# Diffusion Model

J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” Adv. Neur. In., vol. 33, pp. 6840–6851, 2020.

# 为什么扩散模型？

# 生成模型的“三难困境”

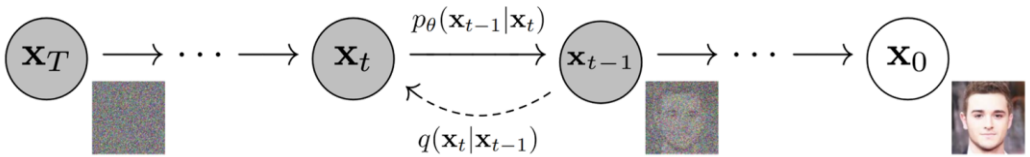
- 变分自编码器 (VAE):
  - 训练稳定，潜空间有意义，能显式计算似然下界
  - 生成的图像普遍模糊，难以达到逼真效果
- 生成对抗网络 (GAN)
  - 能够生成非常清晰、锐利的图像
  - 训练不稳定，易发生模式坍塌 (Mode Collapse)，难以优化
- 能否设计一个模型，既能稳定地训练，又能生成高质量、多样化的样本？
  - 三个期望目标

- 扩散模型 提出了一种全新的范式。它不直接学习从简单噪声到复杂图像的一步映射，而是学习一个可控的、多步的、逐步去噪的过程
  - 提升了样本质量 (Sample Quality) 和训练稳定性 (Training Stability)

# 预测风雨的“数字雕塑家”

# 类比：能预测风雨的“数字雕塑家”

- 原始数据（ $x_0$ ）：一座精美绝伦的数字雕塑（例如，一张清晰的猫的照片）
- 扩散过程（Forward / Diffusion Process），对于每个雕塑
  - 每个时间步（timestep），雕塑被轻微地风雨腐蚀（加入少量高斯噪声）
  - 经过若干时间步（如$T = 1000$），雕塑最终变成充满噪声的混沌石块（$x_T$）
  - 要点：“腐蚀”过程是固定的、已知的、简单的。“腐蚀”的强度（噪声大小）也是已知的
- 逆扩散过程（Reverse / Denoising Process）
  - 训练一位技艺高超的 " 数字雕塑家 " （一个神经网络，通常是U-Net）
  - 输入：任意的时间戳 $t$，被腐蚀了 $t$ 步的石块（ $x_t$ ）
  - 训练目标：雕塑家能预测导致$x_t$的风雨腐蚀（然后，可将石块恢复到状态$x_{t-1}$ ）



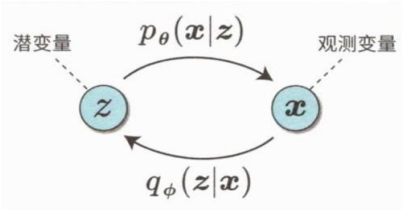
# 如何训练风雨预测能力？

- “数字雕塑家” 风雨预测能力训练的技术路线
  - 对每一雕塑，逐步风雨以生成数据集（$(t, x_t, \epsilon_t)$数据对），用于监督风雨预测网络训练
  - 预测网络洞察的这一批雕塑（训练集）的，各个时间步的风雨腐蚀，得到
    - 噪声预测神经网络：$\boldsymbol{\epsilon_\theta}(\mathbf{x}_t, t)$
- 对于每个时间步$t$
  - 扩散过程
    - 雕塑$x_{t-1}$被轻微地风雨腐蚀（加入少量高斯噪声）成石块$x_t$
  - 逆扩散过程
    - 神经网络雕塑家预测导致$x_t$的风雨腐蚀（噪声）
- 扩散模型的核心
  - 训练：训练一个能精确预测风雨的神经网络
  - 生成：从 $t = T$ 开始，将混沌石块，一步步预测逆转风雨，雕刻出一座全新的雕塑

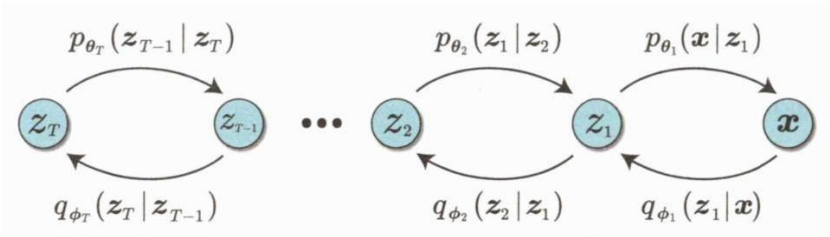
# VAE到扩散模型

# VAE与分层的VAE

- VAE 是一个拥有潜变量的模型
  - $p_\theta(\mathbf{x} \mid \mathbf{z})$ 是从潜变量z到观测变量x的概率分布,我们使用解码器网络建模。
  - $q_\phi(\mathbf{z} \mid \mathbf{x})$ 是从观测变量 $\mathbf{x}$ 到潜变量 $z$ 的概率分布,我们使用编码器网络进行建模

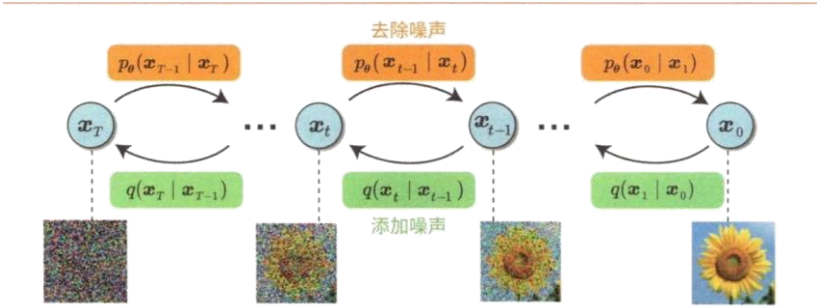



- 分层 VAE
  - 假定马尔可夫性,可以防止参数增加
  - 图中一共有 $2T$ 个概率分布。如果简单通过神经网络进行建模,需要 $T$ 个解码器的神经网络和 $T$ 个编码器的神经网络,即一共需要 $2T$ 个神经网络。随着 $T$ 的增大,这种实现方式越来越困难



# 进入扩散模型

- 改进
  - 使观测变量和潜变量的维度相同
  - 编码器添加基于固定的正态分布采样的噪声
- 添加噪声的过程 $q$ 只是简单地添加固定的高斯噪声，不需要参数 $\phi$
  - 类似于将牛奶倒入咖啡中，牛奶逐渐“扩散” 至整杯咖啡
    - 第 $t$ 个噪声数据 $x_t$ 称为时刻 $t$ 的噪声数据
- 扩散模型使用神经网络对去除噪声的处理进行建模
  - $\mathbf{x}_0$ 是观测变量，其余的 $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T$ 为潜变量

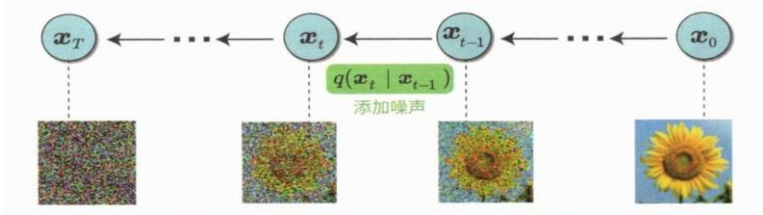


# 扩散过程和逆扩散过程的计算

# 扩散过程

- 扩散过程：固定参数，无需学习， 添加噪声，共$T$步
  - $q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I}\right)$
    - 均值为 $\sqrt{1-\beta_t}\mathbf{x}_{t-1}$，协方差矩阵为 $\beta_t \mathbf{I}$，正态分布。
    - $t$ 为 $1 \leq t \leq T$ 的整数。 $\beta_t$ 是预设值。 $\beta_t$越大， 添加的噪声就越大，$\{\beta_1, \beta_2, \cdots, \beta_T\}$ 为 “噪声调度”参数
    - 如果 $T$足够大 (如 1000)，且噪声调度调整到一定的范围，那么可以得到 $p(\mathbf{x}_T) \approx \mathcal{N}(\mathbf{x}_T; \mathbf{0}, \mathbf{I})$
    - 重参数化技巧计算：

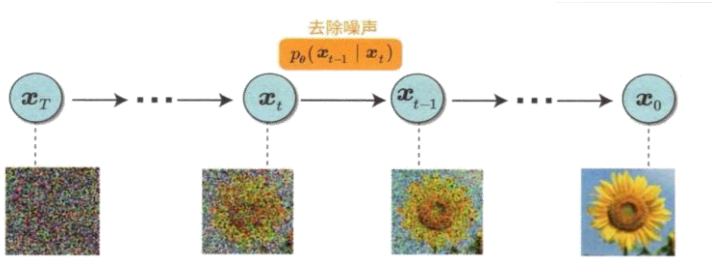
$$\varepsilon \sim N(\varepsilon; \mathbf{0}, \mathbf{I})$$
$$\mathbf{x}_t = \sqrt{1-\beta_t}\mathbf{x}_{t-1} + \sqrt{\beta_t}\varepsilon$$

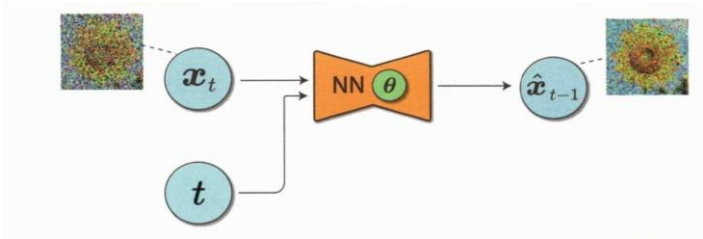


# 逆扩散过程

- 逆扩散过程是去除噪声的处理 。这个去除噪声的过程通过神经网络进行
- 使用单个神经网络对每个时刻数据进行噪声去除处理
- $\hat{\mathbf{x}}_{t-1} = \mathrm{NN}(\mathbf{x}_t, t; \boldsymbol{\theta})$

  $p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \hat{\mathbf{x}}_{t-1}, \mathbf{I})$
- 训练时，只关注每一次的噪声预测，并没有后续的处理
- 推理时，噪声去除后，直接进入下一个阶段
  - 重参数化技巧实现：$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \hat{\mathbf{x}}_{t-1}, \mathbf{I})$



# ELBO的近似计算

# 扩散模型的 ELBO

- 【VAE的ELBO】：$\mathrm{ELBO}(\mathbf{x};\boldsymbol{\theta},\phi)=\int q_{\phi}(\mathbf{z}\mid\mathbf{x})\log\frac{p_{\theta}(\mathbf{x},\mathbf{z})}{q_{\phi}(\mathbf{z}|\mathbf{x})}\mathrm{d}\mathbf{z}=\mathbb{E}_{q_{\phi}(z|x)}\left[\log\frac{p_{\theta}(x,z)}{q_{\phi}(z|x)}\right]$
- ELBO在扩散模型上推广
  - 将 $x$ 变更为 $x_0$
  - 将 $z$ 变更为 $x_1,x_2,\cdots,x_T$
  - 去除编码器参数 $\phi$ 【加噪是固定过程，没有参数】
- $\mathrm{ELBO}(\mathbf{x}_0;\boldsymbol{\theta})=\mathbb{E}_{q(\mathbf{x}_1,\mathbf{x}_2,\cdots,\mathbf{x}_T|\mathbf{x}_0)}\left[\log\frac{p_{\boldsymbol{\theta}}(\mathbf{x}_0,\mathbf{x}_1,\cdots,\mathbf{x}_T)}{q(\mathbf{x}_1,\mathbf{x}_2,\cdots,\mathbf{x}_T|\mathbf{x}_0)}\right]$
  - 记$\mathbf{x}_{0:T}=\mathbf{x}_0,\mathbf{x}_1,\cdots,\mathbf{x}_T$，扩散模型的 ELBO

$$\mathrm{ELBO}(\mathbf{x}_0;\boldsymbol{\theta})=\mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\left[\log\frac{p_{\theta}(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}\mid\mathbf{x}_0)}\right]$$

# 扩散模型的 ELBO

- 【扩散模型ELBO】$\mathrm{ELBO}(\mathbf{x}_0;\boldsymbol{\theta}) = \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\left[\log\frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\right]$
  - $p_\theta(\mathbf{x}_{0:T}) = p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1)p_\theta(\mathbf{x}_1 \mid \mathbf{x}_2)\ldots p_\theta(\mathbf{x}_{T-1} \mid \mathbf{x}_T)p(\mathbf{x}_T) = p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)$
    - $p(\mathbf{x}_T)$ 表示完全高斯噪声 $\mathcal{N}(\mathbf{x}_T;\mathbf{0},\mathbf{I})$; 马尔可夫性
  - $q(\mathbf{x}_{1:T} \mid \mathbf{x}_0) = \prod_{t=1}^{T} q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right)$

$$\mathrm{ELBO}(\mathbf{x}_0;\boldsymbol{\theta}) = \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\left[\log\frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T} \mid \mathbf{x}_0)}\right]$$

$$= \mathbb{E}_{q(\mathbf{x}_{t,T}|\mathbf{x}_0)}\left[\log\frac{p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)}{\prod_{t=1}^{T} q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right)}\right]$$

$$= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\left(\log\prod_{\mathrm{t}=1}^{\mathrm{T}} p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) + \log\frac{p(\mathbf{x}_T)}{\prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1})}\right)$$ 【第二项与$\theta$无关】

移除扩散模型ELBO中不含$\boldsymbol{\theta}$部分，剩下部分为 $J(\boldsymbol{\theta})$，于是优化目标为

$$J(\boldsymbol{\theta}) = \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\left[\log\prod_{\mathrm{t}=1}^{\mathrm{T}} p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)\right] = \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}\left[\sum_{\mathrm{t}=1}^{\mathrm{T}} \log p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)\right]$$

# 一次计算T个采样

采样数时针对样本来说的

# 优化目标$J(\theta)$的近似计算

- 目标函数 $J(\boldsymbol{\theta}) = \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}[\sum_{t=1}^{T} \log p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)]$
- 期望的蒙特卡洛近似，假设样本量为1，则 $J(\boldsymbol{\theta})$的计算（期望计算最自然的近似）
- $\mathbf{x}_{1:T} \sim q(\mathbf{x}_{1:T} \mid \mathbf{x}_0)$，$T$次采样
  - 扩散过程基于原始数据中的 $\mathbf{x}_0$，生成 $\mathbf{x}_{1:T}$（$\mathbf{x}_{1:T}$ 共有 $T$ 个变量）
  - 利用 $\mathbf{x}_{1:T}$ 计算每个时刻的 $\log p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$
    - $\hat{\mathbf{x}}_{t-1} = \text{NeuralNet}(\mathbf{x}_t, t; \boldsymbol{\theta})$
    - $p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \hat{\mathbf{x}}_{t-1}, \mathbf{I})$

$$J(\boldsymbol{\theta}) \approx \sum_{t=1}^{T} \log p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \sum_{t=1}^{T} \log \mathcal{N}(\mathbf{x}_{t-1}; \hat{\mathbf{x}}_{t-1}, \mathbf{I}) = \sum_{t=0}^{T-1} \log \mathcal{N}(\mathbf{x}_t; \hat{\mathbf{x}}_t, \mathbf{I})$$

$$= \sum_{t=0}^{T-1} \log \frac{1}{\sqrt{(2\pi)^D |\mathbf{I}|}} \exp\left\{-\frac{1}{2}(\mathbf{x}_t - \hat{\mathbf{x}}_t)^\top \mathbf{I}^{-1}(\mathbf{x}_t - \hat{\mathbf{x}}_t)\right\} = \sum_{t=0}^{T-1} \left(-\frac{1}{2}(\mathbf{x}_t - \hat{\mathbf{x}}_t)^\top (\mathbf{x}_t - \hat{\mathbf{x}}_t) + \log \frac{1}{\sqrt{(2\pi)^D}}\right)$$

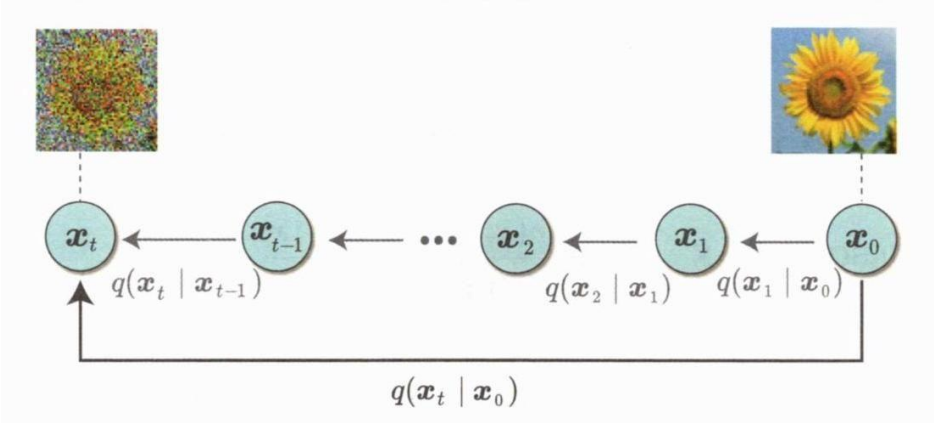
$$= -\frac{1}{2}\sum_{t=0}^{T-1}(\mathbf{x}_t - \hat{\mathbf{x}}_t)^\top (\mathbf{x}_t - \hat{\mathbf{x}}_t) + T \log \frac{1}{\sqrt{(2\pi)^D}}$$，由于最后一个常数项可以忽略，于是

$$J(\boldsymbol{\theta}) \approx -\frac{1}{2}\sum_{t=0}^{T-1}(\mathbf{x}_t - \hat{\mathbf{x}}_t)^\top (\mathbf{x}_t - \hat{\mathbf{x}}_t) = -\frac{1}{2}\sum_{t=0}^{T-1}\|\mathbf{x}_t - \hat{\mathbf{x}}_t\|^2$$

# 2个采样

# 思路

- $q(\mathbf{x}_t \mid \mathbf{x}_0)$ 有解析表达
  - 只需在原始数据 $\mathbf{x}_0$ 中添加一次噪声，就可以对任意时刻 $t$ 的 $\mathbf{x}_t$ 进行采样



# $q(\mathbf{x}_t \mid \mathbf{x}_0)$ 的表达式推导

- $q(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}\right)$
  - 其中，$\alpha_t = 1 - \beta_t$，$\bar{\alpha}_t = \alpha_t \alpha_{t-1} \cdots \alpha_1$
  - $\beta_t$ 是用户设定的值
- 证明
  - $x \sim \mathcal{N}(\mu_x, \sigma_x^2)$，$y \sim \mathcal{N}\left(\mu_y, \sigma_y^2\right)$，$z = x + y \sim \mathcal{N}\left(\mu_x + \mu_y, \sigma_x^2 + \sigma_y^2\right)$
  - $q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}\right)$
    - 令$\alpha_t = 1 - \beta_t$，则$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{\alpha_t}\mathbf{x}_{t-1}, (1-\alpha_t)\mathbf{I}\right)$
    - $\varepsilon_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$，$\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\varepsilon_t$；$\boldsymbol{\varepsilon}_{t-1} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$，$\mathbf{x}_{t-1} = \sqrt{\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1-\alpha_{t-1}}\boldsymbol{\varepsilon}_{t-1}$
- 于是
  - $\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\boldsymbol{\varepsilon}_t = \sqrt{\alpha_t}\left(\sqrt{\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1-\alpha_{t-1}}\boldsymbol{\varepsilon}_{t-1}\right) + \sqrt{1-\alpha_t}\boldsymbol{\varepsilon}_t = \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{\alpha_t - \alpha_t\alpha_{t-1}}\boldsymbol{\varepsilon}_{t-1} + \sqrt{1-\alpha_t}\boldsymbol{\varepsilon}_t$
  - $\mathbf{x}_t = \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\varepsilon$，继续，$\mathbf{x}_t = \sqrt{\alpha_t\alpha_{t-1}\cdots\alpha_1}\mathbf{x}_0 + \sqrt{1-\alpha_t\alpha_{t-1}\cdots\alpha_1}\boldsymbol{\varepsilon} = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\varepsilon}$

# 2次数据采样的蒙特卡洛

➢目标函数 $J(\boldsymbol{\theta}) = \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}[\sum_{t=1}^{T} \log p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)]$

$J(\boldsymbol{\theta}) = \sum_{t=1}^{T} \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} [\log p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)]$【1: $\mathbb{E}_{p(x,y)}[x+y] = \mathbb{E}_{p(x)}[x] + \mathbb{E}_{p(y)}[y]$】

$= \sum_{t=1}^{T} \mathbb{E}_{q(x_{t-1},x_t|x_0)} [\log p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)]$【2: $\mathbb{E}_{p(x,y)}[f(x)] = \mathbb{E}_{p(x)}[f(x)]$】

$J(\boldsymbol{\theta}) = \sum_{t=1}^{T} \mathbb{E}_{q(x_{t-1},x_t|x_0)}[\log p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)] = T\mathbb{E}_{u(t)}\left[\mathbb{E}_{q(\mathbf{x}_{t-1},\mathbf{x}_t|\mathbf{x}_0)}[\log p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)]\right]$

蒙特卡洛采样【时间T次累加转为1次时间采样，然后$q(\mathbf{x}_{t-1}, \mathbf{x}_t \mid \mathbf{x}_0)$进行2次数据采样】

$t \sim U\{1, T\}$

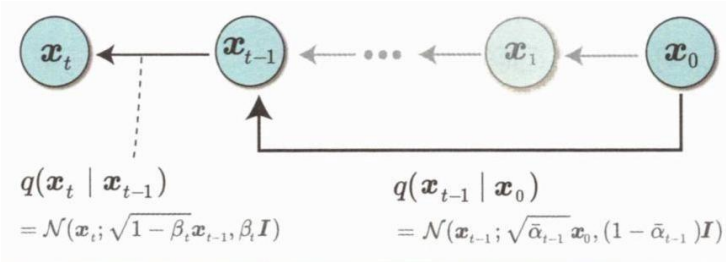
$\mathbf{x}_{t-1} \sim q(\mathbf{x}_{t-1} \mid \mathbf{x}_0)$

$\mathbf{x}_t \sim q(\mathbf{x}_t \mid \mathbf{x}_{t-1})$

$J(\theta) \approx T \log p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) \approx -\frac{T}{2}\|\mathbf{x}_{t-1} - \hat{\mathbf{x}}_{t-1}\|^2$，其中

$\hat{\mathbf{x}}_{t-1} = \text{NeuralNet}(\mathbf{x}_t, t; \theta)$

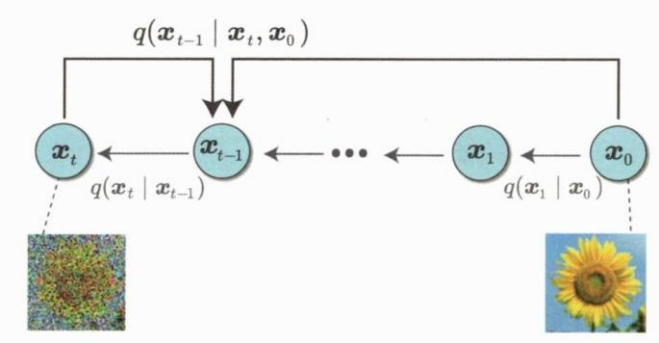
$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \hat{\mathbf{x}}_{t-1}, \mathbf{I})$



# 一次计算1个采样

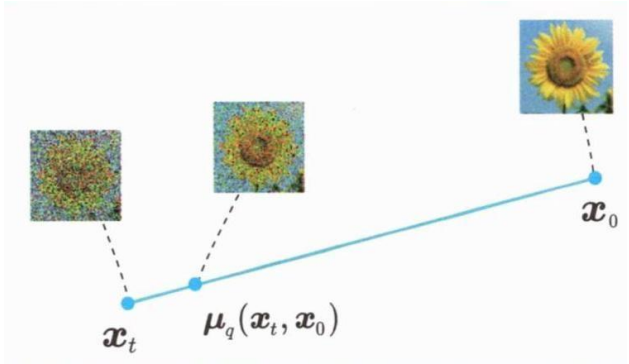
# 思路

- $q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0)$ 表示给定 $\mathbf{x}_0$ 和 $\mathbf{x}_t$ 时的 $\mathbf{x}_{t-1}$ 的概率，一次计算即可完成



# $q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0)$的计算

- $q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}\left(\mathbf{x}_{t-1}; \boldsymbol{\mu}_q(\mathbf{x}_t, \mathbf{x}_0), \sigma_q^2(t)\mathbf{I}\right)$
  - $\alpha_t = 1 - \beta_t, \bar{\alpha}_t = \alpha_t \alpha_{t-1} \cdots \alpha_1$
  - $\boldsymbol{\mu}_q(\mathbf{x}_t, \mathbf{x}_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\mathbf{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\mathbf{x}_0}{1-\bar{\alpha}_t}$, $\sigma_q^2(t) = \frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}$
- 可以将该线性和，解释为两个点的内分点



# 1次采样的$J(\theta)$优化

【目标函数】$J(\boldsymbol{\theta}) = T\mathbb{E}_{u(t)}\left[\mathbb{E}_{q(\mathbf{x}_{t-1},\mathbf{x}_t|\mathbf{x}_0)}[\log p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)]\right]$

令$J_0 = \mathbb{E}_{q(\mathbf{x}_{t-1},\mathbf{x}_t|\mathbf{x}_0)}[\log p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)]$，于是，$J(\boldsymbol{\theta}) = T\mathbb{E}_{u(t)}[J_0]$

令$J_1 = \mathbb{E}_{q(\mathbf{x}_{t-1},\mathbf{x}_t|\mathbf{x}_0)}\left[\log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)}{q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0)}\right] = J_0 - \mathbb{E}_{q(\mathbf{x}_{t-1},\mathbf{x}_t|\mathbf{x}_0)}[\log q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0)]$,

于是，优化子目标为：$J_1 = -\mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)}\left[D_{\mathrm{KL}}\left(q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)\right)\right]$

又，$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \mathcal{N}\left(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \sigma_q^2(t)\mathbf{I}\right)$，其中，$\hat{\mathbf{x}}_{t-1} = \mathrm{NeuralNet}(\mathbf{x}_t, t; \boldsymbol{\theta}) = \boldsymbol{\mu}_\theta(\mathbf{x}_t, t)$,

且，$q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}\left(\mathbf{x}_{t-1}; \boldsymbol{\mu}_q(\mathbf{x}_t, \mathbf{x}_0), \sigma_q^2(t)\mathbf{I}\right)$

于是，$D_{\mathrm{KL}}\left(q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)\right) = \frac{1}{2\sigma_q^2(t)}\left\|\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) - \boldsymbol{\mu}_q(\mathbf{x}_t, \mathbf{x}_0)\right\|^2$,

因此，定义$\mathrm{LOSS}(\mathbf{x}_0; \boldsymbol{\theta}) = \mathbb{E}_{u(t)}\left[\mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)}\left[\frac{1}{\sigma_q^2(t)}\left\|\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) - \boldsymbol{\mu}_q(\mathbf{x}_t, \mathbf{x}_0)\right\|^2\right]\right]$，其计算

$t \sim U\{1, T\}$

$\mathbf{x}_t \sim q(\mathbf{x}_t \mid \mathbf{x}_0)$

$$\mathrm{LOSS}(\mathbf{x}_0; \boldsymbol{\theta}) = \frac{1}{\sigma_q^2(t)}\left\|\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) - \boldsymbol{\mu}_q(\mathbf{x}_t, \mathbf{x}_0)\right\|^2$$

# 扩散模型的训练

# 1次采样的扩散模型训练

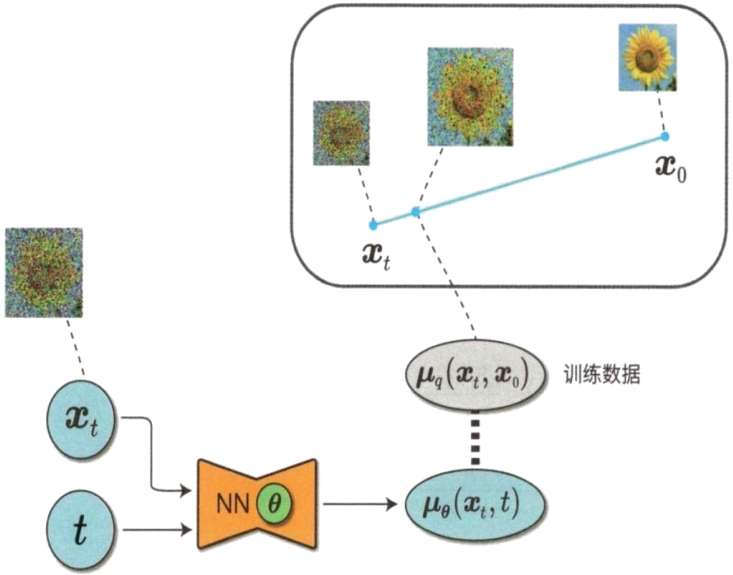
$\boldsymbol{\mu}_q(\mathbf{x}_t, \mathbf{x}_0)$ 作为训练数据

# 扩散模型的训练算法

- Repeat:
  - 1. 从训练数据中随机获取 $\mathbf{x}_0$
  - 2. $t \sim U\{1, T\}$ (基于均匀分布生成整数 $t$ )
  - 3. $\varepsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  - 4. $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\varepsilon}$ (基于 $q(\mathbf{x}_t \mid \mathbf{x}_0)$ 采样)
  - 5. $\boldsymbol{\mu}_q(\mathbf{x}_t, \mathbf{x}_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\mathbf{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\mathbf{x}_0}{1-\bar{\alpha}_t}$
  - 6. $\sigma_q^2(t) = \frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}$
  - 7. $\mathrm{LOSS}(\mathbf{x}_0; \boldsymbol{\theta}) = \frac{1}{\sigma_q^2(t)} \left\| \boldsymbol{\mu}_\theta(\mathbf{x}_t, t) - \boldsymbol{\mu}_q(\mathbf{x}_t, \mathbf{x}_0) \right\|^2$
  - 8. 计算 $\frac{\partial}{\partial \boldsymbol{\theta}} \mathrm{LOSS}(\mathbf{x}_0; \boldsymbol{\theta})$ ,并通过梯度法更新参数

# 以 $\boldsymbol{\mu}_q(\mathbf{x}_t, \mathbf{x}_0)$ 作为训练数据的扩散模型

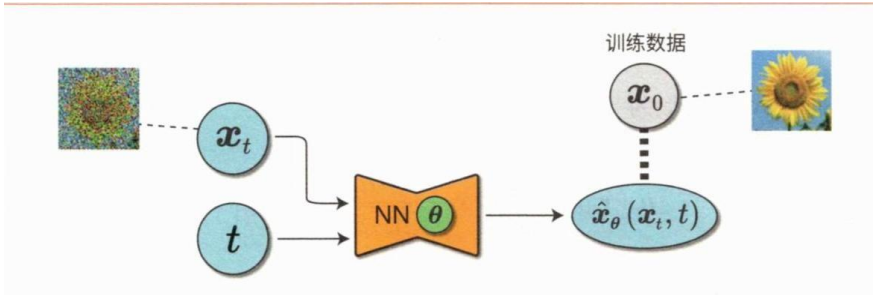
- 将训练数据 $\boldsymbol{\mu}_q(\mathbf{x}_t, \mathbf{x}_0)$ 解释为 $\mathbf{x}_t$ 和 $\mathbf{x}_0$ 的 “内分点”
  - $\boldsymbol{\mu}_q(\mathbf{x}_t, \mathbf{x}_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\mathbf{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\mathbf{x}_0}{1-\bar{\alpha}_t}$
- 神经网络
  - $\boldsymbol{\mu}_\theta(\mathbf{x}_t, t)$
- 损失函数
  - $\mathrm{LOSS}(\mathbf{x}_0; \boldsymbol{\theta}) = \frac{1}{\sigma_q^2(t)} \|\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) - \boldsymbol{\mu}_q(\mathbf{x}_t, \mathbf{x}_0)\|^2$



# 恢复原始数据的神经网络

# 恢复原始数据的神经网络

- $\boldsymbol{\mu}_q(\mathbf{x}_t, \mathbf{x}_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\mathbf{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\mathbf{x}_0}{1-\bar{\alpha}_t}$
- 神经网络调整类似形式
  - $\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\mathbf{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\hat{x}_\theta(\mathbf{x}_t, t)}{1-\bar{\alpha}_t}$
- 神经网络：$\hat{\mathbf{x}}_\theta(\mathbf{x}_t, t)$ 推断的值
  - 损失函数
  - $D_{\mathrm{KL}}\left(q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)\right) = \frac{1}{2\sigma_q^2(t)} \left\| \boldsymbol{\mu}_\theta(\mathbf{x}_t, t) - \boldsymbol{\mu}_q(\mathbf{x}_t, \mathbf{x}_0) \right\|^2$
- $= \frac{1}{2\sigma_q^2(t)} \left( \frac{\sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)}{1-\bar{\alpha}_t} \right)^2 \left\| \hat{\mathbf{x}}_\theta(\mathbf{x}_t, t) - \mathbf{x}_0 \right\|^2$
- 神经网络以 $\mathbf{x}_0$ 作为训练数据进行训练,使得输出的 $\hat{\mathbf{x}}_\theta(\mathbf{x}_t, t)$ 与 $\mathbf{x}_0$ 相同



# 预测噪声的神经网络

# 预测噪声的神经网络

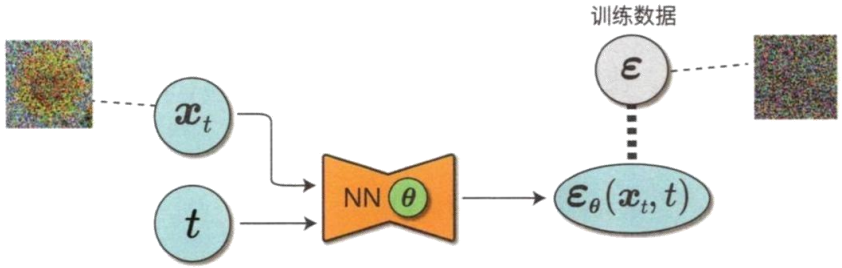
- $q(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}\right)$开始，$\varepsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$，$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\varepsilon$，

  因此，$\mathbf{x}_0 = \frac{\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\varepsilon}}{\sqrt{\bar{\alpha}_t}}$，$\boldsymbol{\mu}_q(\mathbf{x}_t, \mathbf{x}_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\mathbf{x}_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\mathbf{x}_0}{1-\bar{\alpha}_t} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\varepsilon\right)$

  - 将 $\mu_\theta(x_t, t)$ 变形为与该式类似形式，$\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\varepsilon_\theta(\mathbf{x}_t, t)\right)$

- 损失函数$D_{\mathrm{KL}}\left(q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)\right)$

- $= \frac{1}{2\sigma_q^2(t)}\left\|\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) - \boldsymbol{\mu}_q(\mathbf{x}_t, \mathbf{x}_0)\right\|^2 = \frac{1}{2\sigma_q^2(t)}\frac{(1-\alpha_t)^2}{(1-\bar{\alpha}_t)\alpha_t}\left\|\varepsilon_\theta(\mathbf{x}_t, t) - \varepsilon\right\|^2$



# 新数据的采样

# 新数据的采样

扩散模型的训练算法 (噪声预测版本)

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

2: for $t$ in $[T, \cdots, 1]$ :

3: $\varepsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

4: if $t = 1$ then $\varepsilon = 0$

5: $\sigma_q(t) = \sqrt{\frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}}$

6: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\varepsilon_\theta(\mathbf{x}_t, t)\right) + \sigma_q(t)\varepsilon$

7: return $x_0$

# 核心困惑：为何需要多步“预估-修正”？

- 我们为什么不直接从 $x_T$ 一步到位生成 $x_0$？为何要费力地走 $T$ 步？
  - $x_T$ $\xrightarrow{\text{神经网络一步预测}}$ $\hat{x}_0$ (为什么不行？)
- 神经网络的预测，在噪声很大时是极其不准确
  - 预估
    - 在噪声很大时（例如 $t$ 接近 $T$），$x_t$ 中几乎不包含任何关于 $x_0$ 的有效信息。此时让网络直接预测 $x_0$，其结果（我们称之为 $\hat{x}_0$）会非常粗糙、模糊，仅仅是一个“大概的轮廓”或“方向性的猜测”
    - 这个 $\hat{x}_0$ 本身质量很差，不能作为最终结果
  - 修正
    - 粗糙的“预估” $\hat{x}_0$为我们如何从当前的 $x_t$ 迈向更清晰一点的 $x_{t-1}$ 提供了至关重要的指引
    - 利用这个粗糙的 $\hat{x}_0$，代入之前推导出的后验分布 $q(x_{t-1}|x_t, x_0)$ 的公式中，计算出一个更可靠的 $x_{t-1}$
    - 本质：不完全相信神经网络一步到位的预测，只相信它给出的“一小步”方向
      - 我们只利用它的预测结果，来稳健地将状态从 $t$ 推进到 $t-1$。

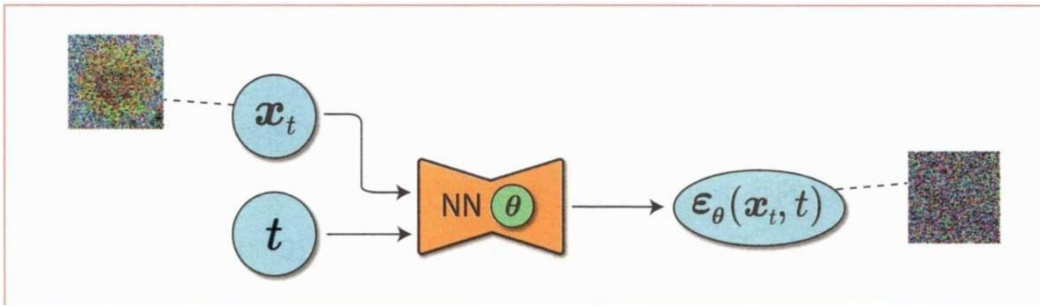
# 核心困惑：为何需要多步“预估-修正”？

- 扩散模型的生成过程，是一个典型的“预估-修正”迭代过程
- 它通过 $T$ 次微小、稳健的“预估-修正”循环，将一个极其困难的“一步生成”问题，分解为 $T$ 个更简单的“单步去噪”问题，从而在保证稳定性的前提下，逐步累积细节，最终生成高质量的图像

# 扩散模型的实现

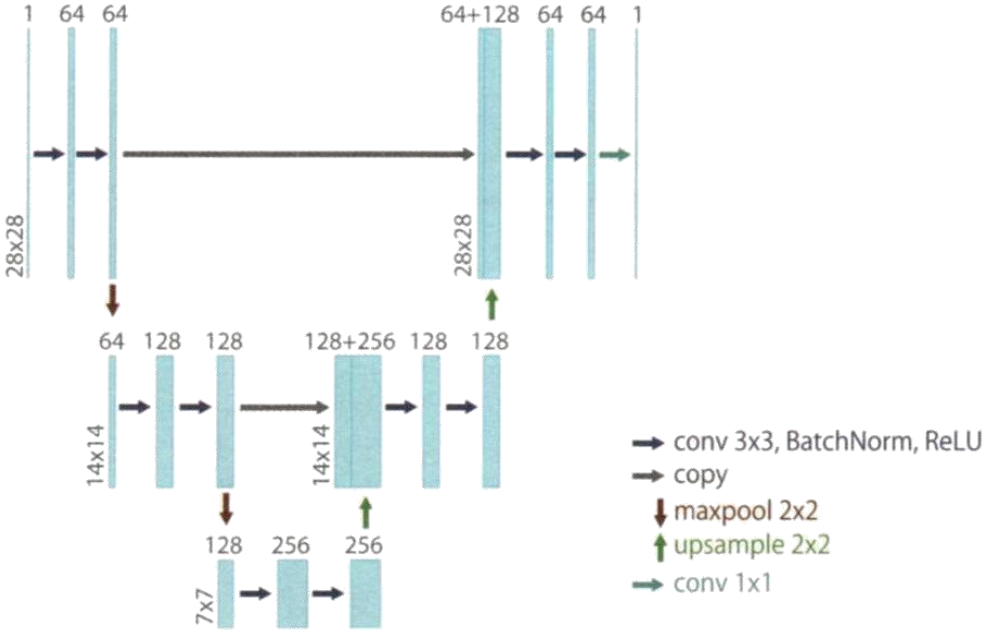
# U-Net

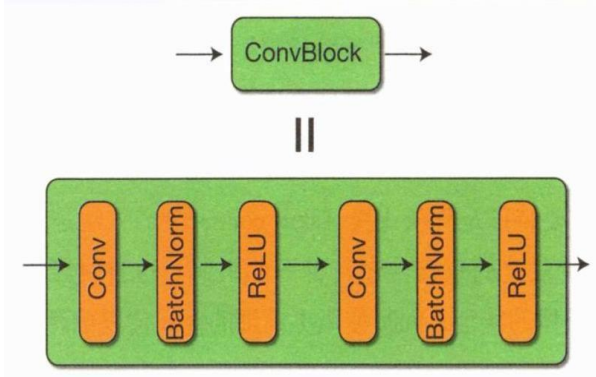
- 噪声预测网络$\varepsilon_\theta(\mathbf{x}_t, t)$



# U-Net 的网络结构

- 基本结构ConvBlock

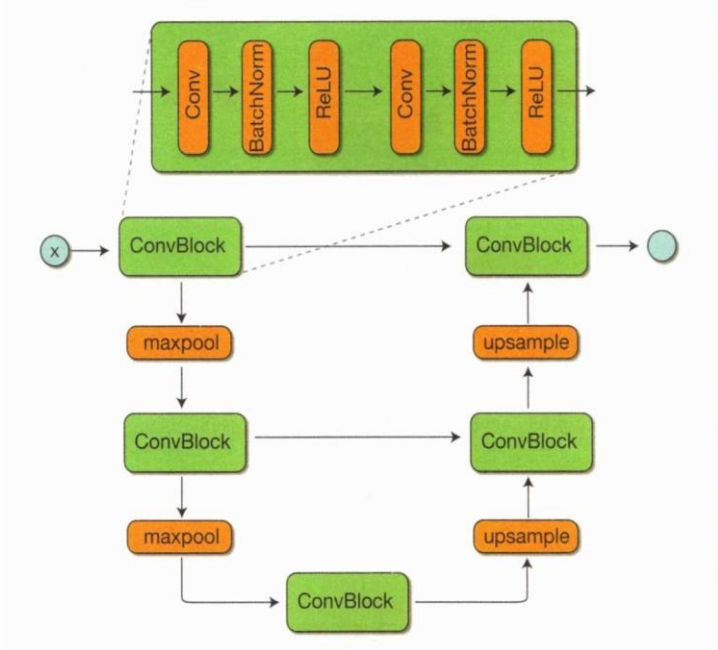


# 正弦位置编码

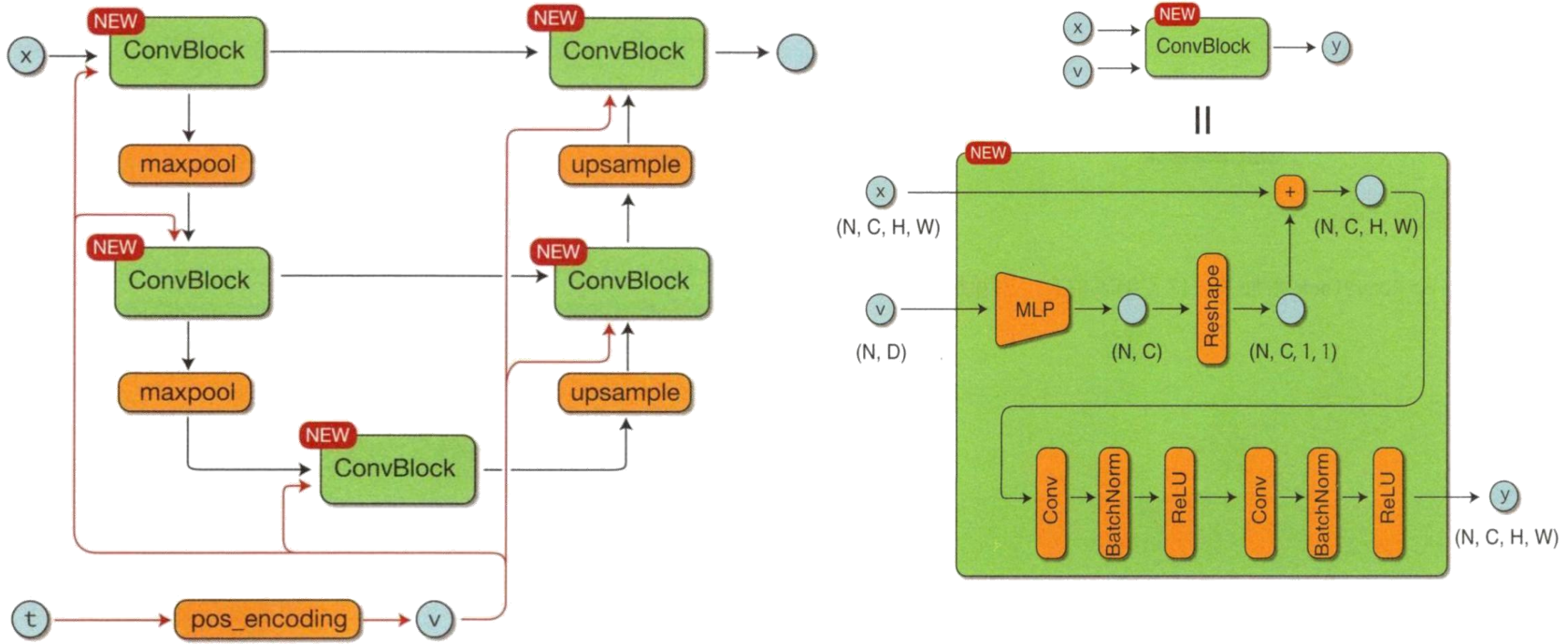
- 正弦位置编码

$$\mathbf{v}_i = \begin{cases} \sin\left(\dfrac{t}{10000^{\frac{i}{D}}}\right) & \text{(i为偶数时)} \\ \cos\left(\dfrac{t}{10000^{\frac{i}{D}}}\right) & \text{(i为奇数时)} \end{cases}$$



# 正弦位置编码嵌入U-Net

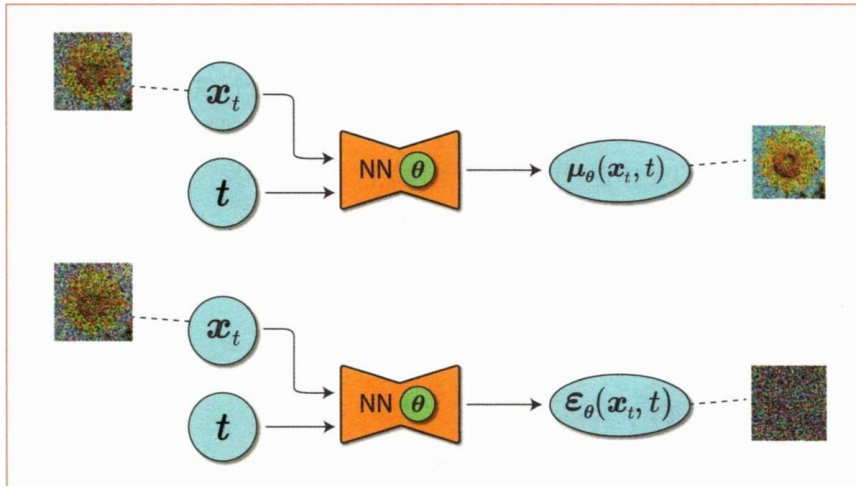
- 嵌入后的U-Net以及新ConvBlock



# 条件扩散模型

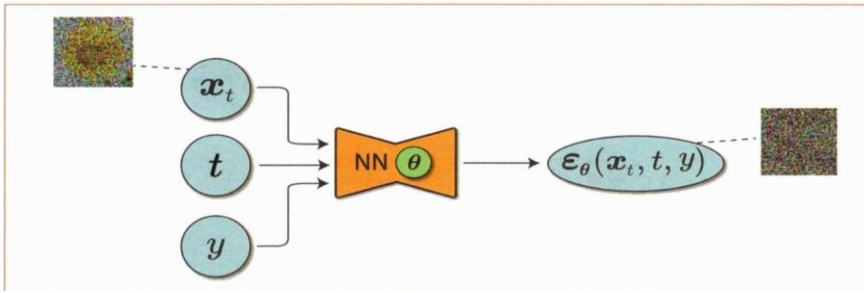
# 复习

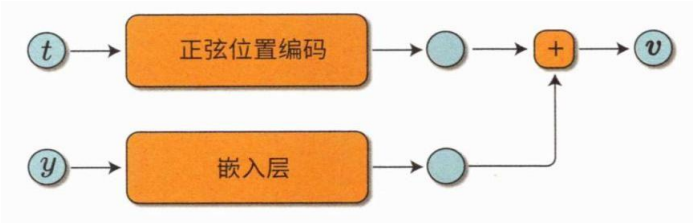
- 可以考虑使用神经网络对 $\boldsymbol{\mu}_\theta(\mathbf{x}_t, t)$ 和 $\varepsilon_\theta(\mathbf{x}_t, t)$ 进行建模



# 条件扩散模型

- 通过向神经网络中添加标签 $y$ ,可以使模型 “进化” 为条件扩散模型

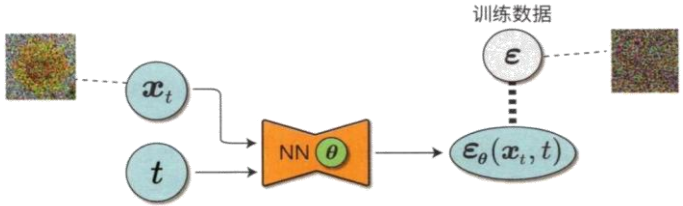


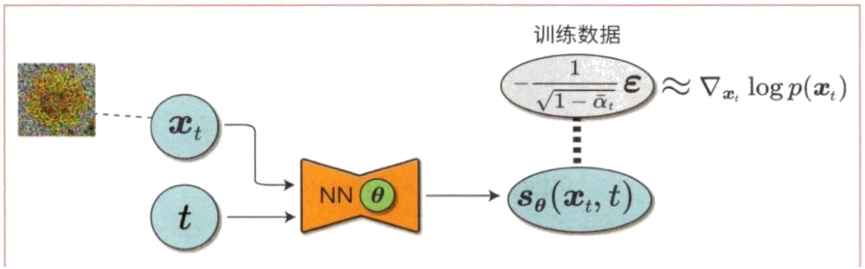
# 得分函数

# 使用的推断噪声 $\varepsilon$ 的神经网络

- 指引是一种将给定条件纳入模型并给予更多重视的机制
- 神经网络$\varepsilon_\theta(\mathbf{x}_t, t)$ 基于 $\mathbf{x}_t$ 和 $t$ 来推断噪声 $\varepsilon$：$\varepsilon \approx -\sqrt{1-\bar{\alpha}_t}\nabla_{x_t}\log p(\mathbf{x}_t)$




- “得分函数” 或 “得分”：$s_\theta(x_t, t) = \nabla_{x_t}\log p(\mathbf{x}_t) = -\frac{1}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)$
  - 输入数据 $\mathbf{x}_t$ 的对数似然$\log p(\mathbf{x}_t)$对$\mathbf{x}_t$的梯度



# 分类器指引

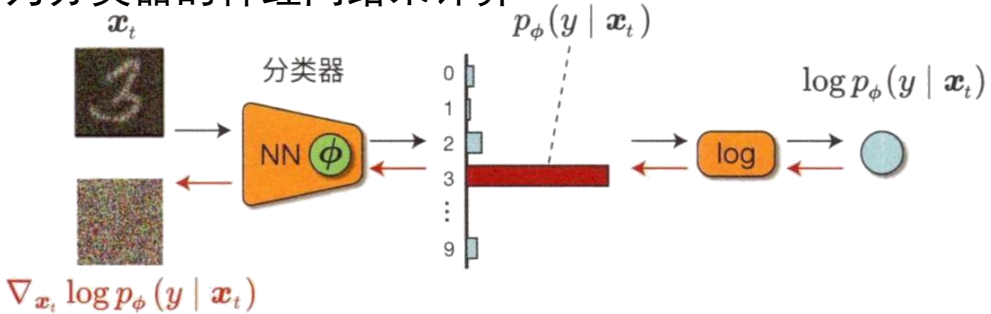
# 分类器

- 从得分的角度的指引方法，两种主要类型: 分类器指引 (classifier guidance) 和无分类器指引 (classifier-free guidance)

- $\nabla_{\mathbf{x}_t}\log p(\mathbf{x}_t \mid y) = \nabla_{\mathbf{x}_t}\log p(\mathbf{x}_t) + \nabla_{\mathbf{x}_t}\log p(y \mid \mathbf{x}_t)$

  条件得分　　1得分　　2分类器的对数似然的梯度

  - 1此项可以使用预测得分的神经网络 $\mathbf{s}_\theta(\mathbf{x}_t, t)$ 来计算
  - 2指导项可以使用作为分类器的神经网络来计算




- $\nabla_{\mathbf{x}_t}\log p(y \mid \mathbf{x}_t)$ 可以通过反向传播求出

# 阶段一 - 分类器$p_\phi(c \mid x_t, t)$

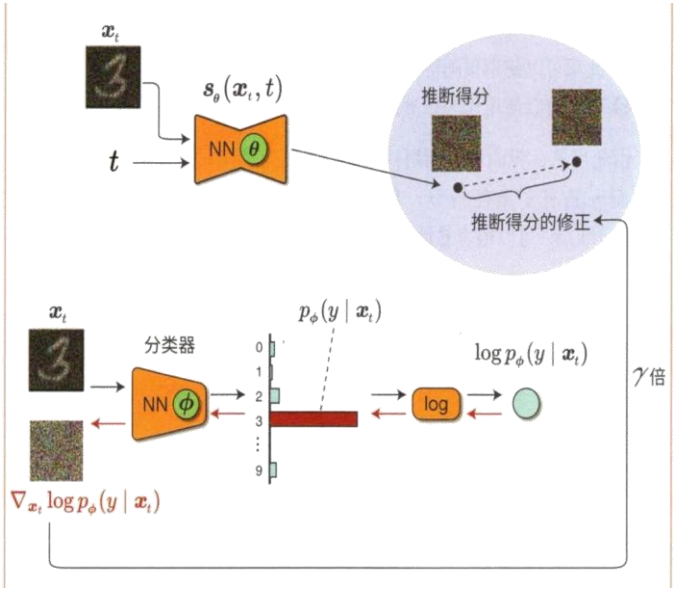
- 1.分类器
  - 能够处理任意噪声水平的分类器 $p_\psi$ 的输入是带噪图像 $x_t$ 和时间步 $t$（或等价的噪声水平 $\sigma_t$），输出类别概率
- 2．训练数据生成
  - 从带标签的数据集中取一个样本（ $x_0, c$ ）。随机采样一个时间步 $t$（从 1 到 $T$ ）。根据扩散模型的前向过程，生成对应的带噪图像 $x_t = \alpha_t x_0 + \sigma_t \epsilon$ ，其中 $\epsilon \sim \mathcal{N}(0, I)$ 。这样得到了一个训练样本：输入是 $(x_t, t)$ ，标签是 $c$
- 3．训练目标：使用标准的交叉熵损失

# 阶段二 - $\nabla_{\mathbf{x}_t} \log p_\phi(y \mid \mathbf{x}_t)$计算

- 在扩散模型的每一步反向采样过程中（例如，从 $t_i$ 到 $t_{i-1}$ ），执行以下操作：
  - 1．获取当前状态 - 带噪图像 $x_{t_i}$
  - 2．计算指导梯度 - 计算对于目标类别 $c$ 的指导项$\nabla_{\mathbf{x}_t} \log p_\phi\left(c \mid x_{t_i}, t_i\right)$
    - 启用梯度：将 $x_{t_i}$ 视为一个需要计算梯度的变量
    - 前向传播：将 $\left(x_{t_i}, t_i\right)$ 输入到我们训练好的分类器 $p_\psi$ 中，得到所有类别的对数概率（logits）
    - 选择目标：从输出的对数概率中，取出目标类别 $c$ 对应的那个值，即$\log p_\phi\left(c \mid x_{t_i}, t_i\right)$
    - 反向传播：对这个标量值调用 ．backward()
    - 提取梯度：此时，框架会自动计算出$\log p_\phi\left(c \mid x_{t_i}, t_i\right)$相对于输入 $x_{t_i}$ 的梯度，并存储在$x_{t_i}.grad$ 中

# 分类器指引

- 向分类器指引中引入权重 $\gamma$，$\nabla_{\mathbf{x}_t}\log p(\mathbf{x}_t \mid y) = \nabla_{\mathbf{x}_t}\log p(\mathbf{x}_t) + \gamma\nabla_{\mathbf{x}_t}\log p(y \mid \mathbf{x}_t)$
- 使用表示推断得分的神经网络 $\mathbf{s}_\theta(\mathbf{x}_t, t)$ 和表示分类器的 $p_\phi(y \mid \mathbf{x}_t)$
- $\nabla_{\mathbf{x}_t}\log p(\mathbf{x}_t \mid y) \approx \mathbf{s}_\theta(\mathbf{x}_t, t) + \gamma\nabla_{\mathbf{x}_t}\log p_\phi(y \mid \mathbf{x}_t)$



# 无分类器指引

# 无分类器指引（Classifier-Free Guidance, CFG）

- 从分类器指引的核心公式$\nabla_{x_t}\log p(x_t \mid c) = \nabla_{x_t}\log p(x_t) + \nabla_{x_t}\log p(c \mid x_t)$出发
- 贝叶斯定理，$p(c \mid x_t) = \frac{p(x_t|c)p(c)}{p(x_t)}$，取对数并对 $x_t$ 求梯度
  - $\nabla_{x_t}\log p(c \mid x_t) = \nabla_{x_t}\log p(x_t \mid c) - \nabla_{x_t}\log p(x_t)$
- 代入带指导强度$\gamma$的分类器指引公式中，得到CFG的插值形式
  - $\tilde{s}(x_t, t, c) = \nabla_{x_t}\log p(x_t) + \gamma \cdot \nabla_{x_t}\log p(c \mid x_t)$

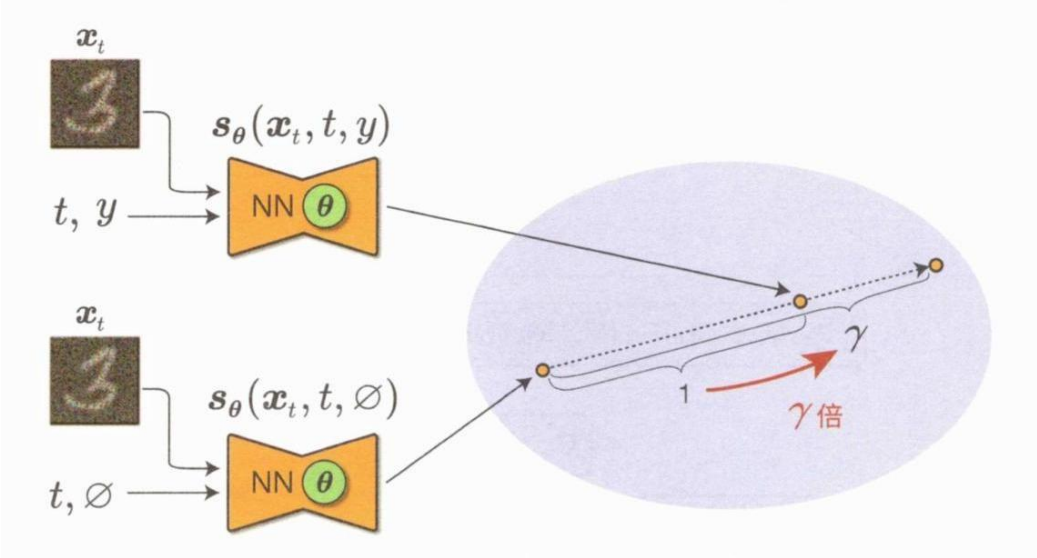
  $= \nabla_{x_t}\log p(x_t) + \gamma \cdot \left(\nabla_{x_t}\log p(x_t \mid c) - \nabla_{x_t}\log p(x_t)\right)$

  $= (1-\gamma)\nabla_{x_t}\log p(x_t) + \gamma\nabla_{x_t}\log p(x_t \mid c)$
- 更常用一个等价的外插形式：
  - $\tilde{s}(x_t, t, c) = \nabla_{x_t}\log p(x_t \mid c) + \gamma \cdot \left(\nabla_{x_t}\log p(x_t \mid c) - \nabla_{x_t}\log p(x_t)\right)$

    $= (1+\gamma)\nabla_{x_t}\log p(x_t \mid c) - \gamma\nabla_{x_t}\log p(x_t)$
- 其含义：从标准的条件分数出发，额外加上$\gamma$倍的＂方向偏差＂，从而强化条件信号

# 无分类器指引

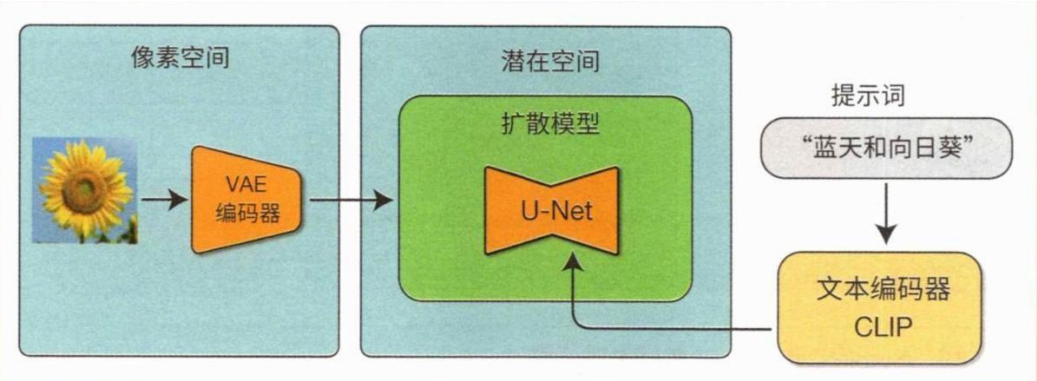
- 使用推断得分的神经网络实现的无分类器指引
  - $\nabla_{x_t}\log p(\mathbf{x}_t \mid y) \approx \mathbf{s}_\theta(\mathbf{x}_t, t, \varnothing) + \gamma\big(\mathbf{s}_\theta(\mathbf{x}_t, t, y) - \mathbf{s}_\theta(\mathbf{x}_t, t, \varnothing)\big)$



# Stable Diffusion

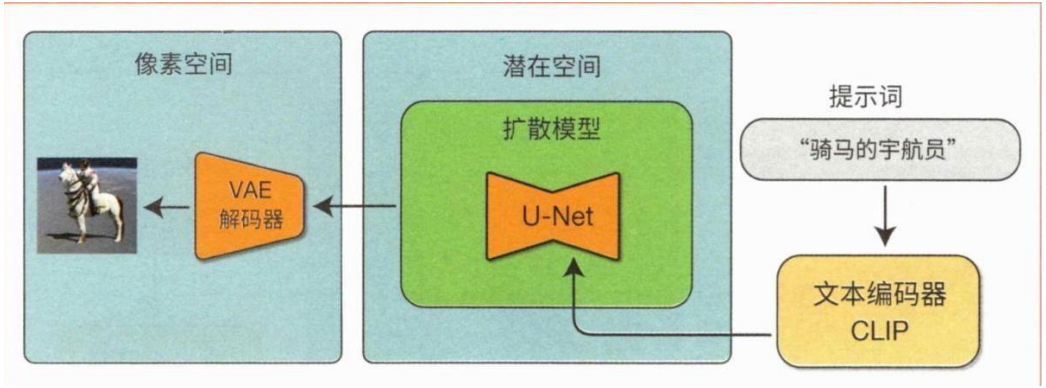
# Stable Diffusion 的工作原理

- Stable Diffusion 的训练流程



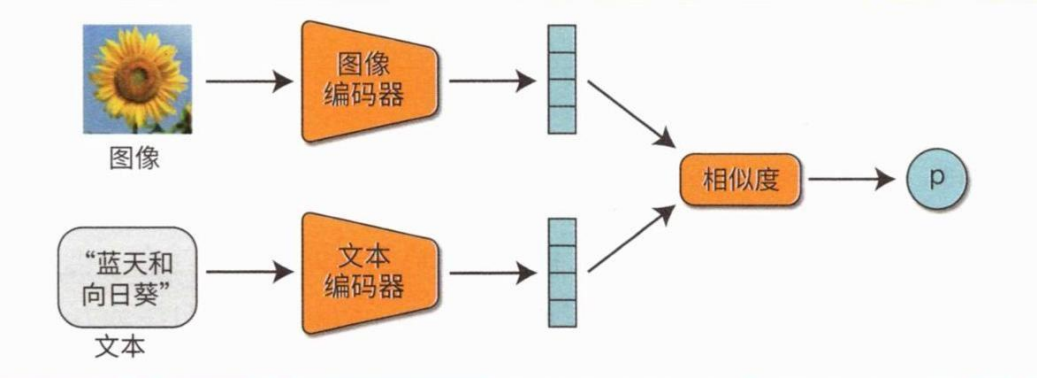
# 生成新数据

- 在生成新数据时, Stable Diffusion 会从潜在空间中的高斯噪声开始,执行逆扩散过程。最后在逆扩散过程结束时,它会再使用 VAE 解码器变换到像素空间



# CLIP 的训练流程

- CLIP 的训练流程



# 注意力机制

- 引入了注意力层的 U-Net