生成模型 1：DDPM

Step 1：数据向量化与建模对象

设原始数据为图像等高维对象。经过固定的预处理（如解码、缩放、标准化）后，每个样本记为向量或张量

$$x_0 \in \mathbb{R}^d. \tag{1}$$

例如，$H \times W$ 的灰度图可以展平为 $d = HW$ 维向量，RGB 图像可以展平为 $d = 3HW$ 维向量。本文统一把数据表示写成 $x_0 \in \mathbb{R}^d$。

假设这些样本是从某个真实但未知的总体分布抽样得到：

$$x_0 \sim p_{\text{data}}(x_0). \tag{2}$$

这里的 p_{data} 是一个理论上的真实分布，无法显式写出；训练时用经验平均近似关于 p_{data} 的期望，即

$$\mathbb{E}_{x_0 \sim p_{\text{data}}}[f(x_0)] \approx \frac{1}{N} \sum_{i=1}^{N} f\big(x_0^{(i)}\big). \tag{3}$$

在理论上，p_{data} 表示真实数据的总体分布，一般无法显式写出（如果你都知道的话，那就不需学一个生成模型了）；训练时只能通过有限样本的经验分布和参数化模型 p_θ 去近似它。目标是构造一个参数化模型分布 $p_\theta(x_0)$，在分布意义上尽量接近 $p_{\text{data}}(x_0)$，从而在模型中重现真实数据的统计结构（**生成模型的本质**）。

Step 2：目标函数——从 KL 到最大似然

我们希望 $p_\theta(x_0)$ 逼近 $p_{\text{data}}(x_0)$，一个自然的度量是 KL 散度：

$$\mathrm{KL}\big(p_{\text{data}} \,\|\, p_\theta\big) = \mathbb{E}_{x_0 \sim p_{\text{data}}}\big[\log p_{\text{data}}(x_0) - \log p_\theta(x_0)\big]. \tag{4}$$

第一项 $\mathbb{E}[\log p_{\text{data}}(x_0)]$ 与 θ 无关，因此

$$\arg\min_\theta \mathrm{KL}\big(p_{\text{data}} \,\|\, p_\theta\big) = \arg\max_\theta \mathbb{E}_{x_0 \sim p_{\text{data}}}[\log p_\theta(x_0)]. \tag{5}$$

这就是最大似然目标：

$$\max_\theta \ \mathbb{E}_{x_0 \sim p_{\text{data}}}[\log p_\theta(x_0)]. \tag{6}$$

用样本平均近似，有经验目标

$$\max_\theta \ \frac{1}{N} \sum_{i=1}^{N} \log p_\theta\big(x_0^{(i)}\big). \tag{7}$$

问题在于：对高维图像，直接构造、优化 $p_\theta(x_0)$ 非常困难。DDPM 的核心思想是：引入一个可控的前向“加噪”过程 $q(x_{1:T} \mid x_0)$，再构造一个反向生成过程 $p_\theta(x_{0:T})$，通过变分下界把式 *(6)* 中对 $\log p_\theta(x_0)$ 的优化，转化为一系列局部、简单的子问题。

在 DDPM 的框架中，结论告诉我们，实际最终优化目标等价于：一个关于“噪声预测误差”的期望损失。**“补充”** 中给出了原因。

Step 3：模型结构——前向扩散与反向生成链

本步回答：在什么空间上、用什么随机过程来建模 $p_\theta(x_0)$。

前向扩散过程 $q(x_{1:T} \mid x_0)$

选择一个步数 T，给定噪声调度 $\{\beta_t\}_{t=1}^T \subset (0,1)$，定义

$$\alpha_t = 1 - \beta_t, \quad \bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s. \tag{8}$$

前向过程 q 是一个从 x_0 出发的马尔可夫链：

$$q(x_{1:T} \mid x_0) = \prod_{t=1}^{T} q(x_t \mid x_{t-1}), \tag{9}$$

其中每一步都是加各向同性高斯噪声：

$$q(x_t \mid x_{t-1}) = \mathcal{N}\big(\sqrt{\alpha_t}\, x_{t-1}, (1-\alpha_t)I\big). \tag{10}$$

由于高斯的叠加仍为高斯，可以直接写出 $q(x_t \mid x_0)$ 的闭式：

$$q(x_t \mid x_0) = \mathcal{N}\big(\sqrt{\bar{\alpha}_t}\, x_0, (1-\bar{\alpha}_t)I\big). \tag{11}$$

即我们可以在一步中构造任意时间步的带噪样本：

$$x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1-\bar{\alpha}_t}\, \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, I). \tag{12}$$

注意，这里的 ε 维度为 d，I 维度为 $d * d$. 训练时，式 (12) 用于从干净样本 x_0 直接生成任意噪声等级 t 的输入 x_t。

反向生成模型 $p_\theta(x_{0:T})$

为了生成数据，我们需要一个从纯噪声“反向走回”数据的过程。引入反向马尔可夫链：

$$p_\theta(x_{0:T}) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1} \mid x_t). \tag{13}$$

通常选顶层先验为标准高斯：

$$p(x_T) = \mathcal{N}(0, I). \tag{14}$$

对每个反向条件分布 $p_\theta(x_{t-1} \mid x_t)$，假设其为高斯：

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}\big(\mu_\theta(x_t, t),\, \Sigma_t\big), \tag{15}$$

其中 Σ_t 为仅依赖时间步 t 的对角协方差，典型设定为

$$\Sigma_t = \sigma_t^2 I. \tag{16}$$

这个 σ_t^2 通常不是随便取的，而是从前向过程的噪声调度推出来的一个函数，例如直接选为前向后验的方差 $\tilde{\beta}_t$，或者在 β_t 和 $\tilde{\beta}_t$ 之间做插值。通过这个马尔可夫链，边缘分布 $p_\theta(x_0)$ 就由 $\{p_\theta(x_{t-1} \mid x_t)\}$ 诱导出来。

核心 DL 学习目标具体是什么？以及细节补充？

模型框架已经搭好了。我们要清楚：1. 什么部分是可学习的？或者说，学习目标是什么。给答案：学习目标是一个噪声预测网络，我们希望它尽可能贴近原噪声，即高斯标准噪声；为什么预测噪声就可以？补充中给了原因。2. 反向过程中的反向均值 μ 具体形式是什么？

先解决第一个问题!! 由前向过程式 (12) 可知，在给定 x_0 的情况下，x_t 满足

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t}\, \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, I). \tag{17}$$

因此，x_t 可以看作 “衰减后的 x_0 加一层高斯噪声”。DDPM 的学习目标是这层噪声，定义一个随时间步变化的噪声预测网络（例如 U-Net，现成）：希望满足

$$\varepsilon_\theta(x_t, t) -> \varepsilon. \tag{19}$$

其中左边是预测的，右边是高斯标准噪声。

再解决第 2 个问题!! 给定 (x_t, t) 和网络输出 $\varepsilon_\theta(x_t, t)$，可以构造对 x_0 的估计：

$$\hat{x}_{0,\theta}(x_t, t) = \frac{1}{\sqrt{\bar{\alpha}_t}} \left(x_t - \sqrt{1 - \bar{\alpha}_t}\, \varepsilon_\theta(x_t, t)\right). \tag{20}$$

利用高斯条件分布公式，可以把反向均值 $\mu_\theta(x_t, t)$ 写成

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}}\, \varepsilon_\theta(x_t, t)\right). \tag{21}$$

于是反向条件分布为

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}\Big(\frac{1}{\sqrt{\alpha_t}}\big(x_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}}\, \varepsilon_\theta(x_t, t)\big),\, \sigma_t^2 I\Big). \tag{22}$$

实现上，网络直接输出 $\varepsilon_\theta(x_t, t)$，$\mu_\theta(x_t, t)$ 按式 (21) 计算。

到此为止，模型结构（Step 3）已经完整定义：前向 $q(x_{1:T} \mid x_0)$ 固定、只用于加噪；反向 $p_\theta(x_{0:T})$ 由噪声预测网络 ε_θ 决定。

Step 4: 训练算法

设 batch 大小为 B，一次迭代过程为：

1. 从数据集中采样 mini-batch $\{x_0^{(i)}\}_{i=1}^{B}$。

2. 对每个样本 i，采样时间步

$$t^{(i)} \sim \text{Uniform}\{1, \dots, T\}. \tag{24}$$

3. 对每个样本 i，采样高斯噪声

$$\varepsilon^{(i)} \sim \mathcal{N}(0, I), \tag{25}$$

并按式 (12) 构造带噪样本

$$x_{t^{(i)}}^{(i)} = \sqrt{\bar{\alpha}_{t^{(i)}}}\, x_0^{(i)} + \sqrt{1 - \bar{\alpha}_{t^{(i)}}}\, \varepsilon^{(i)}. \tag{26}$$

4. 将 $(x_{t^{(i)}}^{(i)}, t^{(i)})$ 输入网络（这是已知的，例如 U-Net），得到噪声预测

$$\hat{\varepsilon}_\theta^{(i)} = \varepsilon_\theta\big(x_{t^{(i)}}^{(i)}, t^{(i)}\big). \tag{27}$$

5. 计算 mini-batch 损失（MSE）

$$\mathcal{L}_B(\theta) = \frac{1}{B} \sum_{i=1}^{B} w_{t^{(i)}} \left\| \varepsilon^{(i)} - \hat{\varepsilon}_\theta^{(i)} \right\|^2. \tag{28}$$

6. 按梯度法更新参数

$$\theta \leftarrow \theta - \eta\, \nabla_\theta \mathcal{L}_B(\theta), \tag{29}$$

其中 $\eta > 0$ 为学习率，优化器可以是 SGD / Adam 等。

在训练过程中反复执行上述步骤，直到损失和生成质量收敛，得到训练好的参数 $\hat{\theta}$。此时，Step 2 的最大似然目标在 DDPM 框架下已通过 Step 4 的噪声预测训练近似实现。

Step 5: 测试算法

训练完成后，使用 $\hat{\theta}$ 从模型中生成样本或作为先验服务于下游任务。

无条件采样：从噪声生成数据

生成一条新样本 $\tilde{x}_0$ 的过程为：

1. 初始化

$$x_T \sim \mathcal{N}(0, I). \tag{30}$$

2. 对 $t = T, T-1, \dots, 1$ 依次执行：

(a) 计算噪声预测

$$\hat{\varepsilon}_t = \varepsilon_{\hat{\theta}}(x_t, t). \tag{31}$$

(b) 按式 (21) 计算反向均值

$$\mu_{\hat{\theta}}(x_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \hat{\varepsilon}_t \right). \tag{32}$$

(c) 若 $t > 1$，采样下一步

$$x_{t-1} \sim \mathcal{N}\big(\mu_{\hat{\theta}}(x_t, t), \sigma_t^2 I\big); \tag{33}$$

若 $t = 1$，令

$$x_0 = \mu_{\hat{\theta}}(x_1, 1). \tag{34}$$

3. 输出最终生成样本

$$\tilde{x}_0 = x_0. \tag{35}$$

此时 $\tilde{x}_0$ 就是从模型分布 $p_{\hat{\theta}}(x_0)$ 中采样得到的高维样本，例如一张新生成的图像。

Step6: 应用场景

在实际应用中，DDPM 及其变体可以用于：

1. **生成“像真的一样”的样本.** 例子：头像生成 / 换头像你给系统几张自己喜欢的风格图（动漫头像、油画风、ins 风自拍）。模型学到“人脸长什么样”“这种风格的颜色和纹理是什么分布”。然后从 $p_\theta(x)$ 里采样出一大堆从未在训练集中出现过、但“看起来就像真实自拍/插画”的新头像，让你挑着用。

2. **条件生成、修补、反演.** 例子：把游客照里的路人“干净地抹掉”你拍了一张风景照，前面有路人、垃圾桶。条件：给模型“原始图片 + 一个 mask（标出要抹掉的区域）”。模型基于学到的 p_θ(干净风景图)，在被遮住的区域补出合理的天空、海面、路面，而不是乱涂颜色。结果：看起来像你本来就拍到了一个没人、干净的场景。

3. **提供强先验（当作“数据分布的经验”来用）.** 例子：手机里把糊掉的夜景照片修清晰夜景拍糊了，噪声很多，细节看不清。相机 APP 背后有一个学好的 p_θ(自然场景照片)，知道“正常的夜景大概是什么纹理、轮廓和光斑分布”。在做“从模糊图恢复清晰图”这个反问题时，它把这个 p_θ 当先验：排除那些虽然数学上也能解释模糊图，但看起来很怪的解；优先选择“符合真实照片统计结构”的那一类解。用户感知就是：明明原图糊得一塌糊涂，点一下“增强”，出来的照片细节合理得多。

补充 1：为什么 max 似然等价于 max 噪声期望？

本节说明：为什么 Step 2 中的最大似然目标可以转化为 Step 4 中的“噪声预测误差”损失。

(1) 变分下界（ELBO） 对任意固定的前向分布 $q(x_{1:T} \mid x_0)$，有

$$\log p_\theta(x_0) = \log \int p_\theta(x_{0:T})\,\mathrm{d}x_{1:T} \tag{1}$$

$$= \log \int \frac{p_\theta(x_{0:T})}{q(x_{1:T} \mid x_0)} q(x_{1:T} \mid x_0)\,\mathrm{d}x_{1:T} \tag{2}$$

$$\geq \mathbb{E}_{q(x_{1:T}|x_0)}\big[\log p_\theta(x_{0:T}) - \log q(x_{1:T} \mid x_0)\big] \tag{3}$$

$$\equiv \mathcal{L}_{\mathrm{ELBO}}(x_0;\theta), \tag{4}$$

其中不等号来自 Jensen 不等式。对 $x_0 \sim p_{\mathrm{data}}$ 取期望，得到

$$\mathbb{E}_{p_{\mathrm{data}}}[\log p_\theta(x_0)] \;\geq\; \mathbb{E}_{p_{\mathrm{data}}}\big[\mathcal{L}_{\mathrm{ELBO}}(x_0;\theta)\big]. \tag{5}$$

因此，最大化对数似然可以通过最大化 ELBO 来实现。

(2) 用马尔可夫结构展开 ELBO 在 DDPM 中，

$$q(x_{1:T} \mid x_0) = \prod_{t=1}^{T} q(x_t \mid x_{t-1}), \quad p_\theta(x_{0:T}) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1} \mid x_t).$$

将这些代入 ELBO，可以把 $\mathcal{L}_{\mathrm{ELBO}}$ 写成若干条件 KL 项的和（省略与 θ 无关的常数）：

$$\mathcal{L}_{\mathrm{ELBO}}(x_0;\theta) = C - \sum_{t=2}^{T} \mathrm{KL}\Big(q(x_{t-1} \mid x_t, x_0) \,\Big\|\, p_\theta(x_{t-1} \mid x_t)\Big) - \mathrm{KL}\big(q(x_T \mid x_0) \,\|\, p(x_T)\big) + \mathbb{E}_q[\log p_\theta(x_0 \mid x_1)], \tag{6}$$

其中 C 为与 θ 无关的常数。

若进一步选取合适的方差参数化（例如固定 $p_\theta(x_{t-1} \mid x_t)$ 和 $p_\theta(x_0 \mid x_1)$ 的方差形式），可以使得

- 末端项 $\mathrm{KL}\big(q(x_T \mid x_0) \,\|\, p(x_T)\big)$ 和 $\mathbb{E}_q[\log p_\theta(x_0 \mid x_1)]$ 要么不依赖 θ，要么其依赖部分可以转化为简单项；
- 依赖 θ 的主要部分集中在

$$\sum_{t=2}^{T} \mathrm{KL}\Big(q(x_{t-1} \mid x_t, x_0) \,\Big\|\, p_\theta(x_{t-1} \mid x_t)\Big).$$

因此，最大化 ELBO 等价于最小化这些条件 KL 的期望。

(3) 线性高斯情形：KL 变成均值差平方 在 DDPM 的设定中，前向后验 $q(x_{t-1} \mid x_t, x_0)$ 与反向模型 $p_\theta(x_{t-1} \mid x_t)$ 都是高斯分布：

$$q(x_{t-1} \mid x_t, x_0) = \mathcal{N}\big(\mu_q(x_t, x_0, t),\, \tilde{\sigma}_t^2 I_d\big), \tag{7}$$

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}\big(\mu_\theta(x_t, t),\, \sigma_t^2 I_d\big), \tag{8}$$

其中 $\tilde{\sigma}_t^2$ 和 σ_t^2 仅依赖 t，I_d 为 $d \times d$ 单位矩阵。

若进一步令 $\sigma_t^2 = \tilde{\sigma}_t^2$（或视差异为常数），则两个高斯的 KL 可以写成

$$\mathrm{KL}\Big(q(x_{t-1} \mid x_t, x_0) \,\Big\|\, p_\theta(x_{t-1} \mid x_t)\Big) = \frac{1}{2\tilde{\sigma}_t^2}\big\|\mu_q(x_t, x_0, t) - \mu_\theta(x_t, t)\big\|^2 + \text{const}, \tag{9}$$

其中“const”不依赖 θ。

另一方面，$\mu_q(x_t, x_0, t)$ 可以用 x_0 和 x_t 的线性组合表示，而 x_t 又满足

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t}\, \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, I_d).$$

通过代入并整理，可将 $\mu_q(x_t, x_0, t)$ 与 $\mu_\theta(x_t, t)$ 的差写成关于 ε 与 $\varepsilon_\theta(x_t, t)$ 的差，最终得到

$$\mathrm{KL}\Big(q(x_{t-1} \mid x_t, x_0) \,\Big\|\, p_\theta(x_{t-1} \mid x_t)\Big) = \tilde{w}_t\, \mathbb{E}_{x_0, \varepsilon}\big[\|\varepsilon - \varepsilon_\theta(x_t, t)\|^2\big] + \text{const}, \tag{10}$$

其中 $\tilde{w}_t$ 为只依赖时间步 t 的权重。

(4) 汇总得到噪声预测损失 把所有 t 的 KL 项相加，并忽略与 θ 无关的常数，可以得到一个与下式等价的目标：

$$\mathcal{L}(\theta) = \mathbb{E}_{x_0, t, \varepsilon}\Big[w_t \,\|\varepsilon - \varepsilon_\theta(x_t, t)\|^2\Big], \tag{11}$$

其中 w_t 与上述 $\tilde{w}_t$ 对应，是仅依赖时间步的非负权重。

因此，在 DDPM 的线性高斯设定下，*Step 2* 中的最大似然 / 最小 *KL* 目标，可以通过变分下界和高斯 *KL* 的解析形式，转化为最小化一个“噪声预测误差”的期望损失。也就是说，只要通过随机梯度下降最小化噪声回归损失 $\mathcal{L}(\theta)$，就等价于在该模型类下逼近原始的最大似然目标。

根据 Step 2 的最大似然目标，可以推导出一个变分下界（ELBO），其中依赖 θ 的主要项可以写成若干条件 KL 的和。进一步推导可得：只要最小化下面这个“噪声预测误差”的期望，就等价于在该框架下最大化似然：

$$\mathcal{L}(\theta) = \mathbb{E}_{x_0, t, \varepsilon}\Big[w_t \,\|\varepsilon - \varepsilon_\theta(x_t, t)\|^2\Big]. \tag{23}$$

其中：

- $x_0 \sim p_{\text{data}}(x_0)$；
- $t \sim \text{Uniform}\{1, \ldots, T\}$；
- $\varepsilon \sim \mathcal{N}(0, I)$；
- x_t 按式 (12) 由 (x_0, t, ε) 构造；
- $w_t \geq 0$ 为时间步权重，实践中常取 $w_t = 1$ 或简单函数。

补充 2：DDPM 的直觉

1. **直接建模复杂的 $p_{\text{data}}(x)$ 太难**。

   高维图像分布又多峰又奇怪，直接去学 $p_\theta(x)$ 或直接从噪声“一步生图”通常很不稳定（GAN 易崩、flow 结构复杂、VAE 表达能力有限）。

2. **先把“好分布”变成“简单分布”，再学反过来**。

   想法是构造一个完全可控的“加噪过程”

   $$x_0 \to x_1 \to \cdots \to x_T \approx \mathcal{N}(0, I),$$

   把真实数据一步步「加噪、模糊、洗白」成高斯噪声。这个正向过程 q 完全由我们设计，分布形式都能写出来，也能直接采样。

   真正需要学习的是**反向路径**：

   $$x_T \to x_{T-1} \to \cdots \to x_0,$$

   也就是“从噪声慢慢、一小步一小步去噪回到数据”的过程。

3. **每一步“去一点噪”是简单稳定的，于是把全局生成问题拆成很多小的去噪问题**。

   在步长足够小时，每个反向步 $p_\theta(x_{t-1} \mid x_t)$ 都近似高斯，只要知道一点“局部梯度 / 噪声”信息就能写出均值。

   于是用一个神经网络在不同噪声等级 t 上，做一个非常普通的回归任务：给定 x_t，预测它里面的噪声 ε。

   这样：

   - **训练时：**从干净图 x_0 人为加噪生成 x_t，监督信号 ε 完全已知 $\Rightarrow$ 普通 MSE 回归，训练稳定；
   - **采样时：**从高斯噪声 x_T 出发，按网络预测的一步步去噪，沿整条反向链走回 x_0，得到新图像。

补充 3：DDPM 中的 deep learning 部分

DDPM 里真正 *deep learning* 的部分，就是噪声预测网络 $\varepsilon_\theta(x_t, t)$（以及其可能顺带预测方差的分支）；其他部分主要是概率建模和手工设计的过程。

1. 不属于 deep learning 的部分（“框架”）

- 设计前向扩散：

$$q(x_t \mid x_{t-1}) = \mathcal{N}\big(\sqrt{\alpha_t}\, x_{t-1}, (1-\alpha_t)I\big),$$

  以及其闭式形式 $q(x_t \mid x_0)$。

- 设计反向为高斯链：

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}\big(\mu_\theta(x_t, t), \Sigma_t\big).$$

- 使用 ELBO / KL 将最大似然目标改写成若干条件 KL 项，再推导成噪声 MSE 损失。

这些属于**概率图模型 + 变分推断**的理论设计层面，本身并不依赖具体的深度网络结构。

2. 属于 deep learning 的部分（“可学习的函数”）

- **噪声预测网络** $\varepsilon_\theta(x_t, t)$。

  具体实现通常是 U-Net / CNN / Transformer / GNN 等大规模深度网络：

  - 输入：带噪样本 x_t（张量）加上时间步 t 的嵌入向量；
  - 输出：与 x_t 同形状的噪声估计 $\hat{\varepsilon}$（有时再多一个分支预测 σ_t）。

- **训练方式：**使用 MSE 损失

$$\mathbb{E}\big[\|\varepsilon - \varepsilon_\theta(x_t, t)\|^2\big],$$

  再配合 SGD / Adam 等优化器做反向传播，这就是标准的 deep learning 训练流程。