$\mathcal{L}_{t-1}$ 使得整个路径中，逆向过程的每一步与前向过程的每一步的逆一致。该公式即为扩散模型的最终优化目标。

3.6 从变分自编码器视角审视扩散模型

在概率模型中，我们通常需要对某个后验分布 $p(z|x)$ 进行估计，其中 x 为已知观测变量，通常为数据，z 是未知变量。根据贝叶斯公式 $p(z|x) = \frac{p(x|z)p(z)}{p(x)} = \frac{p(x|z)p(z)}{\int p(x,z')dz'}$，类似于能量模型，分母中的积分通常无法处理。一种常用的方法为变分推断法（Variational Inference），即使用一个可学习的变分分布 $q(z)$ 拟合后验分布 $p(z|x)$。为了做到这一点，我们优化两个分布之间的 KL-散度，所以有如下推导：

$$\begin{aligned} D_{KL}(q(z) \parallel p(z|x)) &= \int q(z)\left[\log \frac{q(z)}{p(z|x)}\right] dz \\ &= \int q(z)\left[\log \frac{q(z)}{\frac{p(x,z)}{p(x)}}\right] dz \\ &= \int q(z)\left[\log \frac{q(z)}{p(x,z)}\right] dz + \int q(z)\left[\log p(x)\right] dz \\ &= D_{KL}(q(z) \parallel p(x,z)) + \log p(x) \ , \\ \log p(x) &= D_{KL}(q(z) \parallel p(z|x)) - D_{KL}(q(z) \parallel p(x,z)) \ . \end{aligned} \tag{3.15}$$

$\log p(x)$ 为对数似然（又称证据），因为我们是优化 $q(z)$ 去拟合 $p(z|x)$ 的，所以 $\log p(x)$ 相当于 q 是固定的，所以对于等式右边来说，极小化 $D_{KL}(q(z) \parallel p(z|x))$ 等价于极大化 $-D_{KL}(q(z) \parallel p(x,z))$，通过对 $q(z)$ 合理的假设和选择，$-D_{KL}(q(z) \parallel p(x,z))$ 是可以计算并且优化的，而 $-D_{KL}(q(z) \parallel p(x,z))$ 又被称为证据下界（Evidence Lower Bound，ELBO），其中证据（evidence）指对数似然 $\log p(x)$。广义上，任何通过优化的方法来拟合目标分布的方法统称为变分推断。

变分自编码器（VAEs）[8]是一类重要的生成模型，其假设我们观测到的数据都是由一个未知的隐变量 z 生成的，它们的联合分布为 $p(x,z)$。从对数似然（证据）出发，我

们依然可以推导出 ELBO：

$$\begin{aligned}\log p(x) &= \log \int p(x,z)\,dz \\ &= \log \int \frac{p(x,z)\,q_\phi(z|x)}{q_\phi(z|x)} dz \\ &= \log \mathbb{E}_{q_\phi(z|x)}\left[\frac{p(x,z)}{q_\phi(z|x)}\right] \\ &\geq \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p(x,z)}{q_\phi(z|x)}\right] = \text{ELBO}\,,\end{aligned} \tag{3.16}$$

其中 $q_\phi(z|x)$ 即为变分推断中的 $q(z)$，是一个参数为 ϕ 的需要优化的变分分布，用来拟合 $p(z|x)$。为了进一步了解 ELBO 的本质，我们提供另一种推导方法：

$$\begin{aligned}\log p(x) &= \log p(x) \int q_\phi(z|x)\,dz \\ &= \int q_\phi(z|x) \log p(x)\,dz \\ &= \mathbb{E}_{q_\phi(z|x)}\left[\log p(x)\right] \\ &= \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p(x,z)}{p(z|x)}\right] \\ &= \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p(x,z)\,q_\phi(z|x)}{p(z|x)\,q_\phi(z|x)}\right] \\ &= \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p(x,z)}{q_\phi(z|x)}\right] + \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{q_\phi(z|x)}{p(z|x)}\right] \\ &= \text{ELBO} + D_{KL}\left(q_\phi(z|x) \parallel p(z|x)\right) \geq \text{ELBO}\,.\end{aligned} \tag{3.17}$$

该推导结论与式 3.15等价，其表明证据等于 ELBO 加上 $q_\phi(z|x)$ 与 $p(z|x)$ 之间的 KL-散度，由于 $D_{KL}\left(q_\phi(z|x) \parallel p(z|x)\right) \geq 0$，所以 ELBO 是证据的下界。由于等式左边的证据相对于 q_ϕ 是常数，所以通过优化 ϕ 来极大化 ELBO 等价于极小化 $D_{KL}\left(q_\phi(z|x) \parallel p(z|x)\right)$，等价于在做变分推断。

在 VAEs 中，为了优化 ELBO，我们还需要对 $p(x,z)$ 进行计算，所以我们将 ELBO 写为：

$$\begin{aligned}\text{ELBO} &= \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p(x,z)}{q_\phi(z|x)}\right] \\ &= \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p_\theta(x|z)\,p(z)}{q_\phi(z|x)}\right] \\ &= \mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x|z)\right] + \mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p(z)}{q_\phi(z|x)}\right] \\ &= \mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x|z)\right] - D_{KL}\left(q_\phi(z|x) \parallel p(z)\right)\,,\end{aligned} \tag{3.18}$$

其中 $p_\theta(x|z)$ 是一个参数为 θ 的解码器，其将任一给定的隐变量 z 转化为对应的数据 x，$p(z)$ 是隐变量 z 的先验分布，一般假设为标准高斯分布。训练时，我们联合训练编码器 $q_\phi(z|x)$ 和解码器 $p_\theta(x|z)$，对于编码器，需要将数据 x 编码为 z，并且符合 z 的先验分布 $p(z)$，对应 $-D_{KL}(q_\phi(z|x) \parallel p(z))$；对于解码器，需要根据编码器的编码结果 z 复原数据 x，对应 $\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)]$。本质上，VAEs 是一个对隐变量空间分布进行限制的自编码器，这种限制的目的是为了使我们摆脱编码器，从而可以从 z 的先验分布 $p(z)$ 中采样并直接使用解码器生成数据，这就是生成模型。

实践中，由于对 $p(z)$ 的假设过于简单，这限制了 VAEs 的建模能力，为此我们可以将其设计为马尔可夫链式的多层 VAEs：第一层编码器对 x 进行编码得到 z_1，解码器根据 z_1 重构 x；第二层编码器对 z_1 进行编码得到 z_2，解码器根据 z_2 重构 z_1，重复 T 次，生成时，我们从 $p(z_T)$ 中采样，不断解码直到生成 z_1，并根据 z_1 生成最后的数据，这种模型称为马尔可夫链式的多层 VAEs（Markovian Hierarchical VAEs，MHVAEs）。公式中我们只需要使用 $z_{1:T}$ 代替 z，即使用变分分布 $q_\phi(z_{1:T}|x)$ 去拟合后验分布 $p_\theta(z_{1:T}|x)$，此时 ELBO 变为 $\mathbb{E}_{q_\phi(z_{1:T}|x)}\left[\log \frac{p_\theta(z_{0:T})}{q_\phi(z_{1:T}|x)}\right]$。可以看到，该公式与扩散模型推导过程中的式 3.6和式 3.8完全一致，这表明扩散模型与 MHVAEs 在本质上是一样的，但有三处区别：

- MHVAEs 中隐变量通常是低维的，而扩散模型中的隐变量维度与数据维度相同；
- MHVAEs 中每一层有不同的网络参数，而扩散模型所有层共享同一网络参数；
- MHVAEs 中虽然对隐变量的先验分布有假设，但隐变量概率分布还是通过编码器预测得出的，而扩散模型中的隐变量概率分布是预定义好的，所以扩散模型不需要学习编码器，只需要学习解码器。这同时也简化了扩散模型的训练过程，因为我们可以直接采样得到隐变量。

第 4 章 去噪扩散概率模型

如前所述，扩散模型是一类概率模型的总称，即使扩散模型都有类似的优化目标，但在实践中其实现方式多种多样，其中去噪扩散概率模型（Denoising Diffusion Probabilistic Models，DDPMs）[9]是效果最佳、应用最广的模型，本文也基于该实现进行科研探索，下面我们介绍 DDPMs 的具体实现方式。

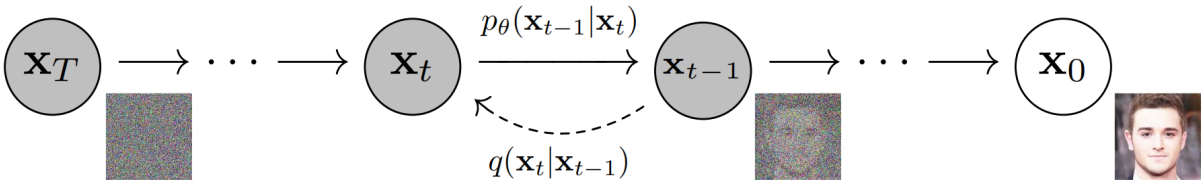

图 4.1 DDPMs 的概率图表示。

4.1 定义

图 4.1展示了 DDPMs 的概率图表示。在扩散模型公式推导过程中，我们假设前向过程的概率分布 q 是已知的，即需要对式 3.2中的扩散核给出明确定义，DDPMs 将其定义为一个高斯分布：

$$q\left(x_t|x_{t-1}\right)=\mathcal{N}\left(\sqrt{1-\beta_t}x_{t-1}\,,\,\beta_t I\right)\,, \tag{4.1}$$

其中 $\{\beta_t\}_{t=1}^T$ 是一个预定义的固定的方差序列，其值是在 0.0001 到 0.02 之间等间距的 T 个数，T 为总扩散步数，其值为 1000。可以看到，该定义假设图像不同像素之间是独立的。

值得注意的是，前向过程是从 x_0 开始的，如果我们想要采样某个时间步 t 的 x_t，我们需要不断重复上述扩散操作 t 次，这将消耗极大的计算资源和时间。幸运的是，我们可以根据高斯分布的运算法则直接推导出 $q\left(x_t|x_0\right)$，这样只需要一步就可以采样出任何时间步 t 的 x_t。具体来说，对于某个 $x_0\sim q\left(x_0\right)$，假设我们想采样一个 x_2，我们需要先从 $q\left(x_1|x_0\right)$ 中采样一个 x_1，由式 4.1可得，$x_1=\sqrt{1-\beta_1}x_0+\sqrt{\beta_1}\epsilon_1$，其中 $\epsilon_1\sim\mathcal{N}\left(0,I\right)$；之后，我们根据 $q\left(x_2|x_1\right)$ 采样一个 $x_2=\sqrt{1-\beta_2}x_1+\sqrt{\beta_2}\epsilon_2$，其中 $\epsilon_2\sim\mathcal{N}\left(0,I\right)$，它与 ϵ_1 是独立的。我们将 x_1 的表达式代入可得 $x_2=\sqrt{\left(1-\beta_1\right)\left(1-\beta_2\right)}x_0+\sqrt{\left(1-\beta_2\right)\beta_1}\epsilon_1+\sqrt{\beta_2}\epsilon_2$，根据两个相互独立的高斯分布相加法则，$q\left(x_2|x_0\right)=\mathcal{N}\left(\sqrt{\left(1-\beta_1\right)\left(1-\beta_2\right)}x_0,\left[1-\left(1-\beta_1\right)\left(1-\beta_2\right)\right]I\right)$，

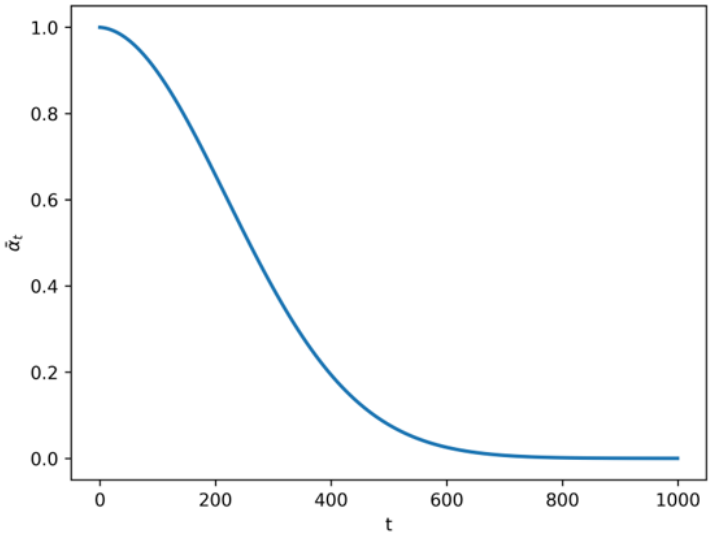

图 4.2 $\bar{\alpha}_t$ 的值与 t 的关系。

为了方便书写，我们定义 $\alpha_t = 1-\beta_t$，$\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$，则 $q\left(x_2|x_0\right) = \mathcal{N}\left(\sqrt{\bar{\alpha}_2}x_0, \left(1-\bar{\alpha}_2\right)I\right)$。更一般地，我们可以推导得到 $q\left(x_t|x_0\right) = \mathcal{N}\left(\sqrt{\bar{\alpha}_t}x_0, \left(1-\bar{\alpha}_t\right)I\right)$。注意到 β_t 都是接近于 0 且大于 0 的数，所以 α_t 都是接近于 1 且小于 0 的数，如图 4.2所示，当 t 趋近于 0 时，$\bar{\alpha}_t$ 趋近于 1，当 t 趋近于 T 时，$\bar{\alpha}_t$ 趋近于 0，所以前向过程的起点趋近于数据分布，终点 x_T 的分布趋近于标准高斯分布 $\mathcal{N}\left(0, I\right)$。直观上，前向过程从数据分布出发，不断对数据进行缩放和加噪，直到将其转化为一个标准高斯分布。

4.2 训练与采样算法

根据节 3.3的同函数形式性质，DDPMs 将要学习的逆向过程 p 定义为与 q 相同的高斯分布形式：

$$p_\theta\left(x_{t-1}|x_t\right) = \mathcal{N}\left(\mu_\theta\left(x_t, t\right), \Sigma_\theta\left(x_t, t\right)\right), \tag{4.2}$$

其中 μ_θ 和 Σ_θ 是以 θ 为参数的深度神经网络模型，根据逆向过程的当前状态 x_t 预测下一状态 x_{t-1} 的均值和方差。与 SMLD 类似，理论上，我们需要为逆向过程的每一步单独训练一个神经网络，但这样会导致计算开销过大，于是 DDPMs 让所有时间步共享同一个模型，此时模型需要识别不同时间步的 x_t，所以我们将 t 也作为模型的输入以指明 x_t 的时间步。逆向过程的起点被设定为标准高斯分布，即 $p\left(x_T\right) = \mathcal{N}\left(0, I\right)$。

由于逆向过程拟合的目标是前向过程的逆，根据式 3.14，我们需要求得前向过程的后验分布 $q\left(x_{t-1}|x_t, x_0\right)$，我们将 $q\left(x_t|x_{t-1}, x_0\right) = q\left(x_t|x_{t-1}\right) = \mathcal{N}\left(\sqrt{1-\beta_t}x_{t-1}, \beta_t I\right)$，

$q\left(x_{t-1}|x_0\right)=\mathcal{N}\left(\sqrt{\bar{\alpha}_{t-1}}x_0,\left(1-\bar{\alpha}_{t-1}\right)I\right)$ 以及 $q\left(x_t|x_0\right)=\mathcal{N}\left(\sqrt{\bar{\alpha}_t}x_0,\left(1-\bar{\alpha}_t\right)I\right)$ 的概率密度函数代入贝叶斯公式 $q\left(x_{t-1}|x_t,x_0\right)=q\left(x_t|x_{t-1},x_0\right)\frac{q\left(x_{t-1}|x_0\right)}{q\left(x_t|x_0\right)}$，可以得到：

$$\begin{aligned}q\left(x_{t-1}|x_t,x_0\right)&=q\left(x_t|x_{t-1},x_0\right)\frac{q\left(x_{t-1}|x_0\right)}{q\left(x_t|x_0\right)}\\&=\frac{\frac{1}{\sqrt{2\pi\beta_t}}\exp\left[-\frac{\left(x_t-\sqrt{\alpha_t}x_{t-1}\right)^2}{2\beta_t}\right]\cdot\frac{1}{\sqrt{2\pi(1-\bar{\alpha}_{t-1})}}\exp\left[-\frac{\left(x_{t-1}-\sqrt{\bar{\alpha}_{t-1}}x_0\right)^2}{2(1-\bar{\alpha}_{t-1})}\right]}{\frac{1}{\sqrt{2\pi(1-\bar{\alpha}_t)}}\exp\left[-\frac{\left(x_t-\sqrt{\bar{\alpha}_t}x_0\right)^2}{2(1-\bar{\alpha}_t)}\right]}\\&=\frac{1}{\sqrt{2\pi}\sqrt{\frac{\beta_t(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}}}\exp\left[\frac{-E}{2\frac{\beta_t(1-\bar{\alpha}_{t-1})}{(1-\bar{\alpha}_t)}}\right],\end{aligned}\tag{4.3}$$

其中，

$$\begin{aligned}E=&\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\left(x_t^2+\alpha_t x_{t-1}^2-2\sqrt{\alpha_t}x_{t-1}x_t\right)\\&+\frac{\beta_t}{1-\bar{\alpha}_t}\left(x_{t-1}^2+\bar{\alpha}_{t-1}x_0^2-2\sqrt{\bar{\alpha}_{t-1}}x_0x_{t-1}\right)\\&-\frac{\beta_t\left(1-\bar{\alpha}_{t-1}\right)}{\left(1-\bar{\alpha}_t\right)^2}\left(x_t^2+\bar{\alpha}_t x_0^2-2\sqrt{\bar{\alpha}_t}x_0x_t\right)\\=&\left(x_{t-1}-\frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0-\frac{\sqrt{\alpha_t}\left(1-\bar{\alpha}_{t-1}\right)}{1-\bar{\alpha}_t}x_t\right)^2.\end{aligned}\tag{4.4}$$

所以，

$$\begin{aligned}q\left(x_{t-1}|x_t,x_0\right)&=\mathcal{N}\left(\tilde{\mu}_t\left(x_t,x_0\right),\tilde{\beta}_t I\right)\\\tilde{\mu}_t\left(x_t,x_0\right)&=\frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0+\frac{\sqrt{\alpha_t}\left(1-\bar{\alpha}_{t-1}\right)}{1-\bar{\alpha}_t}x_t\,,\;\tilde{\beta}_t=\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t\,.\end{aligned}\tag{4.5}$$

最后，我们将式 4.2和式 4.5代入式 3.14计算最终的优化目标。

对于 $\mathcal{L}_T$，由于 $p\left(x_T\right)=\mathcal{N}\left(0,I\right)$ 中不含可学习的参数，所以此项可以忽略。直观上，前向过程将任何分布转化为 $q\left(x_T|x_0\right)=\mathcal{N}\left(\sqrt{\bar{\alpha}_t}x_0,\left(1-\bar{\alpha}_t\right)I\right)$，其本身已经与 $\mathcal{N}\left(0,I\right)$ 足够接近，不需要任何优化。

对于 $\mathcal{L}_{t-1}$，我们需要计算两个多元高斯分布之间的 KL-散度。对于两个多元高斯分布 $u\sim\mathcal{N}\left(\mu_1,\Sigma_1\right)$ 和 $v\sim\mathcal{N}\left(\mu_2,\Sigma_2\right)$，它们之间的 KL-散度为：

$$D_{KL}\left(u\parallel v\right)=\frac{1}{2}\left\{log\frac{|\Sigma_2|}{|\Sigma_1|}+tr\left(\Sigma_2^{-1}\Sigma_1\right)+\left(\mu_2-\mu_1\right)^T\Sigma_2^{-1}\left(\mu_2-\mu_1\right)-n\right\},\tag{4.6}$$

其中 n 为高斯分布的维度，$tr\left(\cdot\right)$ 为矩阵的迹。当 $\mu_1=\mu_2$ 且 $\Sigma_1=\Sigma_2$ 时，$D_{KL}\left(u\parallel v\right)=0$ 最小。注意到式 4.5中前向过程的后验分布的协方差为已知的常数，所以其对应的逆向过程不再需要使用神经网络去拟合该协方差，我们直接将式 4.2中的 $\Sigma_\theta\left(x_t,t\right)$ 设为

该常数，即 $\Sigma_\theta(x_t,t)=\sigma_t^2 I=\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t I$，这样可以降低模型复杂度和计算量。分别将式 4.5和式 4.2代入式 4.6的 u 和 v，得到 $\mathcal{L}_{t-1}=\mathbb{E}_q\left[\frac{1}{2\sigma_t^2}\|\tilde{\mu}_t(x_t,x_0)-\mu_\theta(x_t,t)\|^2\right]$。直观上，我们需要训练一个神经网络 $\mu_\theta(x_t,t)$ 去拟合 $\tilde{\mu}_t(x_t,x_0)$，然而，在深度学习领域，很多在理论上等价的学习目标，其学习难度和效果却大相径庭。在 DDPMs 中，作者发现这种预测均值的方法效果欠佳，经过实验，作者发现使用预测噪声的方法效果最佳。具体来说，在预测目标 $\tilde{\mu}_t(x_t,x_0)=\frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0+\frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t$ 中，x_t 是已知的，因为它是神经网络的输入，只有 x_0 是未知的，由于 $x_t=\sqrt{\bar{\alpha}_t}x_0+\sqrt{1-\bar{\alpha}_t}\epsilon$，则 $x_0=\frac{x_t-\sqrt{1-\bar{\alpha}_t}\epsilon}{\sqrt{\bar{\alpha}_t}}$，其中只有 ϵ 是未知的，因为我们转而训练一个神经网络 $\epsilon_\theta(x_t,t)$ 去拟合 ϵ，此时损失函数变为 $\mathcal{L}_{t-1}=\mathbb{E}_q\left[\frac{\beta_t^2}{2\sigma_t^2\alpha_t(1-\bar{\alpha}_t)}\|\epsilon-\epsilon_\theta(x_t,t)\|^2\right]$，这种方法称为重参数化（reparameterization）[8]。逆向过程 $p_\theta(x_{t-1}|x_t)$ 预测的均值也变为 $\mu_\theta(x_t,t)=\frac{1}{\sqrt{\alpha_t}}\left[x_t-\frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t,t)\right]$。

对于 $\mathcal{L}_0$，它优化逆向过程的最后一步 $p_\theta(x_0|x_1)=\mathcal{N}(\mu_\theta(x_1,1),\sigma_1^2 I)$，其负责生成最后的图像。假设像素值为 $\{0,1,\cdots,255\}$ 的图像的像素值都被线性缩放到 $[-1,1]$，这么做是为了与逆向过程的起点 $p(x_T)=\mathcal{N}(0,I)$ 的样本的数值量级保持一致。由于图像的像素值是离散的，为了计算其离散对数似然 $\mathcal{L}_0=\mathbb{E}_q[-\log p_\theta(x_0|x_1)]$，我们将 $p_\theta(x_0|x_1)$ 写为：

$$
\begin{aligned}
p_\theta(x_0|x_1)&=\prod_{i=1}^{D}\int_{\delta_-\left(x_0^i\right)}^{\delta_+\left(x_0^i\right)}\mathcal{N}\left(\mu_\theta^i(x_1,1),\sigma_1^2\right)dx\\
\delta_+(x)&=\begin{cases}\infty & \text{if } x=1\\ x+\frac{1}{255} & \text{if } x<1\end{cases}\qquad \delta_-(x)=\begin{cases}-\infty & \text{if } x=-1\\ x-\frac{1}{255} & \text{if } x>-1\end{cases},
\end{aligned}
\tag{4.7}
$$

其中 D 为图像像素总数，i 为像素坐标。直观上，我们认为逆向过程的最后一步为每个像素预测了一个高斯分布，我们优化这个高斯分布使得原始图像中对应的像素在该高斯分布中的概率尽可能大。在计算时，我们近似地把这个积分转化为一个黎曼求和，即求高斯概率密度函数曲线下指定区间的长方形的面积，忽略 $x=1$ 和 $x=-1$ 时的边缘效

应：

$$\begin{aligned}\mathcal{L}_0 &= \mathbb{E}_q\left[-\log p_\theta\left(x_0|x_1\right)\right] \\ &= \mathbb{E}_q\left[-\sum_{i=1}^{D}\log\int_{\delta_-\left(x_0^i\right)}^{\delta_+\left(x_0^i\right)}\frac{1}{\sigma_1\sqrt{2\pi}}\exp\left(-\frac{\left(x-\mu_\theta^i\left(x_1,1\right)\right)^2}{2\sigma_1^2}\right)dx\right] \\ &\approx \mathbb{E}_q\left[-\sum_{i=1}^{D}\log\left[\frac{2}{255}\frac{1}{\sigma_1\sqrt{2\pi}}\exp\left(-\frac{\left(x_0^i-\mu_\theta^i\left(x_1,1\right)\right)^2}{2\sigma_1^2}\right)\right]\right] \\ &= \mathbb{E}_q\left[\sum_{i=1}^{D}\frac{\left(x_0^i-\mu_\theta^i\left(x_1,1\right)\right)^2}{2\sigma_1^2}\right]+\text{constant}\,.\end{aligned} \tag{4.8}$$

所以，我们希望 $\mu_\theta\left(x_1,1\right)$ 是对 x_0 的直接估计，因为 $x_1=\sqrt{\bar{\alpha}_1}x_0+\sqrt{1-\bar{\alpha}_1}\epsilon$，所以 $\mu_\theta\left(x_1,1\right)$ 应该直接估计 $\frac{x_1-\sqrt{1-\bar{\alpha}_1}\epsilon}{\sqrt{\bar{\alpha}_1}}$，如果选择之前的去噪形式的参数方法，则 $\mu_\theta\left(x_1,1\right)=\frac{x_1-\sqrt{1-\bar{\alpha}_1}\epsilon_\theta(x_1,1)}{\sqrt{\bar{\alpha}_1}}$，其刚好符合 $\mathcal{L}_{t-1}$ 中的公式 $\mu_\theta\left(x_t,t\right)=\frac{1}{\sqrt{\alpha_t}}\left[x_t-\frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta\left(x_t,t\right)\right]$，即 $\mu_\theta\left(x_1,1\right)=\frac{1}{\sqrt{\alpha_1}}\left[x_1-\frac{\beta_1}{\sqrt{1-\bar{\alpha}_1}}\epsilon_\theta\left(x_1,1\right)\right]=\frac{x_1-\sqrt{1-\bar{\alpha}_1}\epsilon_\theta(x_1,1)}{\sqrt{\bar{\alpha}_1}}$，其中 $\bar{\alpha}_1=\alpha_1=1-\beta_1$。这样，$\mathcal{L}_0$ 就可以写为：

$$\begin{aligned}\mathcal{L}_0 &= \mathbb{E}_q\left[\sum_{i=1}^{D}\frac{\left(x_0^i-\mu_\theta^i\left(x_1,1\right)\right)^2}{2\sigma_1^2}\right]+\text{constant} \\ &= \mathbb{E}_q\left[\sum_{i=1}^{D}\frac{\left(\frac{x_1^i-\sqrt{1-\bar{\alpha}_1}\epsilon}{\sqrt{\bar{\alpha}_1}}-\frac{1}{\sqrt{\alpha_1}}\left[x_1^i-\frac{\beta_1}{\sqrt{1-\bar{\alpha}_1}}\epsilon_\theta\left(x_1^i,1\right)\right]\right)^2}{2\sigma_1^2}\right]+\text{constant} \\ &= \mathbb{E}_q\left[\sum_{i=1}^{D}\frac{\beta_1}{2\sigma_1^2\alpha_1}\|\epsilon-\epsilon_\theta\left(x_1^i,1\right)\|^2\right]+\text{constant}\,,\end{aligned} \tag{4.9}$$

最终 $\mathcal{L}_0$ 也被重参数化为和 $\mathcal{L}_{t-1}$ 一样的去噪形式。注意我们还没有定义 σ_1，如果采用 $\mathcal{L}_{t-1}$ 中 $t>1$ 时的定义 $\sigma_t^2=\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$，假设 $\bar{\alpha}_0=1$，则 $\sigma_1=0$，此时高斯分布的概率密度函数变为一个狄拉克 δ 函数，模型需要根据 x_1 直接预测出 x_0 的值。因此，在逆向过程的最后一步采样时，我们不进行加噪，直接输出 $\mu_\theta\left(x_1,1\right)=\frac{x_1-\sqrt{1-\bar{\alpha}_1}\epsilon_\theta(x_1,1)}{\sqrt{\bar{\alpha}_1}}$。

最后，将 $\mathcal{L}_{t-1}$ 和 $\mathcal{L}_0$ 合并，并省略常系数，得到最终的简化版的损失函数 $\mathcal{L}_{simple}$：

$$\mathcal{L}_{simple}=\mathbb{E}_{x_0,t,\epsilon}\left[\left\|\epsilon-\epsilon_\theta\left(\sqrt{\bar{\alpha}_t}x_0+\sqrt{1-\bar{\alpha}_t}\epsilon,t\right)\right\|^2\right], \tag{4.10}$$

其中 t 从 1 至 T 中均匀采样。直观上，在 DDPMs 训练时需要每次随机采样一个数据，并随机选择该数据前向过程中的某一步，训练模型拟合该步的逆，批次训练同理。省略常系数等价于对较小时间步的损失函数降权，因为噪声较小时预测相对简单，这种省略也能加快训练和提升效果。算法 4.1和算法 4.2分别展示了 DDPMs 的训练和采样算法。

算法 4.1 DDPMs 训练算法

Prepare: dataset distribution $q(x_0)$.

Initialize: ϵ_θ.

Run:

repeat

$x_0 \sim q\left(x_0\right)$

$t \sim \text{Uniform}\left(1, 2, \cdots, T\right)$

$\epsilon \sim \mathcal{N}\left(0, I\right)$

$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$

Update θ by taking gradient descent step on $\nabla_\theta \left\| \epsilon - \epsilon_\theta\left(x_t, t\right) \right\|^2$

until converged;

算法 4.2 DDPMs 采样算法

Prepare: well-trained DDPM ϵ_θ.

Run:

$x_T \sim \mathcal{N}\left(0, I\right)$

for $t = T$ **to** 1 **do**

$z \sim \mathcal{N}\left(0, I\right)$ if $t > 1$, else $z = 0$

$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left[x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta\left(x_t, t\right)\right] + \sigma_t z$

return x_0

至此，上述 DDPMs 均为无条件的，其生成结果不可控，如果想要引入条件，只需要将逆向过程定义为 $p_\theta\left(x_{t-1}|x_t, y\right)$，其中 y 为数据 x_0 对应的条件，如类别、文本等，并将其作为网络的输入，此时损失函数变为 $\mathcal{L}_{simple} = \mathbb{E}_{x_0,t,\epsilon}\left[\left\| \epsilon - \epsilon_\theta\left(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, y, t\right) \right\|^2\right]$。

4.3 从分数视角审视 DDPMs

在 DDPMs 的训练过程中，我们需要从 $q\left(x_t|x_0\right)=\mathcal{N}\left(\sqrt{\bar{\alpha}_t}x_0,\left(1-\bar{\alpha}_t\right)I\right)$ 中进行采样，考虑到其协方差矩阵为对角矩阵，我们可以将其视为单变量高斯分布，则有：

$$\begin{aligned} q\left(x_t|x_0\right)&=\frac{1}{\sqrt{1-\bar{\alpha}_t}\sqrt{2\pi}}\exp\left[-\frac{1}{2}\left(\frac{x_t-\sqrt{\bar{\alpha}_t}x_0}{\sqrt{1-\bar{\alpha}_t}}\right)^2\right] \\ \log q\left(x_t|x_0\right)&=\log\frac{1}{\sqrt{1-\bar{\alpha}_t}\sqrt{2\pi}}-\frac{1}{2}\left(\frac{x_t-\sqrt{\bar{\alpha}_t}x_0}{\sqrt{1-\bar{\alpha}_t}}\right)^2 \\ \nabla_{x_t}\log q\left(x_t|x_0\right)&=-\frac{x_t-\sqrt{\bar{\alpha}_t}x_0}{\sqrt{1-\bar{\alpha}_t}}\frac{1}{\sqrt{1-\bar{\alpha}_t}}=-\frac{1}{\sqrt{1-\bar{\alpha}_t}}\epsilon\,. \end{aligned} \tag{4.11}$$

由于 DDPMs 是训练 $\epsilon_\theta\left(x_t,t\right)$ 拟合 ϵ，而 $-\frac{1}{\sqrt{1-\bar{\alpha}_t}}$ 为常数，所以实际上 $\epsilon_\theta\left(x_t,t\right)$ 也是在拟合 $\nabla_{x_t}\log q\left(x_t|x_0\right)$，而 $\nabla_{x_t}\log q\left(x_t|x_0\right)$ 即为分布 $q\left(x_t|x_0\right)$ 的分数，根据式 1.10，DDPMs 在某一时间步 t 的优化目标等价于一个分数模型 $s_\theta\left(x_t\right)=\nabla_{x_t}\log p_\theta\left(x_t\right)$ 的去噪分数匹配法的优化目标。这表明 DDPMs 与 SMLD 殊途同归，两者都使用了基于去噪分数匹配法的多级别加噪技术，所以有相似的训练目标，两者之间的区别在于加噪方式和采样方式的不同，由于 DDPMs 对逆向过程更为细致和精确的定义，使得 DDPMs 在效果上要远超后者。

4.4 模型架构

在模型架构上，考虑到 ϵ_θ 的输入 x_t 和预测目标 ϵ 是相同维度的，DDPMs 采用经典的 UNet[10]网络结构，如图 4.3所示，最初它被提出用于进行图像分割。大体上，UNet 可以被分为两个部分，即图中左半部分的编码器和右半部分的解码器，编码器不断减小特征图的空间尺寸，同时增大其通道数；编码器编码的结果输入一个与编码器对称的解码器中，不断增大特征图的空间尺寸，同时减小其通道数，同时，将编码器相同空间尺寸的特征图跳跃连接（skip-connection）到解码器，与解码器的特征图相加并输入解码器的下一层，最终，模型输出和输入具有相同的空间尺寸。skip-connection 有助于缓解梯度消失问题，同时可以保留更多的原始输入信息，有助于提高网络的训练效果和性能，很多研究表明，skip-connection 是 DDPMs 的去噪重参数法能够成功的关键。UNet 具有很强的可拓展性，虽然图 4.3展示的是 256×256 分辨率的模型，但其可以拓展到任意分辨率。除此之外，DDPMs 还为 UNet 引入了自注意力层（self-attention layer）[11]以提升训

练效果。时间步 t 作为 ϵ_θ 的另一个输入，以位置编码[11]的形式引入到 UNet 的所有组归一化层（Group Normalization）[12]中。

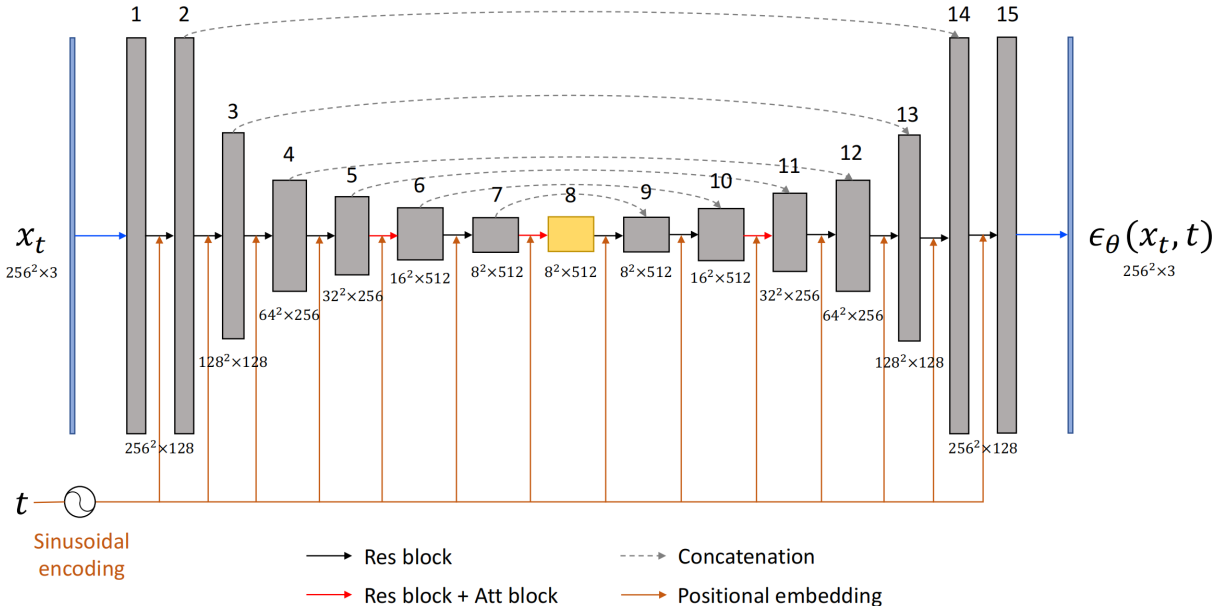

图 4.3 UNet 网络结构与数据流。

第 5 章 去噪扩散隐式模型

相比于经典的生成模型，如对抗生成网络（GANs）[13]和变分自编码器（VAEs）[8,14]，扩散模型训练更加稳定，效果更加出众，但其采样速度却要远慢于 GANs 和 VAEs，因为 DDPMs 需要重复调用网络 T 次，而 GANs 和 VAEs 都只需要一次。为此，Song 等人[15]提出去噪扩散隐式模型（Denoising Diffusion Implicit Models，DDIMs），其可以基于训练好的 DDPMs 模型进行加速采样。

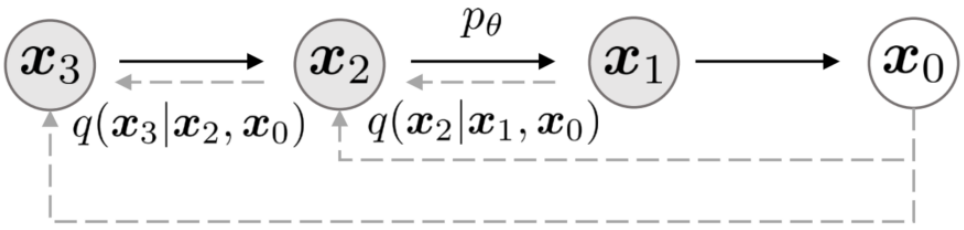

图 5.1 DDIMs 的概率图表示。

5.1 定义

具体来说，Song 等人[15]发现，DDPMs 的优化目标（即式 3.14）只依赖前向过程的边缘分布 $q\left(x_t|x_0\right)$，而不依赖其联合分布 $q\left(x_{1\cdots T}|x_0\right)$。然而，不同的联合分布却可能有相同的边缘分布，我们可以在不改变前向过程的边缘分布的前提下，重新定义一个更灵活的前向过程的联合分布，从而让逆向过程采样更加方便。具体来说，DDIMs 将前向过程重新定义为：

$$\begin{aligned} q\left(x_{1\cdots T}|x_0\right) &= q\left(x_1|x_0\right) q\left(x_2|x_1,x_0\right) q\left(x_3|x_2,x_0\right) \cdots q\left(x_T|x_{T-1},x_0\right) \\ &= q\left(x_1|x_0\right) \frac{q\left(x_1|x_2,x_0\right) q\left(x_2|x_0\right)}{q\left(x_1|x_0\right)} \frac{q\left(x_2|x_3,x_0\right) q\left(x_3|x_0\right)}{q\left(x_2|x_0\right)} \cdots \frac{q\left(x_{T-1}|x_T,x_0\right) q\left(x_T|x_0\right)}{q\left(x_{T-1}|x_0\right)} \\ &= q\left(x_T|x_0\right) \prod_{t=2}^{T} q\left(x_{t-1}|x_t,x_0\right) \ , \end{aligned} \tag{5.1}$$

这里同样用到了贝叶斯公式，即 $q\left(x_t|x_{t-1},x_0\right) = \frac{q(x_{t-1}|x_t,x_0)q(x_t|x_0)}{q(x_{t-1}|x_0)}$。这样，前向过程的扩散核 $q\left(x_t|x_{t-1},x_0\right)$ 就不再是马尔可夫式的了，因为 x_t 不再只依赖 x_{t-1}，还要依赖 x_0。当然，我们也可以依赖其它状态，但从之后的推导我们可以看到，依赖 x_0 的原因是其可以根据 x_t 和 $\epsilon_\theta\left(x_t,t\right)$ 估计出一个 x_0，从而使采样算法的计算变得可能。其概率图如图 5.1所

示。注意，我们将式 5.1写成 $q\left(x_{t-1}|x_t,x_0\right)$ 连乘的形式是为了与逆向过程匹配，并不改变前向过程的方向。如前所述，我们重新定义了前向过程的联合分布，但需要保证其边缘分布不变，因此我们需要对 $q\left(x_T|x_0\right)$ 和 $q\left(x_{t-1}|x_t,x_0\right)$ 进行定义，以保证 $q\left(x_t|x_0\right)=\mathcal{N}\left(\sqrt{\bar{\alpha}_t}x_0,\left(1-\bar{\alpha}_t\right)I\right)$ 不变。显然可以直接定义 $q\left(x_T|x_0\right)=\mathcal{N}\left(\sqrt{\bar{\alpha}_T}x_0,\left(1-\bar{\alpha}_T\right)I\right)$；考虑到

$$q\left(x_{t-1}|x_0\right)=\int_{x_t}q\left(x_{t-1},x_t|x_0\right)dx_t=\int_{x_t}q\left(x_t|x_0\right)q\left(x_{t-1}|x_t,x_0\right)dx_t\,, \tag{5.2}$$

已知 $q\left(x_{t-1}|x_0\right)=\mathcal{N}\left(\sqrt{\bar{\alpha}_{t-1}}x_0,\left(1-\bar{\alpha}_{t-1}\right)I\right)$ 和 $q\left(x_t|x_0\right)=\mathcal{N}\left(\sqrt{\bar{\alpha}_t}x_0,\left(1-\bar{\alpha}_t\right)I\right)$，根据《PRML》[16]中的定理 2.113-2.115，$q\left(x_{t-1}|x_t,x_0\right)$ 依然是高斯分布：

$$q\left(x_{t-1}|x_t,x_0\right)=\mathcal{N}\left(\sqrt{\bar{\alpha}_{t-1}}x_0+\sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\cdot\frac{x_t-\sqrt{\bar{\alpha}_t}x_0}{\sqrt{1-\bar{\alpha}_t}},\sigma_t^2I\right)\,, \tag{5.3}$$

其中 σ_t 为大于等于 0 的实数，它控制着前向过程的随机性，值越大随机性越大，当 $\sigma_t=0$ 时，前向过程变为一个确定的过程；当 $\sigma_t=\sqrt{\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}}\sqrt{1-\frac{\bar{\alpha}_t}{\bar{\alpha}_{t-1}}}$ 时，该公式与式 4.5中 DDPMs 前向过程的后验分布相同，所以我们一般取 $\sigma_t\left(\eta\right)=\eta\cdot\sqrt{\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}}\sqrt{1-\frac{\bar{\alpha}_t}{\bar{\alpha}_{t-1}}}$，其中超参数 η 控制前向过程随机性。此时，我们可以根据贝叶斯公式 $q\left(x_t|x_{t-1},x_0\right)=\frac{q(x_{t-1}|x_t,x_0)q(x_t|x_0)}{q(x_{t-1}|x_0)}$ 计算出新的前向过程的扩散核公式，它也是一个高斯分布，但不再具备马尔可夫性质，我们也并不会用到该公式。

重新定义了前向过程后，其对应的逆向过程也会发生改变，此时 $p_\theta\left(x_{t-1}|x_t\right)$ 需要拟合式 5.3。然而式 5.3中含有 x_0，这在逆向过程中是未知的，但是我们可以根据 $x_t=\sqrt{\bar{\alpha}_t}x_0+\sqrt{1-\bar{\alpha}_t}\epsilon$ 和 $\epsilon_\theta\left(x_t,t\right)\sim\epsilon$ 对 x_0 进行估算：

$$f_\theta\left(x_t,t\right)=\frac{x_t-\sqrt{1-\bar{\alpha}_t}\epsilon_\theta\left(x_t,t\right)}{\sqrt{\bar{\alpha}_t}}\,. \tag{5.4}$$

因此逆向过程可以定义为：

$$p_\theta\left(x_{t-1}|x_t\right)=\begin{cases}\mathcal{N}\left(f_\theta\left(x_1,1\right),\sigma_1^2I\right) & \text{if } t=1\\ q\left(x_{t-1}|x_t,f_\theta\left(x_t,t\right)\right) & \text{otherwise.}\end{cases} \tag{5.5}$$

根据式 5.1，逆向过程的采样公式为：

$$x_{t-1}=\sqrt{\bar{\alpha}_{t-1}}\left(\frac{x_t-\sqrt{1-\bar{\alpha}_t}\cdot\epsilon_\theta\left(x_t,t\right)}{\sqrt{\bar{\alpha}_t}}\right)+\sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\cdot\epsilon_\theta\left(x_t,t\right)+\sigma_t\epsilon\,, \tag{5.6}$$

其中 $\bar{\alpha}_0=1$，$\epsilon\sim\mathcal{N}\left(0,I\right)$。

5.2 优化目标推导

重新定义了前向与逆向过程后，我们需要重新推导 DDIMs 的优化目标，此时式 3.14变为：

$$\begin{aligned}\mathcal{L} &= \mathbb{E}_q\left[\underbrace{-\log p\left(x_0|x_1\right)}_{\mathcal{L}_0}+\sum_{t>1}\underbrace{D_{KL}\left(q\left(x_{t-1}|x_t,x_0\right)\parallel p\left(x_{t-1}|x_t\right)\right)}_{\mathcal{L}_{t-1}}\right] \\ &= \mathbb{E}_q\left[\underbrace{-\log p_\theta\left(x_0|x_1\right)}_{\mathcal{L}_0}+\sum_{t>1}\underbrace{D_{KL}\left(q\left(x_{t-1}|x_t,x_0\right)\parallel q\left(x_{t-1}|x_t,f_\theta\left(x_t,t\right)\right)\right)}_{\mathcal{L}_{t-1}}\right].\end{aligned} \tag{5.7}$$

对于 $\mathcal{L}_{t-1}$，我们将式 5.3和式 5.5代入式 4.6：

$$\begin{aligned}\mathcal{L}_{t-1} &= \mathbb{E}_q\left[c_t\left\|x_0-f_\theta\left(x_t,t\right)\right\|^2\right] \\ &= \mathbb{E}_q\left[c_t\left\|\frac{x_t-\sqrt{1-\bar{\alpha}_t}\epsilon}{\sqrt{\bar{\alpha}_t}}-\frac{x_t-\sqrt{1-\bar{\alpha}_t}\epsilon_\theta\left(x_t,t\right)}{\sqrt{\bar{\alpha}_t}}\right\|^2\right] \\ &= \mathbb{E}_q\left[c_t'\left\|\epsilon-\epsilon_\theta\left(x_t,t\right)\right\|^2\right],\end{aligned} \tag{5.8}$$

其中 c_t 和 c_t' 为常数。对于 $\mathcal{L}_0$：

$$\mathcal{L}_0 = \mathbb{E}_q\left[-\log p_\theta\left(x_0|x_1\right)\right] = \mathbb{E}_q\left[d\left\|x_0-f_\theta\left(x_1,1\right)\right\|^2\right] = \mathbb{E}_q\left[d'\left\|\epsilon-\epsilon_\theta\left(x_1,1\right)\right\|^2\right], \tag{5.9}$$

其中 d 和 d' 为常数。可以看到，DDIMs 的优化目标与 DDPMs 的优化目标只有不同时间步权重上的不同，所以训练好的 DDPMs 可以直接用于 DDIMs 采样而不需要重新训练。

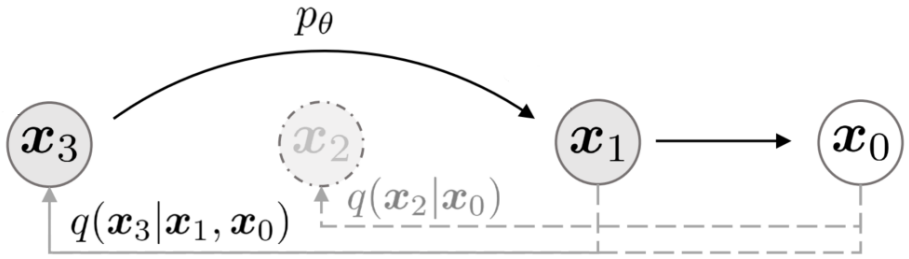

图 5.2 加速版 DDIMs 的概率图表示。

5.3 加速采样算法

至此，我们利用训练好的 DDPMs 定义了 DDIMs，然而其前向过程依然有 T 步，对应的逆向过程采样也需要 T，采样速度相比于 DDPMs 并没有差别。我们考虑少于 T

步的前向过程，我们定义一个 $\{0,1,\cdots,T\}$ 的子序列 $\tau=\{\tau_1,\tau_2,\cdots,\tau_S\}$，其长度为 S，$\tau_1=0$，$\tau_S=T$，类似式 5.1，我们可以重新定义前向过程为：

$$q\left(x_{1\cdots T}|x_0\right)=q\left(x_{\tau_S}|x_0\right)\prod_{i=1}^{S}q\left(x_{\tau_{i-1}}|x_{\tau_i},x_0\right)\prod_{t\in\bar{\tau}}q\left(x_t|x_0\right), \tag{5.10}$$

其中 $\bar{\tau}=\{0,1,\cdots,T\}\backslash\tau$。其概率图如图 5.2所示，$t\in\tau$ 的节点构成了一条链，x_0 的 $t\in\bar{\tau}$ 的节点构成了一个星图。类似地，为了确保前向过程的边缘分布不变，我们可以推导出：

$$\begin{aligned}
q\left(x_t|x_0\right)&=\mathcal{N}\left(\sqrt{\bar{\alpha}_t}x_0,\left(1-\bar{\alpha}_t\right)I\right)\qquad t\in\bar{\tau}\cup T\\
q\left(x_{\tau_{i-1}}|x_{\tau_i},x_0\right)&=\mathcal{N}\left(\sqrt{\bar{\alpha}_{\tau_{i-1}}}x_0+\sqrt{1-\bar{\alpha}_{\tau_{i-1}}-\sigma_{\tau_i}^2}\cdot\frac{x_{\tau_i}-\sqrt{\bar{\alpha}_{\tau_i}}x_0}{\sqrt{1-\bar{\alpha}_{\tau_i}}},\sigma_{\tau_i}^2I\right)\qquad i\in\{1,2,\cdots,S\}.
\end{aligned} \tag{5.11}$$

其对应的逆向过程也被重新定义为：

$$\begin{aligned}
p_\theta\left(x_{0\cdots T}\right)&=\underbrace{p_\theta\left(x_T\right)\prod_{i=1}^{S}p_\theta\left(x_{\tau_{i-1}}|x_{\tau_i}\right)}_{\text{use to sample}}\underbrace{\prod_{t\in\bar{\tau}}p_\theta\left(x_0|x_t\right)}_{\text{in objective}}\\
p_\theta\left(x_0|x_t\right)&=\mathcal{N}\left(f_\theta\left(x_t,t\right),\sigma_t^2I\right)\qquad t\in\bar{\tau}\cup T\\
p_\theta\left(x_{\tau_{i-1}}|x_{\tau_i}\right)&=q\left(x_{\tau_{i-1}}|x_{\tau_i},f_\theta\left(x_{\tau_i},\tau_i\right)\right)\qquad i\in\{2,3,\cdots,S\}.
\end{aligned} \tag{5.12}$$

此时我们只需要在子序列 τ 上进行采样，其对应的采样公式为：

$$x_{\tau_{i-1}}=\sqrt{\bar{\alpha}_{\tau_{i-1}}}\left(\frac{x_{\tau_i}-\sqrt{1-\bar{\alpha}_{\tau_i}}\cdot\epsilon_\theta\left(x_{\tau_i},\tau_i\right)}{\sqrt{\bar{\alpha}_{\tau_i}}}\right)+\sqrt{1-\bar{\alpha}_{\tau_{i-1}}-\sigma_{\tau_i}^2}\cdot\epsilon_\theta\left(x_{\tau_i},\tau_i\right)+\sigma_{\tau_i}\epsilon_{\tau_i}, \tag{5.13}$$

其中 $\bar{\alpha}_0=1$，$\epsilon_{\tau_i}\sim\mathcal{N}\left(0,I\right)$，$\sigma_{\tau_i}\left(\eta\right)=\eta\cdot\sqrt{\frac{1-\bar{\alpha}_{\tau_{i-1}}}{1-\bar{\alpha}_{\tau_i}}}\sqrt{1-\frac{\bar{\alpha}_{\tau_i}}{\bar{\alpha}_{\tau_{i-1}}}}$。此时，DDIMs 的优化目标变为：

$$\begin{aligned}
\mathcal{L}&=\mathbb{E}_q\left[\sum_{t\in\bar{\tau}}D_{KL}\left(q\left(x_t|x_0\right)\parallel p_\theta\left(x_0|x_t\right)\right)+\sum_{i=1}^{S}D_{KL}\left(q\left(x_{\tau_{i-1}}|x_{\tau_i},x_0\right)\parallel p_\theta\left(x_{\tau_{i-1}}|x_{\tau_i}\right)\right)\right]\\
&=\mathbb{E}_q\left[\sum_{t\in\bar{\tau}}D_{KL}\left(q\left(x_t|x_0\right)\parallel p_\theta\left(x_0|x_t\right)\right)+\sum_{i=1}^{S}D_{KL}\left(q\left(x_{\tau_{i-1}}|x_{\tau_i},x_0\right)\parallel q\left(x_{\tau_{i-1}}|x_{\tau_i},f_\theta\left(x_{\tau_i},\tau_i\right)\right)\right)\right].
\end{aligned} \tag{5.14}$$

由于采样时不涉及 $t\in\bar{\tau}$，所以 $t\in\bar{\tau}$ 的优化目标可以忽略，而 $t\in\tau$ 优化目标与 DDPMs 的优化目标也只有不同时间步权重上的不同，所以训练好的 DDPMs 可以直接用于 DDIMs 加速采样而不需要重新训练。算法 5.1展示了 DDIMs 的采样算法。我们可以使用任意长

度的子序列 τ 进行 DDIMs 加速采样，然而加速采样会导致采样样本质量降低，实践中一般使用 $S = 100$ 或 $S = 50$ 步，以达到采样质量和采样速度之间的平衡。

算法 5.1 DDIMs 采样算法

Prepare: well-trained DDPM ϵ_θ.

Input: sampling sequence $\{\tau_i\}_{i=1}^{S}$ where $\tau_1 = 0$ and $\tau_S = T$, randomness control parameter σ_t

Run:

$x_T \sim \mathcal{N}(0, I)$

for $i = S$ **to** 2 **do**

$\epsilon \sim \mathcal{N}(0, I)$

$x_{\tau_{i-1}} = \sqrt{\bar{\alpha}_{\tau_{i-1}}} \left(\frac{x_t - \sqrt{1-\bar{\alpha}_{\tau_i}} \epsilon_\theta(x_{\tau_i}, \tau_i)}{\sqrt{\bar{\alpha}_{\tau_i}}} \right) + \sqrt{1 - \bar{\alpha}_{t_{\tau-1}} - \sigma_t^2} \cdot \epsilon_\theta(x_{\tau_i}, \tau_i) + \sigma_t \epsilon$

return x_0

5.4 从常微分方程视角审视 DDIMs

值得注意的是，式 5.6也为我们提供了另一个观察扩散模型的视角，在 $\sigma_t = 0$ 时有：

$$
\begin{aligned}
x_{t-1} &= \sqrt{\bar{\alpha}_{t-1}} \left(\frac{x_t - \sqrt{1-\bar{\alpha}_t} \cdot \epsilon_\theta(x_t, t)}{\sqrt{\bar{\alpha}_t}} \right) + \sqrt{1 - \bar{\alpha}_{t-1}} \cdot \epsilon_\theta(x_t, t) \\
\sqrt{\frac{1}{\bar{\alpha}_{t-1}}} x_{t-1} &= \sqrt{\frac{1}{\bar{\alpha}_t}} x_t + \left(\sqrt{\frac{1-\bar{\alpha}_{t-1}}{\bar{\alpha}_{t-1}}} - \sqrt{\frac{1-\bar{\alpha}_t}{\bar{\alpha}_t}} \right) \epsilon_\theta(x_t, t) ,
\end{aligned}
\tag{5.15}
$$

可以看到 DDIMs 采样公式可以被视为一个常微分方程（ODE），DDIMs 采样即可视为在初值为 x_T 的情况下对该 ODE 进行数值求解，DDIMs 加速采样即可视为步长较长的数值求解方式，其速度更快，但误差更大。此外，我们可以在时间序列上反向使用该公式，对某个 x_0 进行 DDIMs 加噪至某个 x_T，该 x_T 即为该 DDIMs 生成 x_0 所需的初始噪声，该技术被广泛用于基于扩散模型的编辑、翻译等任务以保留原始图像的信息。这与 GANs 中的逆映射技术（GAN inversion）类似，其将给定图像映射到预训练 GANs 的隐变量空间，以得到可以从生成器重构该图像的隐变量，方便进行图像编辑。

第 6 章 对扩散模型的思考

在式 3.9的推导过程中，我们利用马尔可夫性质为前向过程的后验分布引入了 x_0，如果不引入 x_0，会得到：

$$\begin{aligned} K &= \int dx_{0:T} q\left(x_{0:T}\right)\left[\log p\left(x_T\right)+\sum_{t\geq 1}\log\frac{p\left(x_{t-1}|x_t\right)}{q\left(x_t|x_{t-1}\right)}\right] \\ &= \int dx_{0:T} q\left(x_{0:T}\right)\left[\log p\left(x_T\right)+\sum_{t\geq 1}\log\frac{p\left(x_{t-1}|x_t\right)}{q\left(x_{t-1}|x_t\right)}\frac{q\left(x_{t-1}\right)}{q\left(x_t\right)}\right] \\ &= \int dx_{0:T} q\left(x_{0:T}\right)\left[\log\frac{p\left(x_T\right)}{q\left(x_T\right)}+\sum_{t\geq 1}\log\frac{p\left(x_{t-1}|x_t\right)}{q\left(x_{t-1}|x_t\right)}-\log q\left(x_0\right)\right], \end{aligned} \tag{6.1}$$

我们可以称使用该优化目标的扩散模型为原始扩散模型（Vanilla Diffusion Models）。原始扩散模型假设 $q\left(x_{t-1}|x_t\right)$ 是高斯分布并期望对其进行拟合，有两种情况可以保证 $q\left(x_{t-1}|x_t\right)$ 是高斯分布：由于 $q\left(x_{t-1}|x_t\right)=\frac{q(x_t|x_{t-1})q(x_{t-1})}{q(x_t)}$，当步长很小时，$q\left(x_t\right)$ 和 $q\left(x_{t-1}\right)$ 很接近，$\frac{q(x_t|x_{t-1})q(x_{t-1})}{q(x_t)}$ 被 $q\left(x_t|x_{t-1}\right)$ 主导，此时 $q\left(x_{t-1}|x_t\right)$ 和 $q\left(x_t|x_{t-1}\right)$ 拥有相同的函数形式[7]；根据《PRML》2.113、2.114 和 2.116，如果 $q\left(x_{t-1}|x_t\right)$ 和 $q\left(x_t\right)$ 都是高斯分布（注意是 $q\left(x_t\right)$ 不是 $q\left(x_t|x_0\right)$），那么 $q\left(x_{t-1}|x_t\right)$ 就是高斯分布。实践中很难保证 $q\left(x_t\right)$ 是高斯分布，因为数据分布 $q\left(x_0\right)$ 一般都不是高斯分布。

原始扩散模型使用第一种情况保证 $q\left(x_{t-1}|x_t\right)$ 是高斯分布，然而虽然 $q\left(x_{t-1}|x_t\right)$ 和 $q\left(x_t|x_{t-1}\right)$ 具有相同的函数形式（即高斯分布），但它是 intractable 的，所以原始扩散模型是无法训练的。但如果以 x_0 为条件，根据马尔可夫性质可知 $q\left(x_t|x_{t-1},x_0\right)=q\left(x_t|x_{t-1}\right)$ 是高斯分布，由于 $q\left(x_t|x_0\right)$ 也是高斯分布，且都有闭式解，利用上述的第二种情况可以推出 $q\left(x_{t-1}|x_t,x_0\right)$ 也是高斯分布且有闭式解，这就是 DDPMs。值得注意的是，此时我们就不需要步长很小这一前提了，因为 $p_\theta\left(x_{t-1}|x_t\right)$ 的拟合目标 $q\left(x_{t-1}|x_t,x_0\right)$ 天然是高斯分布，与步长无关；马尔可夫性质也不是必需的，因为只要 $q\left(x_t|x_{t-1},x_0\right)$ 是高斯分布即可。DDIMs 利用这两点进行大步长采样，但大步长（步数少）时扩散模型拟合的变分下界更差，效果就会更差，所以步长还是越小越好，即步数越多越好，变分扩散模型（Variational Diffusion Models）[17]也证明了这一点。因此，DDPMs 实际上是使用了第二种情况来确保以高斯形式学习逆向过程 p_θ，但其文献中的表述是根据第一种情况来确保的，实际上只是因为步数太少时效果会差，所以才使用了较多的步数，而不是为了确保 p_θ 是高斯形式才使用了较多的步数。

以 x_0 为条件很重要，式 1.10中的 Denoising Score Matching 也是以 x_0 为条件后变得简洁优雅；Flow Matching[18]中的拟合目标 u_t 本身是 intractable 的，这导致 Flow Matching 无法训练，但如果让 u_t 以 x_0 为条件，其就变得 tractable 了，此时 Flow Matching 就被转换为 Conditional Flow Matching，可以进行训练。正如 Flow Matching 必须转换成 Conditional Flow Matching 才能训练一样，扩散模型也必须转换为以 x_0 为条件才能训练，因为原始扩散模型中 $p_\theta\left(x_{t-1}|x_t\right)$ 要拟合的 $q\left(x_{t-1}|x_t\right)$ 是 intractable 的，导致其无法训练。

第 7 章 分类器指导采样

7.1 动机

如前所述，对于有条件的 DDPMs，我们需要训练 $\epsilon_\theta(x_t, y, t)$ 拟合 ϵ，然而训练 DDPMs 是比较消耗计算资源的，每当我们需要引入新的条件时都需要重新训练模型。幸运的是，Sohl 等人[6,19]提出了分类器指导采样（Classifier-Guided Sampling）的方法，使得我们可以基于无条件的 DDPMs 进行有条件的采样。

7.2 基于 DDPMs 的公式推导与采样算法

对于有条件的逆向过程，Sohl 等人[6,19]证明了：

$$p_{\theta,\phi}(x_t|x_{t+1}, y) = Z p_\theta(x_t|x_{t+1})\, p_\phi(y|x_t)\ , \tag{7.1}$$

其中 Z 为归一化常数，$p_\phi(y|x_t)$ 是参数为 ϕ 的噪声版本的分类器（noisy classifier）。

虽然我们只将条件引入到逆向过程的建模中，但为了推导方便，我们定义一个与原前向过程 q 等价的有条件的前向过程 $\hat{q}$：

$$\begin{aligned}
\hat{q}(x_0) &= q(x_0) \\
\hat{q}(y|x_0) &= \text{Known labels per sample} \\
\hat{q}(x_{t+1}|x_t, y) &= q(x_{t+1}|x_t) \\
\hat{q}(x_{1:T}|x_0, y) &= \prod_{t=1}^{T} \hat{q}(x_t|x_{t-1}, y)\ .
\end{aligned} \tag{7.2}$$

我们首先推导出无条件的前向过程的扩散核 $\hat{q}\left(x_{t+1}|x_t\right)$：

$$\begin{aligned}
\hat{q}\left(x_{t+1}|x_t\right) &= \int_y \hat{q}\left(x_{t+1}, y|x_t\right)\, dy \\
&= \int_y \hat{q}\left(x_{t+1}|x_t, y\right) \hat{q}\left(y|x_t\right)\, dy \\
&= \int_y q\left(x_{t+1}|x_t\right) \hat{q}\left(y|x_t\right)\, dy \\
&= q\left(x_{t+1}|x_t\right) \int_y \hat{q}\left(y|x_t\right)\, dy \\
&= q\left(x_{t+1}|x_t\right) \\
&= \hat{q}\left(x_{t+1}|x_t, y\right)\,.
\end{aligned} \tag{7.3}$$

类似地，我们可以推导出无条件的前向过程的联合分布：

$$\begin{aligned}
\hat{q}\left(x_{1:T}|x_0\right) &= \int_y \hat{q}\left(x_{1:T}, y|x_0\right)\, dy \\
&= \int_y \hat{q}\left(y|x_0\right) \hat{q}\left(x_{1:T}|x_0, y\right)\, dy \\
&= \int_y \hat{q}\left(y|x_0\right) \prod_{t=1}^{T} \hat{q}\left(x_t|x_{t-1}, y\right)\, dy \\
&= \int_y \hat{q}\left(y|x_0\right) \prod_{t=1}^{T} q\left(x_t|x_{t-1}\right)\, dy \\
&= \prod_{t=1}^{T} q(x_t|x_{t-1}) \int_y \hat{q}\left(y|x_0\right)\, dy \\
&= \prod_{t=1}^{T} q(x_t|x_{t-1}) \\
&= q\left(x_{1:T}|x_0\right)\,.
\end{aligned} \tag{7.4}$$

利用该公式，我们可以得到 $\hat{q}\left(x_t\right) = q\left(x_t\right)$：

$$\begin{aligned}
\hat{q}\left(x_t\right) &= \int_{x_{0:t-1}} \hat{q}\left(x_{0:t}\right)\, dx_{0:t-1} \\
&= \int_{x_{0:t-1}} \hat{q}\left(x_0\right) \hat{q}\left(x_{1:t}|x_0\right)\, dx_{0:t-1} \\
&= \int_{x_{0:t-1}} q\left(x_0\right) q\left(x_{1:t}|x_0\right)\, dx_{0:t-1} \\
&= \int_{x_{0:t-1}} q\left(x_{0:t}\right)\, dx_{0:t-1} \\
&= q(x_t)\,.
\end{aligned} \tag{7.5}$$

有了 $\hat{q}(x_t) = q(x_t)$ 和 $\hat{q}(x_{t+1}|x_t) = q(x_{t+1}|x_t)$,根据贝叶斯公式,我们可以得到 $\hat{q}(x_t|x_{t+1}) = q(x_t|x_{t+1})$。进一步，我们可以证明 $\hat{q}(y|x_t)$ 与 x_{t+1} 是无关的，即：

$$\begin{aligned}\hat{q}(y|x_t, x_{t+1}) &= \hat{q}(x_{t+1}|x_t, y)\frac{\hat{q}(y|x_t)}{\hat{q}(x_{t+1}|x_t)} \\ &= \hat{q}(x_{t+1}|x_t)\frac{\hat{q}(y|x_t)}{\hat{q}(x_{t+1}|x_t)} \\ &= \hat{q}(y|x_t)\ .\end{aligned} \tag{7.6}$$

最后，我们推导出有条件的前向过程的后验分布 $\hat{q}(x_t|x_{t+1}, y)$：

$$\begin{aligned}\hat{q}(x_t|x_{t+1}, y) &= \frac{\hat{q}(x_t, x_{t+1}, y)}{\hat{q}(x_{t+1}, y)} \\ &= \frac{\hat{q}(x_t, x_{t+1}, y)}{\hat{q}(y|x_{t+1})\hat{q}(x_{t+1})} \\ &= \frac{\hat{q}(x_t|x_{t+1})\hat{q}(y|x_t, x_{t+1})\hat{q}(x_{t+1})}{\hat{q}(y|x_{t+1})\hat{q}(x_{t+1})} \\ &= \frac{\hat{q}(x_t|x_{t+1})\hat{q}(y|x_t, x_{t+1})}{\hat{q}(y|x_{t+1})} \\ &= \frac{\hat{q}(x_t|x_{t+1})\hat{q}(y|x_t)}{\hat{q}(y|x_{t+1})} \\ &= \frac{q(x_t|x_{t+1})\hat{q}(y|x_t)}{\hat{q}(y|x_{t+1})}\ .\end{aligned} \tag{7.7}$$

由于 $\hat{q}(y|x_{t+1})$ 和 x_t 是无关的,所以可以将其看为常数,即 $\hat{q}(x_t|x_{t+1}, y) = Zq(x_t|x_{t+1})\hat{q}(y|x_t)$，这就是逆向过程需要拟合的公式，对于 $q(x_t|x_{t+1})$，我们可以使用预训练好的无条件 DDPMs $p_\theta(x_t|x_{t+1})$ 进行拟合，对于 $\hat{q}(y|x_t)$，我们可以额外训练一个分类器 $p_\phi(y|x_t)$ 进行拟合，最终，逆向过程为 $p_{\theta,\phi}(x_t|x_{t+1}, y) = Zp_\theta(x_t|x_{t+1})p_\phi(y|x_t)$。

在 $p_{\theta,\phi}(x_t|x_{t+1}, y) = Zp_\theta(x_t|x_{t+1})p_\phi(y|x_t)$ 中，$p_\theta(x_t|x_{t+1})$ 是已知的：

$$\begin{aligned}p_\theta(x_t|x_{t+1}) &= \mathcal{N}(\mu, \Sigma) \\ \log p_\theta(x_t|x_{t+1}) &= -\frac{1}{2}(x_t - \mu)^T\Sigma^{-1}(x_t - \mu) + C\ .\end{aligned} \tag{7.8}$$

由于高斯分布 $p_\theta(x_t|x_{t+1})$ 的协方差都较小，其概率密度函数曲线是一个在均值 μ 附近又高又窄的钟型，所以从 $p_\theta(x_t|x_{t+1})$ 中采样得到的 x_t 大都集中在 μ 附近，因此，我们可以使用 $x_t = \mu$ 处的泰勒展开来估计 $\log p_\phi(y|x_t)$，即：

$$\begin{aligned}\log p_\phi(y|x_t) &\approx \log p_\phi(y|x_t)|_{x_t=\mu} + (x_t - \mu)\nabla_{x_t}\log p_\phi(y|x_t)|_{x_t=\mu} \\ &= (x_t - \mu)g + C_1\ ,\end{aligned} \tag{7.9}$$

其中 $g = \nabla_{x_t} \log p_\phi (y|x_t)|_{x_t=\mu}$，$\log p_\phi (y|x_t)|_{x_t=\mu}$ 与 x_t 无关，视为 C_1 常数。逆向过程 $p_{\theta,\phi}(x_t|x_{t+1}, y) = Z p_\theta (x_t|x_{t+1}) p_\phi (y|x_t)$ 的对数可以表示为：

$$\begin{aligned}
\log \left[p_\theta (x_t|x_{t+1}) p_\phi (y|x_t)\right] &\approx -\frac{1}{2}(x_t - \mu)^T \Sigma^{-1}(x_t - \mu) + (x_t - \mu) g + C_2 \\
&= -\frac{1}{2}(x_t - \mu - \Sigma g)^T \Sigma^{-1}(x_t - \mu - \Sigma g) + \frac{1}{2} g^T \Sigma g + C_2 \\
&= -\frac{1}{2}(x_t - \mu - \Sigma g)^T \Sigma^{-1}(x_t - \mu - \Sigma g) + C_3 \\
&= \log p(z) + C_4,\ z \sim \mathcal{N}(\mu + \Sigma g, \Sigma),
\end{aligned} \tag{7.10}$$

其中 C_2, C_3, C_4 均为常数。最终，在无条件 DDPMs 的逆向过程 $p_\theta (x_t|x_{t+1}) = \mathcal{N}(\mu, \Sigma)$ 的基础上，实现对符合条件 y 的样本的采样只需要对其均值加一个偏移量 Σg。

在实践中，Dhariwal 等人[19]发现，如果按上述方法在 ImageNet 数据集[20]上进行分类器指导采样，那么生成样本中大概只有一半符合想要的类别，一个解决方案是给 g 乘上一个大于 1 的常数因子 s，此时 $s \cdot \nabla_{x_t} \log p_\phi (y|x_t) = \nabla_{x_t} \log \frac{1}{Z} p_\phi (y|x_t)^s$，其中 Z 为一个常数，其将 $p_\phi (y|x_t)^s$ 归一化为一个概率分布。当 $s > 1$ 时，如果 $1 > a > b > 0$，那么 $\frac{a^s}{b^s} = \left(\frac{a}{b}\right)^s > \frac{a}{b}$，所以 $p_\phi (y|x_t)$ 中较大的值相比于较小的值会被放大，导致 $\frac{1}{Z} p_\phi (y|x_t)^s$ 的概率密度函数曲线变得更加尖锐，分布区间变得更窄，导致采样的确定性增加，生成样本的质量得到提升，但多样性被抑制，这被称为截断效应（truncation effect）。算法 7.1展示了分类器指导的 DDPMs 采样算法。

算法 7.1 Classifier-guided DDPMs 采样算法

Prepare: well-trained DDPM ϵ_θ, well-trained noisy classifier $p_\phi(y|x_t)$, classifier guidance scale s.

Input: desired condition y

Run:

$x_T \sim \mathcal{N}(0, I)$

for $t = T$ **to** 1 **do**

- $z \sim \mathcal{N}(0, I)$ if $t > 1$, else $z = 0$
- $\mu = \frac{1}{\sqrt{\alpha_t}} \left[x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta (x_t, t)\right]$
- $g = \nabla_{x_t} \log p_\phi (y|x_t)|_{x_t=\mu}$
- $x_{t-1} = \mu + s \cdot \sigma_t \cdot g + \sigma_t z$

return x_0

7.3 基于 DDIMs 的采样算法

上述推导只对 DDPMs 有效，无法直接应用于 DDIMs，为此，我们需要使用一些基于分数技巧进行推导。根据式 4.11，我们可以从分数的视角来看 $\epsilon_\theta\left(x_t, t\right)$：

$$\nabla_{x_t} \log p_\theta\left(x_t\right)=-\frac{1}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta\left(x_t, t\right) , \tag{7.11}$$

其中 $\nabla_{x_t} \log p_\theta\left(x_t\right)$ 是一个分数模型。进一步，根据贝叶斯公式 $p\left(x_t|y\right)=\frac{p(x_t)p(y|x_t)}{p(y)}$ 可得：

$$\begin{aligned} \nabla_{x_t} \log p_\theta\left(x_t|y\right) &= \nabla_{x_t} \log \left(p_\theta\left(x_t\right) p_\phi\left(y|x_t\right)\right) \\ &= \nabla_{x_t} \log p_\theta\left(x_t\right)+\nabla_{x_t} \log p_\phi\left(y|x_t\right) \\ &= -\frac{1}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta\left(x_t, t\right)+\nabla_{x_t} \log p_\phi\left(y|x_t\right) , \end{aligned} \tag{7.12}$$

其中 $\nabla_{x_t} \log p_\theta\left(x_t|y\right)$ 是一个有条件的分数模型。我们定义一个有条件的 DDIMs 模型 $\hat{\epsilon}_\theta\left(x_t, t\right)$，从分数的视角来看，$\nabla_{x_t} \log p_\theta\left(x_t|y\right)=-\frac{1}{\sqrt{1-\bar{\alpha}_t}} \hat{\epsilon}_\theta\left(x_t, t\right)$，将其代入上式可以得到：

$$\hat{\epsilon}_\theta\left(x_t, t\right)=\epsilon_\theta\left(x_t, t\right)-\sqrt{1-\bar{\alpha}_t} \nabla_{x_t} \log p_\phi\left(y|x_t\right) . \tag{7.13}$$

使用 $\hat{\epsilon}_\theta\left(x_t, t\right)$ 代替式 5.6中的 $\epsilon_\theta\left(x_t, t\right)$ 即可实现分类器指导的 DDIMs 采样。算法 7.2展示了分类器指导的 DDIMs 采样算法。

算法 7.2 Classifier-guided DDIMs 采样算法

Prepare: well-trained DDPM ϵ_θ, well-trained noisy classifier $p_\phi\left(y|x_t\right)$, classifier guidance scale s.

Input: sampling sequence $\{\tau_i\}_{i=1}^S$ where $\tau_1=0$ and $\tau_S=T$, desired condition y, randomness control parameter σ_t

Run:

$x_T \sim \mathcal{N}\left(0, I\right)$

for $i=S$ **to** 2 **do**

$\epsilon \sim \mathcal{N}\left(0, I\right)$

$\hat{\epsilon}_\theta\left(x_{\tau_i}, \tau_i\right)=\epsilon_\theta\left(x_{\tau_i}, \tau_i\right)-s \cdot \sqrt{1-\bar{\alpha}_{\tau_i}} \cdot \nabla_{x_{\tau_i}} \log p_\phi\left(y|x_{\tau_i}\right)$

$x_{\tau_{i-1}}=\sqrt{\bar{\alpha}_{\tau_{i-1}}}\left(\frac{x_t-\sqrt{1-\bar{\alpha}_{\tau_i}} \hat{\epsilon}_\theta\left(x_{\tau_i}, \tau_i\right)}{\sqrt{\bar{\alpha}_{\tau_i}}}\right)+\sqrt{1-\bar{\alpha}_{t_{\tau-1}}+\sigma_t^2} \cdot \hat{\epsilon}_\theta\left(x_{\tau_i}, \tau_i\right)+\sigma_t \epsilon$

return x_0

7.4 分类器拓展

值得注意的是，目前为止我们都假设 $p_\phi(y|x_t)$ 是一个离散分类器，只能实现对指定类别的采样。实践中我们可以将该理论拓展到任意的相似度度量函数 $D(x_t, c)$ 或者损失函数 $L(x_t, c)$ 上，其中 c 为任意条件，$D(x_t, c)$ 衡量 x_t 与 c 之间的相似度，值越大代表它们越相似，$L(x_t, c)$ 衡量 x_t 与 c 之间的差异度，值越小代表它们差异度越小，本质上它们是等价的。我们使用 $\nabla_{x_t} D(x_t, c)$ 或 $\nabla_{x_t} - L(x_t, c)$ 替换算法 7.1和算法 7.2中的 $\nabla_{x_t} \log p_\phi(y|x_t)$ 即可实现在训练好的无条件扩散模型上采样符合条件 c 的样本，这使得我们可以将任意训练好的感知模型以分类器的形式应用到扩散模型的可控生成中。

具体来说，对于某个感知模型 $p(y|x)$（如素描提取模型、物体分割模型等），其根据输入图像 x 预测目标 y（如素描图和分割图等）。如果想要使用预训练扩散模型生成符合某个给定条件 y 的图像，只需要使用感知模型预测 x_t 对应的目标 y_t，计算 y_t 与 y 之间的相似度，使用其对 x_t 的梯度作为指导进行采样。相似度的计算方式需要根据目标的属性进行定义，一般可以使用感知模型训练时的损失函数的形式进行定义，如素描图之间的 MSE 损失。

然而，一个很显然的问题是，现有的感知模型几乎都是在非带噪图像上进行训练的，不支持对 x_t 进行预测，目前有两种方法解决这一问题：一种方法是在带噪图像数据上对感知模型进行微调（当然也可以重新训练）；另一种方法是根据公式 $\hat{x}_0 = \frac{x_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta(x_t, t)}{\sqrt{\bar{\alpha}_t}}$ 从扩散模型每一步采样结果中估计出一个非带噪图像并输入感知模型进行预测。实践中第二种方法更为常用，因此，现有的感知模型都可以用于分类器指导采样，并且不需要额外的训练，这极大地拓展和增强了扩散模型的生成能力。当然，这种方法的效果是不如重新训练一个条件扩散模型的，首先，分类器指导采样的推导本身就涉及到一些假设和近似处理，其次，根据公式计算出的非带噪图像 $\hat{x}_0$ 也是不精确的，特别是当 t 较大时，$\hat{x}_0$ 中带有明显的噪声，这超出了已有的判别模型能够识别的分布。

第 8 章 通过随机微分方程求解基于分数的生成模型

如节 4.3所述，DDPMs 和 SMLD 有相似的训练目标，但由于 DDPMs 对逆向过程更为细致和精确的定义，使得 DDPMs 在效果上要远超后者。Song 等人[21]提出了通过随机微分方程（SDE）求解的基于分数的生成模型，使得其在效果上追平甚至赶超了扩散模型，并一定程度了统一了两者。SDE 的概念最早以布朗运动（Brownina motion）的形式由爱因斯坦提出，随后由郎之万、伊藤等人完善。SDE 是 ODE 的扩展，其项是随机过程，解也是随机过程，用于形容一个随机变量的变动过程。

8.1 定义

类似于扩散模型，我们需要先构建一个连续时间的扩散过程 $\{x_t\}_{t=0}^{1}$，其中 $x_0 \sim p_{\text{data}}(x)$，$x_1$ 是某个已知的、简单的、容易采样的分布。我们可以使用如下前向 SDE 对其进行描述：

$$dx = f(x,t)\,dt + g(t)\,d\mathrm{w}\,, \tag{8.1}$$

其中 w 是标准维纳过程（Wiener process），也即布朗运动，$f(\cdot,t) \in \mathbb{R}^d \to \mathbb{R}^d$ 称为拖拽系数，d 为 x 的维度，$g(\cdot) \in \mathbb{R} \to \mathbb{R}$ 称为扩散系数。实际上扩散系数也应为一个 $d \times d$ 的矩阵，但为了简单，我们假设它为一个标量。Anderson 等人[22]证明式 8.1中描述的扩散过程的逆可以被该逆向 SDE 描述：

$$dx = \left[f(x,t) - g^2(t)\,\nabla_x \log p_t(x)\right] dt + g(t)\,d\bar{\mathrm{w}}\,, \tag{8.2}$$

其中 $\bar{\mathrm{w}}$ 是另一个逆向的标准维纳过程，即 t 是从 1 向 0 的；$p_t(x)$ 是 x_t 的概率密度函数。可以看到，只要我们知道了 $\nabla_x \log p_t(x)$，我们就可以模拟该 SDE 进行采样。因此，我们学习一个分数模型 $s_\theta(x_t,t)$ 去拟合 $\nabla_x \log p_t(x)$，使用去噪分数匹配法，根据式 1.10，损失函数可以转化为：

$$\mathcal{L} = \mathbb{E}_{x_0,t}\left[\lambda_t \| s_\theta(x_t,t) - \nabla_{x_t} \log p(x_t|x_0) \|_2^2\right]\,, \tag{8.3}$$

其中 t 取自 $[0,1]$ 的连续均匀分布中，$\lambda_t \in [0,1] \to \mathbb{R}^+$ 定义了不同时间步 t 的损失函数的权重，$p(x_t|x_0)$ 由式 8.1定义，x_t 根据 x_0 和 t 从 $p(x_t|x_0)$ 中采样而来。值得注意的是，

当式 8.1中的 $f(x,t)$ 是仿射函数时，$p(x_t|x_0)$ 就是一个高斯分布，其分数有闭式解。可以看到，该训练目标就是基于分数的生成模型的训练目标，所以本质上，这里只是使用 SDE 重新解释和定义了基于分数的生成模型，并得到了一个效果更好的采样方法。

8.2 SMLD 与 DDPMs 的 SDE 形式

下面我们证明 SMLD 和 DDPMs 所使用的加噪方式分别对应式 8.1中两种不同的 SDE 的离散化形式。

对于 SMLD，假设其噪声级别序列为 $\{\sigma_i\}_{i=1}^N$，根据其加噪公式 $q_{\sigma_i}(\tilde{x}|x)=\mathcal{N}(x,\sigma_i^2 I)$，我们可以推导出其对应的离散形式的加噪过程为：

$$x_i = x_{i-1} + \sqrt{\sigma_i^2 - \sigma_{i-1}^2}\, z_{i-1}\,, \tag{8.4}$$

其中 $i=1,2,\cdots,N$，$z_{i-1}\sim\mathcal{N}(0,I)$，$x_0\sim p_{\text{data}}(x)$，$\sigma_0=0$。我们取 $N\to\infty$，该离散过程变为一个连续过程，定义 $t\in\{0,\frac{1}{N-1},\frac{2}{N-1},\cdots,\frac{N-2}{N-1}\}$，$\Delta t=\frac{1}{N-1}$，$x\left(\frac{i-1}{N-1}\right)=x_i$，$\sigma\left(\frac{i-1}{N-1}\right)=\sigma_i$，$z\left(\frac{i-1}{N-1}\right)=z_i$，上式变为：

$$\begin{aligned} x(t+\Delta t) &= x(t) + \sqrt{\sigma^2(t+\Delta t)-\sigma^2(t)}\, z(t) \approx x(t) + \sqrt{\frac{\Delta\left[\sigma^2(t)\right]}{\Delta t}}\sqrt{\Delta t}\, z(t)\,, \\ dx &= \sqrt{\frac{d\left[\sigma^2(t)\right]}{dt}}\, d\mathrm{w}\,. \end{aligned} \tag{8.5}$$

这里我们用到了标准维纳过程的一个性质，即 $\mathrm{w}_t-\mathrm{w}_s\sim\mathcal{N}(0,(t-s)I)$，其中 $t>s\geq 0$。因此，其对应前向 SDE 中 $f(x,t)=0$ 和 $g(t)=\sqrt{\frac{d[\sigma^2(t)]}{dt}}$。由于该加噪方法是不断增大分布的方差，所以称之为方差爆炸法（Variance Exploding，VE）。

对于 DDPMs，根据式 4.1，其离散形式的加噪过程为：

$$x_i = \sqrt{1-\beta_i}\, x_{i-1} + \sqrt{\beta_i}\, z_{i-1}\,, \tag{8.6}$$

其中 $i=1,2,\cdots,N$，$z_{i-1}\sim\mathcal{N}(0,I)$。我们定义辅助变量 $\bar{\beta}_i=(N-1)\beta_i$，则 $\beta_i=\frac{\bar{\beta}_i}{N-1}$，定义 $t\in\{0,\frac{1}{N-1},\frac{2}{N-1},\cdots,\frac{N-2}{N-1}\}$，$\Delta t=\frac{1}{N-1}$，$\bar{\beta}\left(\frac{i-1}{N-1}\right)=\bar{\beta}_i$，$x\left(\frac{i-1}{N-1}\right)=x_i$，$z\left(\frac{i-1}{N-1}\right)=z_i$，

上式变为：

$$
\begin{aligned}
x_i &= \sqrt{1-\frac{\bar{\beta}_i}{N-1}}x_{i-1}+\sqrt{\frac{\bar{\beta}_i}{N-1}}z_{i-1}\,,\\
x\left(t+\Delta t\right) &= \sqrt{1-\bar{\beta}\left(t\right)\Delta t}x\left(t\right)+\sqrt{\bar{\beta}\left(t\right)\Delta t}z\left(t\right)\approx x\left(t\right)-\frac{1}{2}\bar{\beta}\left(t\right)\Delta t x\left(t\right)+\sqrt{\bar{\beta}\left(t\right)}\sqrt{\Delta t}z\left(t\right)\,,\\
dx &= -\frac{1}{2}\bar{\beta}\left(t\right)x dt+\sqrt{\bar{\beta}\left(t\right)}d\mathrm{w}\,.
\end{aligned}
\tag{8.7}
$$

这里用到了小量近似，即当 $|x|\ll 1$ 且 α 不太大时 $(1+x)^{\alpha}\approx 1+\alpha x$。因此，其对应前向 SDE 中 $f\left(x,t\right)=-\frac{1}{2}\bar{\beta}\left(t\right)x$ 和 $g\left(t\right)=\sqrt{\bar{\beta}\left(t\right)}$。由于该加噪方法相对于 VE 并不使方差爆炸，所以称之为方差保持法（Variance Preserving，VP）。

我们将 SMLD 和 DDPMs 的加噪过程转化为等价的与式 8.1相同形式的 SDE，分别称为 VE SDE 和 VP SDE，如前所述，它们对应的 $p\left(x_t|x_0\right)$ 都是一个高斯分布，其分数有闭式解，所以都可以通过优化式 8.3分别学习得到一个分数模型。

8.3 逆向扩散采样算法

在根据前向 SDE 训练好分数模型后，我们需要使用其进行采样，即求解式 8.2中的 SDE，由于其没有闭式解，所以我们只能通过数值方法求解。一些通用的 SDE 数值求解算法如 Euler-Maruyama 法和 stochastic Runge-Kutta 法[23]也适用于此，本质上，它们是对式 8.2进行不同形式的离散化。自然地，我们也可以采用和前向 SDE 一样的离散化形式来离散化并求解逆向 SDE。

对于 VE SDE 的逆向 SDE，我们采用式 8.5中的离散方法，则其采样过程为：

$$
\begin{aligned}
dx &= \left[f\left(x,t\right)-g^2\left(t\right)\nabla_x\log p_t\left(x\right)\right]dt+g\left(t\right)d\bar{\mathrm{w}}\,,\\
x\left(t\right)-x\left(t+\Delta t\right) &= -\frac{\Delta\sigma^2\left(t\right)}{\Delta t}\cdot s_\theta\left(x\left(t+\Delta t\right),t+\Delta t\right)\cdot\left[t-\left(t+\Delta t\right)\right]+\sqrt{\frac{\Delta\sigma^2\left(t\right)}{\Delta t}}\sqrt{\Delta t}z\,,\\
x_i &= x_{i+1}+\left(\sigma_{i+1}^2-\sigma_i^2\right)s_\theta\left(x_{i+1},i+1\right)+\sqrt{\sigma_{i+1}^2-\sigma_i^2}z\,,
\end{aligned}
\tag{8.8}
$$

其中 $i=N-1,N-2,\cdots,0$，$z\sim\mathcal{N}\left(0,I\right)$。注意，由于 t 是逆向流动的，所以 dt 应转换为 $-\Delta t$，其中 $\Delta t=\frac{1}{N-1}$，由于 $\bar{\mathrm{w}}$ 本身是逆向的，所以不需要这种转换。

对于 VP SDE 的逆向 SDE，我们采用式 8.7中的离散方法，则其采样过程为：

$$
\begin{aligned}
dx &= \left[f\left(x,t\right) - g^2\left(t\right)\nabla_x \log p_t\left(x\right)\right]dt + g\left(t\right)d\bar{\mathrm{w}}\,,\\
x\left(t\right) - x\left(t+\Delta t\right) &= \left[-\frac{1}{2}\bar{\beta}\left(t+\Delta t\right)x\left(t+\Delta t\right) - \bar{\beta}\left(t+\Delta t\right)\cdot s_\theta\left(x\left(t+\Delta t\right),t+\Delta t\right)\right]\cdot\left[t-\left(t+\Delta t\right)\right]\\
&\quad + \sqrt{\bar{\beta}\left(t+\Delta t\right)}\sqrt{\Delta t}z\,,\\
x_i &= x_{i+1} + \frac{1}{2}\beta_{i+1}x_{i+1} + \beta_{i+1}s_\theta\left(x_{i+1},i+1\right) + \sqrt{\beta_{i+1}}z\,,\\
x_i &\approx x_{i+1} + \left(1-\sqrt{1-\beta_{i+1}}\right)x_{i+1} + \beta_{i+1}s_\theta\left(x_{i+1},i+1\right) + \sqrt{\beta_{i+1}}z\,,\\
x_i &\approx \left(2-\sqrt{1-\beta_{i+1}}\right)x_{i+1} + \beta_{i+1}s_\theta\left(x_{i+1},i+1\right) + \sqrt{\beta_{i+1}}z\,,
\end{aligned}
\tag{8.9}
$$

其中 $i = N-1, N-2, \cdots, 0$，$z \sim \mathcal{N}\left(0,I\right)$。值得注意的是，如果将 DDPMs 视为一个分数模型，即式 4.11中的 $s_\theta\left(x_t,t\right) = -\frac{1}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta\left(x_t,t\right)$，上述采样过程与算法 4.2中的 DDPMs 采样过程近似等价，证明如下：

$$
\begin{aligned}
x_i &= \frac{1}{\sqrt{\alpha_{i+1}}}\left[x_{i+1} - \frac{\beta_{i+1}}{\sqrt{1-\bar{\alpha}_{i+1}}}\epsilon_\theta\left(x_{i+1},i+1\right)\right] + \sigma_{i+1}z\\
&\approx \frac{1}{\sqrt{1-\beta_{i+1}}}\left(x_{i+1} + \beta_{i+1}s_\theta\left(x_{i+1},i+1\right)\right) + \sqrt{\beta_{i+1}}z\\
&= \left(1-\beta_{i+1}\right)^{-\frac{1}{2}}\left(x_{i+1} + \beta_{i+1}s_\theta\left(x_{i+1},i+1\right)\right) + \sqrt{\beta_{i+1}}z\\
&\approx \left(1+\frac{1}{2}\beta_{i+1}\right)\left(x_{i+1} + \beta_{i+1}s_\theta\left(x_{i+1},i+1\right)\right) + \sqrt{\beta_{i+1}}z\\
&= \left(1+\frac{1}{2}\beta_{i+1}\right)x_{i+1} + \beta_{i+1}s_\theta\left(x_{i+1},i+1\right) + \frac{1}{2}\beta_{i+1}^2 s_\theta\left(x_{i+1},i+1\right) + \sqrt{\beta_{i+1}}z\\
&\approx \left(1+\frac{1}{2}\beta_{i+1}\right)x_{i+1} + \beta_{i+1}s_\theta\left(x_{i+1},i+1\right) + \sqrt{\beta_{i+1}}z\\
&= \left[2-\left(1-\frac{1}{2}\beta_{i+1}\right)\right]x_{i+1} + \beta_{i+1}s_\theta\left(x_{i+1},i+1\right) + \sqrt{\beta_{i+1}}z\\
&\approx \left(2-\sqrt{1-\beta_{i+1}}\right)x_{i+1} + \beta_{i+1}s_\theta\left(x_{i+1},i+1\right) + \sqrt{\beta_{i+1}}z\,.
\end{aligned}
\tag{8.10}
$$

这里我们将 $\sigma_t^2 = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$ 近似为 β_t。因此，DDPMs 可以视为该框架下一个特殊的前向 SDE（VP SDE）和其对应的逆向 SDE 上的一种特殊的离散化形式。

我们将这种采用和前向 SDE 一样的离散化形式来离散化并求解逆向 SDE 的采样方法称为逆向扩散（Reverse Diffusion）采样法。

8.4 预测器-校正器采样算法

在实践中，Reverse Diffusion 采样法的效果已经足够好了，但我们仍然可以利用 $s_\theta(x_t, t)$ 继续提升采样效果。具体来说，由于 $s_\theta(x_t, t) \approx \nabla_{x_t} \log p_t(x_t)$，这让我们可以使用基于分数的 MCMC 采样法（如式 1.6中的 Langevin MCMC 采样法）直接从分布 $p_t(x_t)$ 中进行采样，所以我们可以在使用不同数值方法求解逆向 SDE 时，将每一步计算出的 x_t 作为初始值，并使用基于分数的 MCMC 采样方法对其进行校正，使其更加符合分布 $p_t(x_t)$。我们将 Reverse Diffusion 采样法作为预测器，Langevin MCMC 采样法作为校正器，得到预测器-校正器（Predictor-Corrector，PC）采样算法。算法 8.1和算法 8.2分别展示了对于 VE SDE 和 VP SDE 的逆向 SDE 的 PC 采样算法，其中外循环为预测器，内循环为校正器。实践中，我们可以只使用预测器（即 $M = 0$），此时算法退化为 Reverse Diffusion 采样法；也可以只使用校正器（即 $N = 0$），此时算法退化为 SMLD 使用的 Annealed Langevin MCMC 采样法，但采样效果会变差，这再次证明了 SMLD 比 DDPMs 效果差的原因在于采样算法不够好；当两者结合时，在相同计算量的前提下，采样效果会比 Reverse Diffusion 采样法的要稍好。

算法 8.1 Predictor-Corrector 采样算法 (VE SDE)

Prepare: well-trained score-based model s_θ

Input: predictor step N, corrector step M, Langevin MCMC step size $\{\epsilon_i\}_{i=0}^{N-1}$

Run:

$x_N \sim \mathcal{N}(0, \sigma_N^2 I)$

for $i = N-1$ **to** 0 **do**

  $z \sim \mathcal{N}(0, I)$

  $x_i = x_{i+1} + \left(\sigma_{i+1}^2 - \sigma_i^2\right) s_\theta(x_{i+1}, i+1) + \sqrt{\sigma_{i+1}^2 - \sigma_i^2} z$

  for $j = 1$ **to** M **do**

    $z \sim \mathcal{N}(0, I)$

    $x_i = x_i + \frac{\epsilon_i^2}{2} s_\theta(x_{i+1}, i+1) + \epsilon_i z$

return x_0

算法 8.2 Predictor-Corrector 采样算法 (VP SDE)

Prepare: well-trained score-based model s_θ

Input: predictor step N, corrector step M, Langevin MCMC step size $\{\epsilon_i\}_{i=0}^{N-1}$

Run:

$x_N \sim \mathcal{N}(0, I)$

for $i = N-1$ **to** 0 **do**

  $z \sim \mathcal{N}(0, I)$

  $x_i = \left(2 - \sqrt{1-\beta_{i+1}}\right) x_{i+1} + \beta_{i+1} s_\theta(x_{i+1}, i+1) + \sqrt{\beta_{i+1}} z$

  for $j = 1$ **to** M **do**

    $z \sim \mathcal{N}(0, I)$

    $x_i = x_i + \frac{\epsilon_i^2}{2} s_\theta(x_{i+1}, i+1) + \epsilon_i z$

return x_0

8.5 概率流常微分方程

对于式 8.1描述的扩散过程，福克-普朗克方程（Fokker-Planck equation）[24]以偏微分方程的形式描述了其概率密度函数 $p_t(x)$ 是如何随时间演化的：

$$\frac{\partial p_t(x)}{\partial t} = -\frac{\partial}{\partial x}\left[f(x,t)\, p_t(x)\right] + \frac{\partial^2}{(\partial x)^2}\left[\frac{1}{2} g^2(t)\, p_t(x)\right]. \tag{8.11}$$

我们重新构造一个前向 SDE：

$$\begin{aligned} dx &= \tilde{f}(x,t)\, dt + \tilde{g}(t)\, d\mathrm{w} \\ \tilde{f}(x,t) &= f(x,t) - \frac{1}{2} g^2(t)\, \nabla_x \log \tilde{p}_t(x) \\ \tilde{g}(t) &= 0, \end{aligned} \tag{8.12}$$

其对应的福克-普朗克方程为：

$$\begin{aligned} \frac{\partial \tilde{p}_t(x)}{\partial t} &= -\frac{\partial}{\partial x}\left[\tilde{f}(x,t)\, \tilde{p}_t(x)\right] + \frac{\partial^2}{(\partial x)^2}\left[\frac{1}{2}\tilde{g}^2(t)\, \tilde{p}_t(x)\right] \\ &= -\frac{\partial}{\partial x}\left[\left(f(x,t) - \frac{1}{2} g^2(t)\, \nabla_x \log \tilde{p}_t(x)\right) \tilde{p}_t(x)\right] \\ &= -\frac{\partial}{\partial x}\left[f(x,t)\, \tilde{p}_t(x)\right] + \frac{\partial}{\partial x}\left[\frac{1}{2} g^2(t)\, \frac{1}{\tilde{p}_t(x)} \tilde{p}_t(x)\, \frac{\partial}{\partial x} \tilde{p}_t(x)\right] \\ &= -\frac{\partial}{\partial x}\left[f(x,t)\, \tilde{p}_t(x)\right] + \frac{\partial^2}{(\partial x)^2}\left[\frac{1}{2} g^2(t)\, \tilde{p}_t(x)\right], \end{aligned} \tag{8.13}$$

其与式 8.11一致，这表明 $\tilde{p}_t(x) = p_t(x)$，所以我们构造的 SDE 与式 8.1中的 SDE 拥有相同的概率密布函数 $p_t(x)$。因此，对于式 8.1描述的每个扩散过程，都存在一个对应的确定性的过程，其可以被如下 ODE 描述：

$$dx = \left[f(x,t) - \frac{1}{2} g^2(t) \nabla_x \log p_t(x) \right] dt, \tag{8.14}$$

同时，式 8.2中的逆向 SDE 也变换为该 ODE，表明了其可逆性，该 ODE 被称为概率流常微分方程（probability flow ODE，PF ODE）。所以，式 8.8中 VE 的采样过程变为：

$$x_i = x_{i+1} + \frac{1}{2}\left(\sigma_{i+1}^2 - \sigma_i^2\right) s_\theta(x_{i+1}, i+1). \tag{8.15}$$

式 8.9中 VP 的采样过程变为：

$$x_i = \left(2 - \sqrt{1-\beta_{i+1}}\right) x_{i+1} + \frac{1}{2}\beta_{i+1} s_\theta(x_{i+1}, i+1). \tag{8.16}$$

PF ODE 有两个显而易见的好处：第一，我们可以应用更多的 ODE 数值方法（包括一些黑盒求解器）进行更加高效的采样，以达到采样时间和采样质量的平衡；第二，由于 ODE 是确定和可逆的，所以该 ODE 可以用于对数据进行扩散，得到生成某个给定图像的初值，在图像编辑、翻译等场景中使用。进一步，Song 等人[25]基于 PF ODE 提出了一致性模型（Consistency Models），极大地加快了生成速度。

8.6 DDIMs 常微分方程与概率流常微分方程的关系

此外，如果将 DDPMs 视为一个分数模型，即式 4.11中的 $s_\theta(x_t, t) = -\frac{1}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)$，那么式 5.6中的 DDIMs 采样公式可以写为：

$$\begin{aligned}
x_i &= \sqrt{\bar{\alpha}_i}\left(\frac{x_{i+1} - \sqrt{1-\bar{\alpha}_{i+1}}\cdot\epsilon_\theta(x_{i+1}, i+1)}{\sqrt{\bar{\alpha}_{i+1}}}\right) + \sqrt{1-\bar{\alpha}_i}\cdot\epsilon_\theta(x_{i+1}, i+1) \\
&= \frac{1}{\sqrt{\alpha_{i+1}}} x_{i+1} - \frac{\sqrt{1-\bar{\alpha}_{i+1}}}{\sqrt{\alpha_i}}\epsilon_\theta(x_{i+1}, i+1) + \sqrt{1-\bar{\alpha}_i}\cdot\epsilon_\theta(x_{i+1}, i+1) \\
&= \frac{1}{\sqrt{1-\beta_{i+1}}} x_{i+1} + \left(\frac{1-\bar{\alpha}_{i+1}}{\sqrt{\alpha_i}} - \sqrt{1-\bar{\alpha}_i}\sqrt{1-\bar{\alpha}_{i+1}}\right) s_\theta(x_{i+1}, i+1) \\
&\approx \left(2 - \sqrt{1-\beta_{i+1}}\right) x_{i+1} + \left(\frac{1-\bar{\alpha}_{i+1}}{\sqrt{\alpha_i}} - \sqrt{1-\bar{\alpha}_i}\sqrt{1-\bar{\alpha}_{i+1}}\right) s_\theta(x_{i+1}, i+1).
\end{aligned} \tag{8.17}$$

对比其与式 8.16中的采样公式，图 8.1展示了 $\frac{1}{2}\beta_i$ 与 $\frac{1-\bar{\alpha}_{i+1}}{\sqrt{\alpha_i}} - \sqrt{1-\bar{\alpha}_i}\sqrt{1-\bar{\alpha}_{i+1}}$ 的值，它们近似相等，所以式 5.15中的 DDIMs 常微分方程与同系数的 DDPMs 对应的的 VP SDE（式 8.7）的概率流常微分方程近似等价。

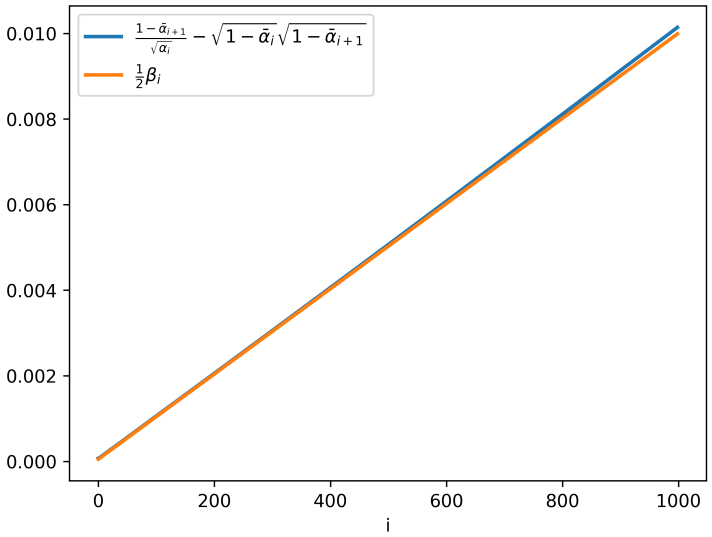

图 8.1 $\frac{1}{2}\beta_i$ 与 $\frac{1-\bar{\alpha}_{i+1}}{\sqrt{\alpha_i}} - \sqrt{1-\bar{\alpha}_i}\sqrt{1-\bar{\alpha}_{i+1}}$ 的值。

另一方面，如果将式 5.15中的 $\sqrt{\frac{1}{\bar{\alpha}_t}}x_t$ 视为一个整体 y_t，定义 $\sigma_t = \sqrt{\frac{1-\bar{\alpha}_t}{\bar{\alpha}_t}}$，则 $x_t = \frac{y_t}{\sqrt{1+\sigma_t^2}}$，$\bar{\alpha}_t = \frac{1}{1+\sigma_t^2}$，此时式 5.15中的 DDIMs 常微分方程变为：

$$y_{t-1} = y_t + (\sigma_{t-1} - \sigma_t)\,\epsilon_\theta\left(\frac{y_t}{\sqrt{1+\sigma_t^2}}, t\right), \tag{8.18}$$

其对应的常微分方程为：

$$\frac{dy(t)}{dt} = \frac{d\sigma(t)}{dt}\epsilon_\theta\left(\frac{y(t)}{\sqrt{1+\sigma^2(t)}}, t\right). \tag{8.19}$$

进一步，关于 $y(t)$ 的 VE SDE 对应的概率流常微分方程为：

$$dy(t) = -\frac{1}{2}g^2(t)\,\nabla_y \log p_t(y)\,dt, \tag{8.20}$$

其中 $g(t) = \sqrt{\frac{d\sigma^2(t)}{dt}}$，如果训练一个分数模型 $s_\theta(y(t), t)$，那么其拟合的是 $\nabla_y \log p_t(y)$，即 $-\frac{\epsilon}{\sigma(t)}$，由于 $y(t)$ 与 $x(t)$ 是等价的，所以 $s_\theta(y(t), t) = -\frac{\epsilon_\theta\left(\frac{y(t)}{\sqrt{1+\sigma^2(t)}}, t\right)}{\sigma(t)}$，则概率流常微

分方程变为：

$$\begin{aligned} dy\left(t\right) &= -\frac{1}{2}g^{2}\left(t\right)s_{\theta}\left(y\left(t\right),t\right) \\ &= \frac{1}{2}\frac{d\sigma^{2}\left(t\right)}{dt}\frac{\epsilon_{\theta}\left(\frac{y(t)}{\sqrt{1+\sigma^{2}(t)}},t\right)}{\sigma\left(t\right)}dt \\ &= \frac{d\sigma\left(t\right)}{dt}\epsilon_{\theta}\left(\frac{y\left(t\right)}{\sqrt{1+\sigma^{2}\left(t\right)}},t\right)dt\,, \\ \frac{dy\left(t\right)}{dt} &= \frac{d\sigma\left(t\right)}{dt}\epsilon_{\theta}\left(\frac{y\left(t\right)}{\sqrt{1+\sigma^{2}\left(t\right)}},t\right)\,, \end{aligned} \tag{8.21}$$

其与上述 DDIMs 常微分方程一致，所以 DDIMs 常微分方程又与另一个系数为 $\sigma_t = \sqrt{\frac{1-\bar{\alpha}_t}{\bar{\alpha}_t}}$ 的 VE SDE 的概率流常微分方程近似等价。

参考文献

[1] PARISI G. Correlation functions and computer simulations[J]. Nuclear Physics B, 1981, 180(3): 378-384.

[2] GRENANDER U, MILLER M I. Representations of knowledge in complex systems[J]. Journal of the Royal Statistical Society: Series B (Methodological), 1994, 56(4): 549-581.

[3] HYVÄRINEN A, DAYAN P. Estimation of non-normalized statistical models by score matching.[J]. Journal of Machine Learning Research, 2005, 6(4).

[4] VINCENT P. A connection between score matching and denoising autoencoders[J]. Neural computation, 2011, 23(7): 1661-1674.

[5] SONG Y, ERMON S. Generative modeling by estimating gradients of the data distribution[J]. Advances in neural information processing systems, 2019, 32.

[6] SOHL-DICKSTEIN J, WEISS E A, MAHESWARANATHAN N, et al. Deep unsupervised learning using nonequilibrium thermodynamics[J]. arXiv preprint arXiv:1503.03585, 2015.

[7] FELLER W. RETRACTED CHAPTER: On the Theory of Stochastic Processes, with Particular Reference to Applications[G]//Selected Papers I. Springer, 2015: 769-798.

[8] KINGMA D P, WELLING M. Auto-encoding variational bayes[J]. arXiv preprint arXiv:1312.6114, 2013.

[9] HO J, JAIN A, ABBEEL P. Denoising diffusion probabilistic models[J]. arXiv preprint arXiv:2006.11239, 2020.

[10] RONNEBERGER O, FISCHER P, BROX T. U-net: Convolutional networks for biomedical image segmentation[C]//Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18. 2015: 234-241.

[11] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.

[12] WU Y, HE K. Group normalization[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 3-19.

[13] GOODFELLOW I, POUGET-ABADIE J, MIRZA M, et al. Generative adversarial nets[J]. Advances in neural information processing systems, 2014, 27.

[14] REZENDE D J, MOHAMED S, WIERSTRA D. Stochastic backpropagation and approximate inference in deep generative models[C]//International conference on machine learning. 2014: 1278-1286.

[15] SONG J, MENG C, ERMON S. Denoising diffusion implicit models[J]. arXiv preprint arXiv:2010.02502, 2020.

[16] BISHOP C M. Pattern recognition[J]. Machine learning, 2006, 128(9).

[17] KINGMA D P, SALIMANS T, POOLE B, et al. Variational diffusion models[J]. arXiv preprint arXiv:2107.00630, 2021.

[18] LIPMAN Y, CHEN R T, BEN-HAMU H, et al. Flow matching for generative modeling[J]. arXiv preprint arXiv:2210.02747, 2022.

[19] DHARIWAL P, NICHOL A. Diffusion models beat gans on image synthesis[J]. Advances in neural information processing systems, 2021, 34: 8780-8794.

[20] DENG J, DONG W, SOCHER R, et al. Imagenet: A large-scale hierarchical image database[C]//2009 IEEE conference on computer vision and pattern recognition. 2009: 248-255.

[21] SONG Y, SOHL-DICKSTEIN J, KINGMA D P, et al. Score-based generative modeling through stochastic differential equations[J]. arXiv preprint arXiv:2011.13456, 2020.

[22] ANDERSON B D. Reverse-time diffusion equation models[J]. Stochastic Processes and their Applications, 1982, 12(3): 313-326.

[23] KLOEDEN P E, PLATEN E, SCHURZ H. Numerical solution of SDE through computer experiments [M]. Springer Science & Business Media, 2012.

[24] ØKSENDAL B, ØKSENDAL B. Stochastic differential equations[M]. Springer, 2003.

[25] SONG Y, DHARIWAL P, CHEN M, et al. Consistency models[J]. arXiv preprint arXiv:2303.01469,

2023.