

# Denoising Diffusion Probabilistic Models

Reporter: 陈思玉

2022.11.19

Ho J, Jain A, Abbeel P. Denoising diffusion probabilistic models[J]. Advances in Neural Information Processing Systems, 2020, 33: 6840-6851.

# Author



Jonathan Ho

Google Brain

在 google.com 的电子邮件经过验证 - 首页

Artificial Intelligence Machine Learning

关注

创建我的个人资料

标题

引用次数 年份

Generative adversarial imitation learning EI检索

2160 2016

J Ho, S Ermon  
Advances in Neural Information Processing Systems, 4565-4573

Evolution strategies as a scalable alternative to reinforcement learning

1259 2017

T Salimans, J Ho, X Chen, S Sidor, I Sutskever  
arXiv preprint arXiv:1703.03864

Denoising diffusion probabilistic models EI检索

680 2020

J Ho, A Jain, P Abbeel  
Advances in Neural Information Processing Systems 33, 6840-6851

Motion planning with sequential convex optimization and convex collision checking EI检索

622 2014

SCI升级版 计算机科学3区 SCI基础版 工程技术2区 JCI 1.80 简介  
SCI Q1 SCIIF(5) 6.376 SCIIF 6.89 SCU 计算机科学C CUG 工程技术T2 XJU 二区

J Schulman, Y Duan, J Ho, A Lee, I Awwal, H Bradlow, J Pan, S Patil, ...  
The International Journal of Robotics Research 33 (9), 1251-1270

One-shot imitation learning EI检索

596 2017

Y Duan, M Andrychowicz, B Stadie, J Ho, J Schneider, I Sutskever, ...  
Advances in Neural Information Processing Systems, 1087-1098

Finding locally optimal, collision-free trajectories with sequential convex optimization.

496 2013

J Schulman, J Ho, AX Lee, I Awwal, H Bradlow, P Abbeel  
Robotics: science and systems 9 (1), 1-10

引用次数



开放获取的出版物数量

[查看全部](#)

0 篇文章

7 篇文章

无法查看的文章

[可查看的文章](#)

根据资金授权书

# Diffusion Model 生成效果

## DALLE2 生成



a dolphin in an astronaut suit on saturn, artstation



a propaganda poster depicting a cat dressed as french emperor napoleon holding a piece of cheese



a teddybear on a skateboard in times square

Figure 1: Selected  $1024 \times 1024$  samples from a production version of our model.

## Imagen 生成



Teddy bears swimming at the Olympics 400m Butter-fly event.

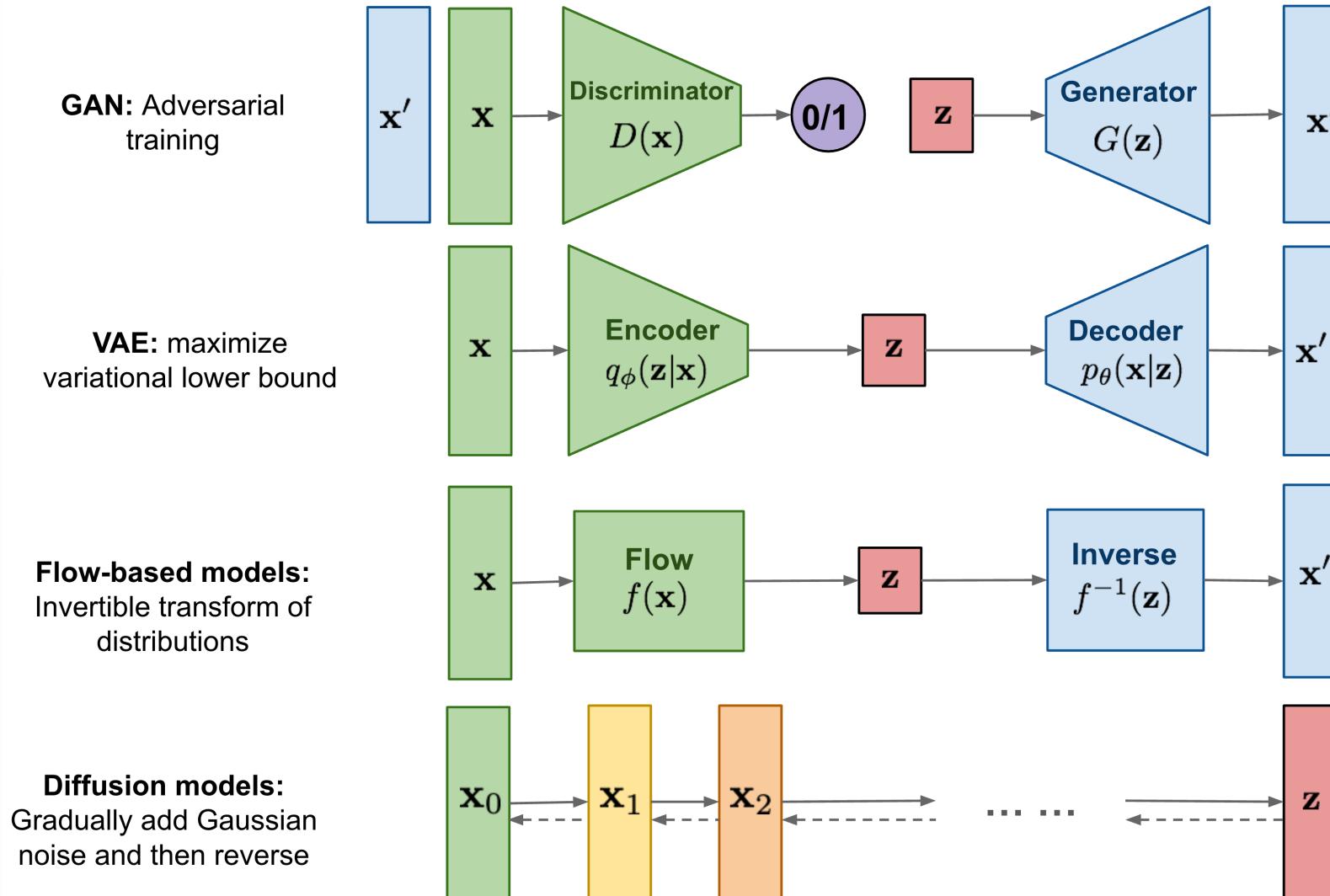


A cute corgi lives in a house made out of sushi.



A cute sloth holding a small treasure chest. A bright golden glow is coming from the chest.

# 生成模型对比



# Diffusion 前向过程

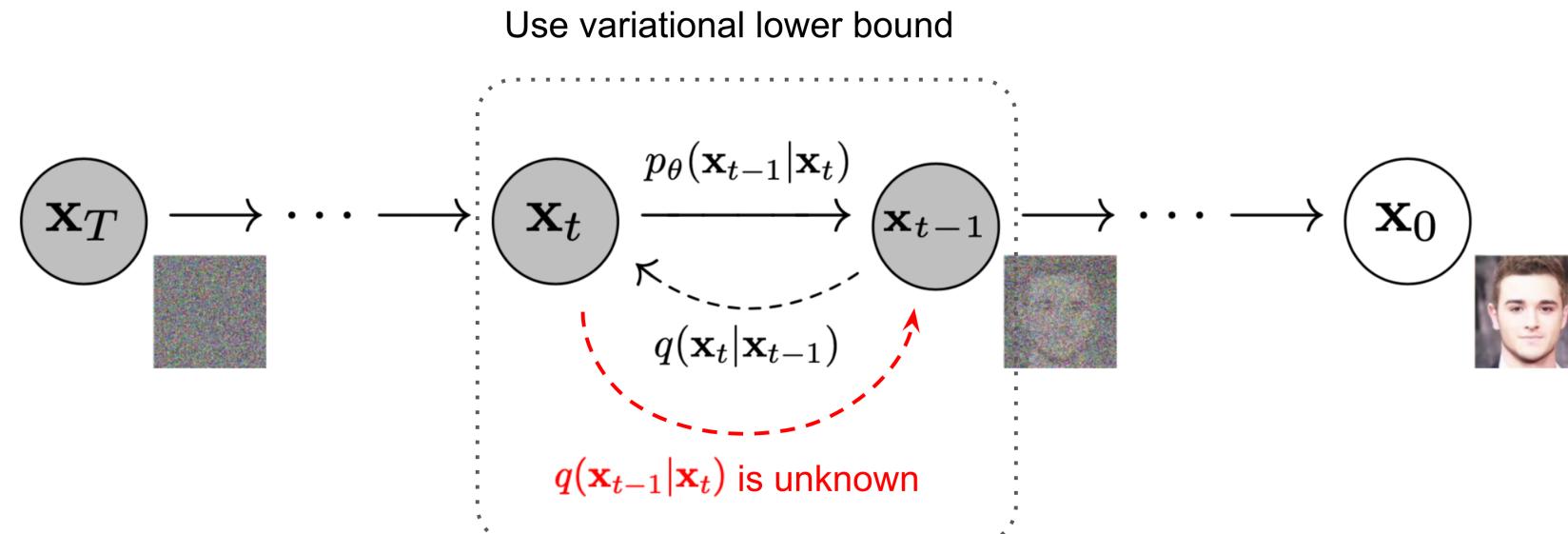
所谓前向过程，即往图片上加噪声的过程。

给定一个从真实数据分布  $\mathbf{x}_0 \sim q(\mathbf{x})$  中采样的数据点，定义一个 Diffusion 前向过程。在该过程中，每一步向样本添加少量高斯噪声，产生一系列噪声样本  $\mathbf{x}_1, \dots, \mathbf{x}_T$ 。权重由  $\{\beta_t \in (0, 1)\}_{t=1}^T$  控制。

$$q(\mathbf{x}_t | \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1 - \beta_t} \mathbf{x}_{t-1}, \beta_t \mathbf{I}) \quad q(\mathbf{x}_{1:T} | \mathbf{x}_0) = \prod_{t=1}^T q(\mathbf{x}_t | \mathbf{x}_{t-1})$$

随着  $t$  变大，数据样本  $\mathbf{x}_t$  逐渐失去其可区分的特征，越来越接近高斯噪声。

最终当  $T \rightarrow \infty$  时， $\mathbf{x}_T$  等价于各向同性高斯分布。



# Diffusion 前向过程

上述过程的一个很好的特性是，我们可以使用 [reparameterization trick](#) 以封闭形式在任意时间步  $t$  对  $\mathbf{x}_t$  进行采样。

$$(\text{不可导}) \text{ 采样 } z \sim \mathcal{N}(z; \mu_\theta, \sigma_\theta^2 \mathbf{I}) \rightarrow (\text{可导}) z = \mu_\theta + \sigma_\theta \odot \epsilon, \epsilon \sim \mathcal{N}(0, \mathbf{I})$$

令  $\alpha_t = 1 - \beta_t$ ,  $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$ :

$$\begin{aligned} \mathbf{x}_t &= \sqrt{\alpha_t} \mathbf{x}_{t-1} + \sqrt{1 - \alpha_t} \epsilon_{t-1} && \text{where } \epsilon_{t-1}, \epsilon_{t-2}, \dots \sim \mathcal{N}(0, \mathbf{I}) \\ &= \sqrt{\alpha_t} (\sqrt{\alpha_{t-1}} \mathbf{x}_{t-2} + \sqrt{1 - \alpha_{t-1}} \epsilon_{t-2}) + \sqrt{1 - \alpha_t} \epsilon_{t-1} \\ &= \sqrt{\alpha_t \alpha_{t-1}} \mathbf{x}_{t-2} + \sqrt{\alpha_t (1 - \alpha_{t-1})} \epsilon_{t-2} + \sqrt{1 - \alpha_t} \epsilon_{t-1} \\ &= \sqrt{\alpha_t \alpha_{t-1}} \mathbf{x}_{t-2} + \sqrt{1 - \alpha_t \alpha_{t-1}} \bar{\epsilon}_{t-2} && \text{where } \bar{\epsilon}_{t-2} \text{ merge two Gaussians (*)} \\ &= \dots \\ &= \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon_t \end{aligned}$$

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

由于独立高斯分布可加性，合并  $\mathcal{N}(0, \sigma_1^2 \mathbf{I}), \mathcal{N}(0, \sigma_2^2 \mathbf{I})$  时，新分布是  $\mathcal{N}(0, (\sigma_1^2 + \sigma_2^2) \mathbf{I})$ 。这里合并的标准差是：

$$\sqrt{\alpha_t (1 - \alpha_{t-1}) + (1 - \alpha_t)} = \sqrt{1 - \alpha_t \alpha_{t-1}}$$

通常，当样本变得更嘈杂时，可以承受更大的更新步长。

因此， $\beta_t$  随着  $t$  增大而递增，即  $\beta_1 < \beta_2 < \dots < \beta_T$ 。

# Diffusion 反向过程

如果可以反转上述过程并从  $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$  中采样，将能够从高斯噪声输入  $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$  中重建真实样本。

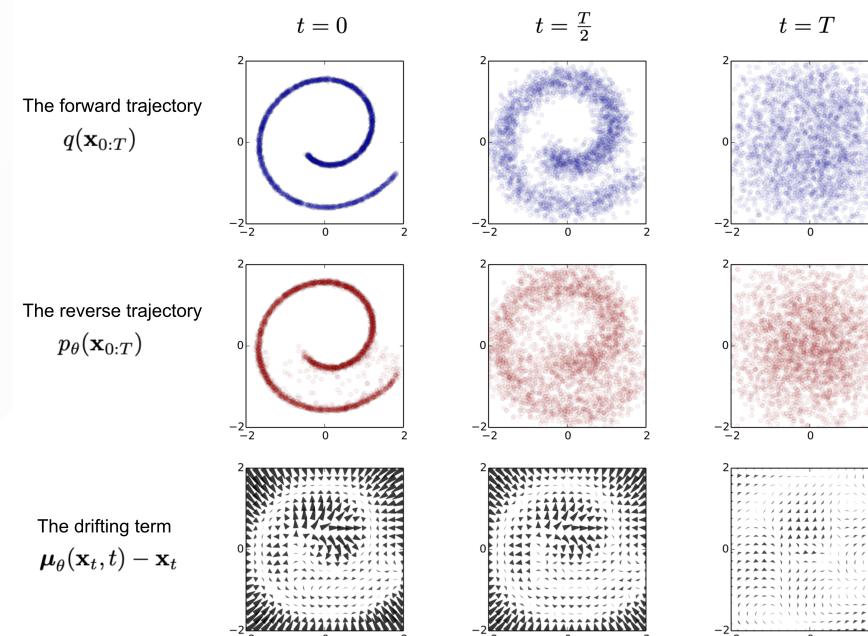
注意，如果  $\beta_t$  足够小， $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$  也将是高斯分布的。

由于这需要使用整个数据集，不能轻易估计  $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ 。因此，需要学习一个模型  $p_\theta$  来逼近。

目前主流模型是 U-Net + Attention 的结构。

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

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



# Diffusion 反向过程

当以  $\mathbf{x}_0$  为条件时，反向条件概率是易于处理的：

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

使用贝叶斯公式，有：

$$\begin{aligned} q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) &= q(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{x}_0) \frac{q(\mathbf{x}_{t-1} | \mathbf{x}_0)}{q(\mathbf{x}_t | \mathbf{x}_0)} \\ &\propto \exp \left( -\frac{1}{2} \left( \frac{(\mathbf{x}_t - \sqrt{\alpha_t} \mathbf{x}_{t-1})^2}{\beta_t} + \frac{(\mathbf{x}_{t-1} - \sqrt{\bar{\alpha}_{t-1}} \mathbf{x}_0)^2}{1 - \bar{\alpha}_{t-1}} - \frac{(\mathbf{x}_t - \sqrt{\bar{\alpha}_t} \mathbf{x}_0)^2}{1 - \bar{\alpha}_t} \right) \right) \\ &= \exp \left( -\frac{1}{2} \left( \left( \frac{\alpha_t}{\beta_t} + \frac{1}{1 - \bar{\alpha}_{t-1}} \right) \mathbf{x}_{t-1}^2 - \left( \frac{2\sqrt{\alpha_t}}{\beta_t} \mathbf{x}_t + \frac{2\sqrt{\bar{\alpha}_{t-1}}}{1 - \bar{\alpha}_{t-1}} \mathbf{x}_0 \right) \mathbf{x}_{t-1} + C(\mathbf{x}_t, \mathbf{x}_0) \right) \right) \end{aligned}$$

其中  $C(\mathbf{x}_t, \mathbf{x}_0)$  是一些不涉及  $\mathbf{x}_{t-1}$  的函数。

# Diffusion 反向过程

按照标准的高斯密度函数，方差和均值可以参数化如下 ( $\alpha_t = 1 - \beta_t$  和  $\bar{\alpha}_t = \prod_{i=1}^T \alpha_i$ ) :

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

由  $\mathbf{x}_t = \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon_t$  得到  $\mathbf{x}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}} (\mathbf{x}_t - \sqrt{1 - \bar{\alpha}_t} \epsilon_t)$ , 并代入上式得到:

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

# Diffusion 反向过程

因此，模型需要预测噪声  $\epsilon_t$ ，并拟合  $\tilde{\mu}_t$ ：

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

其中， $\epsilon(\mathbf{x}, t)$  为模型。

最后，DDPM 的反向过程可以总结为：

1. 每个步骤  $t$  通过  $\mathbf{x}_t, t$  预测噪声  $\epsilon_\theta(\mathbf{x}_t, t)$ ，并得到  $\mu_\theta(\mathbf{x}_t, t)$
2. 得到方差  $\Sigma_\theta(\mathbf{x}_t, t)$ 
  - DDPM 中令方差  $\Sigma_\theta(\mathbf{x}_t, t) = \tilde{\beta}_t$ ，且认为  $\tilde{\beta}_t = \beta_t$  和  $\tilde{\beta}_t = \frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t} \beta_t$  的结果近似。
  - GLIDE 中则是根据网络预测方差  $\Sigma_\theta(\mathbf{x}_t, t)$ 。
3. 计算  $p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t))$ ，并根据 reparameterization trick 得到  $\mathbf{x}_{t-1}$

# Diffusion 训练

为了 Diffusion Models 得到合适的  $\mu_\theta(\mathbf{x}_t, t)$ ,  $\Sigma_\theta(\mathbf{x}_t, t)$ , 最大化模型预测分布与真实数据分布的对数似然, 即优化  $\mathbf{x}_0 \sim q(\mathbf{x}_0)$  下的  $p_\theta(\mathbf{x}_0)$  交叉熵:

$$L = \mathbb{E}_{q(\mathbf{x}_0)}[-\log p_\theta(\mathbf{x}_0)]$$

这种设置与 VAE 非常相似, 因此可以使用 Variational Lower Bound 来优化负对数似然。

$$\begin{aligned} -\log p_\theta(\mathbf{x}_0) &\leq -\log p_\theta(\mathbf{x}_0) + D_{KL}(q(\mathbf{x}_{1:T}|\mathbf{x}_0)\|p_\theta(\mathbf{x}_{1:T}|\mathbf{x}_0)) \\ &= -\log p_\theta(\mathbf{x}_0) + \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[ \log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})/p_\theta(\mathbf{x}_0)} \right] \quad \text{where } p_\theta(\mathbf{x}_{1:T}|\mathbf{x}_0) = \frac{p_\theta(\mathbf{x}_{0:T})}{p_\theta(\mathbf{x}_0)} \\ &= -\log p_\theta(\mathbf{x}_0) + \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[ \log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} + \underbrace{\log p_\theta(\mathbf{x}_0)}_{\substack{\text{常数,} \\ \text{与 } q \text{ 无关}}} \right] \\ &= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[ \log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} \right] \\ \text{Let } L_{VLB} &= \underbrace{\mathbb{E}_{q(\mathbf{x}_0)} \left( \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[ \log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} \right] \right)}_{Fubini \text{ 定理}} = \mathbb{E}_{q(\mathbf{x}_0:T)} \left[ \log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} \right] \\ &\geq \mathbb{E}_{q(\mathbf{x}_0)}[-\log p_\theta(\mathbf{x}_0)] \end{aligned}$$

最小化  $L_{VLB}$  即可最小化目标损失。

# Diffusion 训练

为了将方程中的每个项转换为可分析计算的，可以将目标进一步重写为几个 KL 散度和熵项的组合：

$$\begin{aligned}
 L_{VLB} &= E_{q(\mathbf{x}_0:T)} \left[ \log \frac{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{0:T})} \right] \\
 &= E_q \left[ \log \frac{\prod_{t=1}^T q(\mathbf{x}_t|\mathbf{x}_{t-1})}{p_\theta(\mathbf{x}_T) \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} \right] \\
 &= E_q \left[ -\log p_\theta(\mathbf{x}_T) + \sum_{t=1}^T \log \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1})}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} \right] \\
 &= E_q \left[ -\log p_\theta(\mathbf{x}_T) + \sum_{t=2}^T \log \frac{q(\mathbf{x}_t|\mathbf{x}_{t-1})}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} + \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)} \right] \\
 &= E_q \left[ -\log p_\theta(\mathbf{x}_T) + \sum_{t=2}^T \log \left( \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} \cdot \frac{q(\mathbf{x}_t|\mathbf{x}_0)}{q(\mathbf{x}_{t-1}|\mathbf{x}_0)} \right) + \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)} \right] \\
 &= E_q \left[ -\log p_\theta(\mathbf{x}_T) + \sum_{t=2}^T \log \left( \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} \right) + \sum_{t=2}^T \log \left( \frac{q(\mathbf{x}_t|\mathbf{x}_0)}{q(\mathbf{x}_{t-1}|\mathbf{x}_0)} \right) + \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)} \right] \\
 &= E_q \left[ -\log p_\theta(\mathbf{x}_T) + \sum_{t=2}^T \log \left( \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} \right) + \log \frac{q(\mathbf{x}_t|\mathbf{x}_0)}{q(\mathbf{x}_1|\mathbf{x}_0)} + \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)} \right] \\
 &= E_q \left[ \log \frac{q(\mathbf{x}_T|\mathbf{x}_0)}{p_\theta(\mathbf{x}_T)} + \sum_{t=2}^T \log \left( \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)} \right) - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1) \right] \\
 &= \mathbb{E}_q \left[ \underbrace{D_{KL}(q(\mathbf{x}_T|\mathbf{x}_0)\|p_\theta(\mathbf{x}_T))}_{L_T} + \sum_{t=2}^T \underbrace{D_{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)\|p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}} - \underbrace{\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0} \right]
 \end{aligned}$$

# Diffusion 训练

分别标记  $L_{VLB}$  中的每个分量：

$$L_{VLB} = L_T + L_{T-1} + \cdots + L_0$$

where  $L_T = D_{KL}(q(\mathbf{x}_T | \mathbf{x}_0) \| p_\theta(\mathbf{x}_T))$

$$L_{t-1} = D_{KL}(q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) \| p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)) \quad \text{for } 2 \leq t \leq T$$

$$L_0 = -\log p_\theta(\mathbf{x}_0 | \mathbf{x}_1)$$

$L_{VLB}$  中的每个 KL 项 ( $L_0$  除外) 比较两个高斯分布，因此可以以封闭形式计算它们。

$L_T$  是常数，在训练期间可以忽略，因为  $q$  没有可学习的参数，而  $x_T$  是高斯噪声。

[Ho 等人 2020](#) 模型  $L_0$  使用从  $\mathcal{N}(\mathbf{x}_0; \mu_\theta(\mathbf{x}_1, 1), \Sigma_\theta(\mathbf{x}_1, 1))$  派生的单独离散解码器。

我们想训练  $\mu_\theta$  预测  $\tilde{\mu}_t = \frac{1}{\sqrt{\alpha_t}} \left( \mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_t \right)$ 。

因为  $\mathbf{x}_t$  在训练时作为输入可用，所以可对高斯噪声项进行 reparameterization，在时间步骤  $t$  根据输入  $\mathbf{x}_t$  预测  $\epsilon_t$ ：

$$\mu_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}} \left( \mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_0(\mathbf{x}_t, t) \right)$$

$$\text{Thus } \mathbf{x}_{t-1} = \mathcal{N} \left( \mathbf{x}_{t-1}; \frac{1}{\sqrt{\alpha_t}} \left( \mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_0(\mathbf{x}_t, t) \right), \Sigma_\theta(\mathbf{x}_t, t) \right)$$

# Diffusion 训练

$L_t$  可以看作拉近 2 个高斯分布：

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

根据多元高斯分布的KL散度求解：

$$L_t = \mathbb{E}_{\mathbf{x}_0, \epsilon} \left[ \frac{1}{2\|\Sigma_\theta(\mathbf{x}_t, t)\|_2^2} \|\tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0) - \mu_\theta(\mathbf{x}_t, t)\|^2 \right] + C$$

其中  $C$  是与模型参数  $\theta$  无关的常数。

$$\begin{aligned} L_t &= D_{KL}(q(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}_0) \| p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t)) \\ &= \mathbb{E}_{\mathbf{x}_0, \epsilon} \left[ \frac{1}{2\|\Sigma_\theta(\mathbf{x}_t, t)\|_2^2} \|\tilde{\mu}_t(\mathbf{x}_t, \mathbf{x}_0) - \mu_\theta(\mathbf{x}_t, t)\|^2 \right] \\ &= \mathbb{E}_{\mathbf{x}_0, \epsilon} \left[ \frac{1}{2\|\Sigma_\theta(\mathbf{x}_t, t)\|_2^2} \left\| \frac{1}{\sqrt{\alpha_t}} \left( \mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_t \right) - \frac{1}{\sqrt{\alpha_t}} \left( \mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(\mathbf{x}_t, t) \right) \right\|^2 \right] \\ &= \mathbb{E}_{\mathbf{x}_0, \epsilon} \left[ \frac{(1 - \alpha_t)^2}{2\alpha_t(1 - \bar{\alpha}_t)\|\Sigma_\theta(\mathbf{x}_t, t)\|_2^2} \|\epsilon_t - \epsilon_\theta(\mathbf{x}_t, t)\|^2 \right] \\ &= \mathbb{E}_{\mathbf{x}_0, \epsilon} \left[ \frac{(1 - \alpha_t)^2}{2\alpha_t(1 - \bar{\alpha}_t)\|\Sigma_\theta(\mathbf{x}_t, t)\|_2^2} \|\epsilon_t - \epsilon_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon_t, t)\|^2 \right] \end{aligned}$$

可以看出，Diffusion 训练的核心就是学习高斯噪声  $\epsilon_t, \epsilon_\theta$  间的 MSE

# Diffusion 训练

从经验上, [Ho 等人 2020](#) 发现训练 Diffusion 模型可以通过一个简化的目标来更好地奏效, 该目标忽略了权重术语:

$$\begin{aligned} L_t^{\text{simple}} &= \mathbb{E}_{t \sim [1, T], \mathbf{x}_0, \epsilon_t} [\|\epsilon_t - \epsilon_\theta(\mathbf{x}_t, t)\|^2] \\ &= \mathbb{E}_{t \sim [1, T], \mathbf{x}_0, \epsilon_t} [\|\epsilon_t - \epsilon_\theta(\sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon_t, t)\|^2] \end{aligned}$$

最终的优化目标是:

$$L_{\text{simple}} = L_t^{\text{simple}} + C$$

其中  $C$  是一个与  $\theta$  无关的常数。

## Algorithm 1 Training

```

1: repeat
2:    $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ 
3:    $t \sim \text{Uniform}(\{1, \dots, T\})$ 
4:    $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
5:   Take gradient descent step on
       $\nabla_{\theta} \|\epsilon - \epsilon_{\theta}(\sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, t)\|^2$ 
6: until converged

```

## Algorithm 2 Sampling

```

1:  $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
2: for  $t = T, \dots, 1$  do
3:    $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  if  $t > 1$ , else  $\mathbf{z} = \mathbf{0}$ 
4:    $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left( \mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_{\theta}(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$ 
5: end for
6: return  $\mathbf{x}_0$ 

```

```

def forward(self, x_0, labels):
    device = x_0.device

    t = torch.randint(self.T, size=(x_0.shape[0],), device=device)
    noise = torch.randn_like(x_0).to(device)
    x_t = (
        self.extract(self.sqrt_alphas_bar, t, x_0.shape) * x_0
        + self.extract(self.sqrt_one_minus_alphas_bar, t, x_0.shape) * noise
    )

    loss = nn.MSELoss(reduction="none")(self.model(x_t, t, labels), noise)
    return loss

```