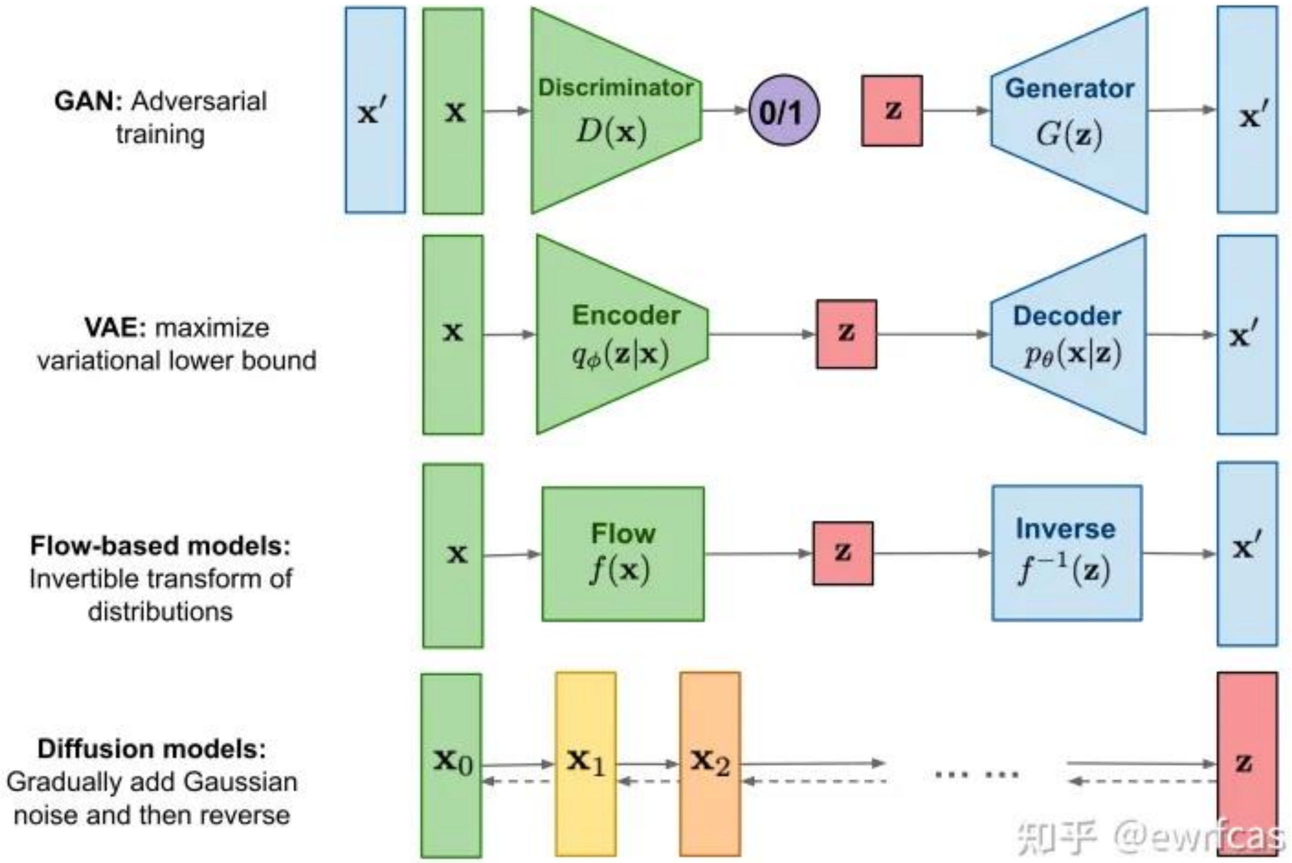
初探Diffusion model

Diffusion model

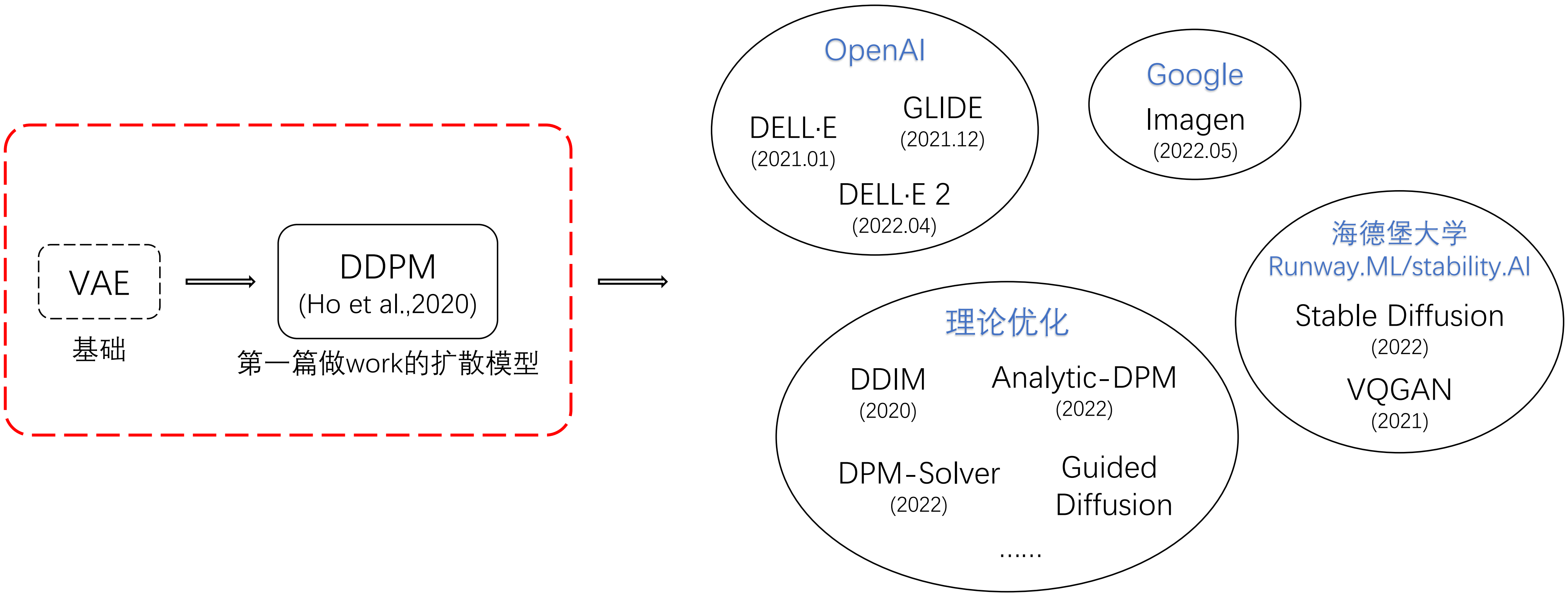
图1. DALLE2生成结果

图2. Imagen生成结果

生成方法 - Diffusion Model



Diffusion model的发展路径



Auto-Encoding Variational Bayes

Diederik P. Kingma
Machine Learning Group
Universiteit van Amsterdam
dpkingma@gmail.com

Max Welling
Machine Learning Group
Universiteit van Amsterdam
welling.max@gmail.com

VAE

Variational Auto-Encoder 2013.12

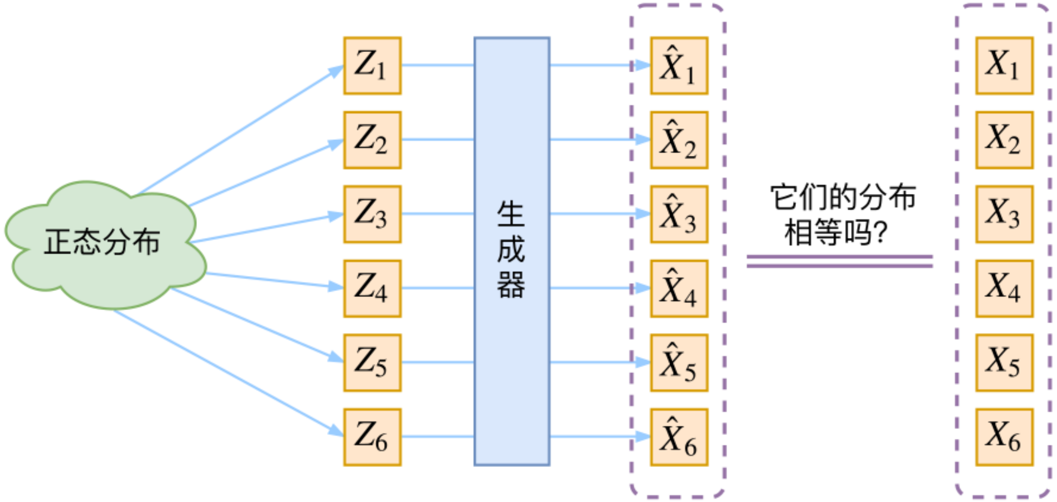
苏剑林. (Mar. 18, 2018). 《变分自编码器（一）：原来是这么一回事》[Blog post]. Retrieved from https://kexue.fm/archives/5253

VAE – 分布变换

- 生成样本
  - 从一个采样的隐变量Z生成目标样本X:

$$\hat{X} = g(z)$$

    - 一般假设Z服从某些分布（正态分布、均匀分布）



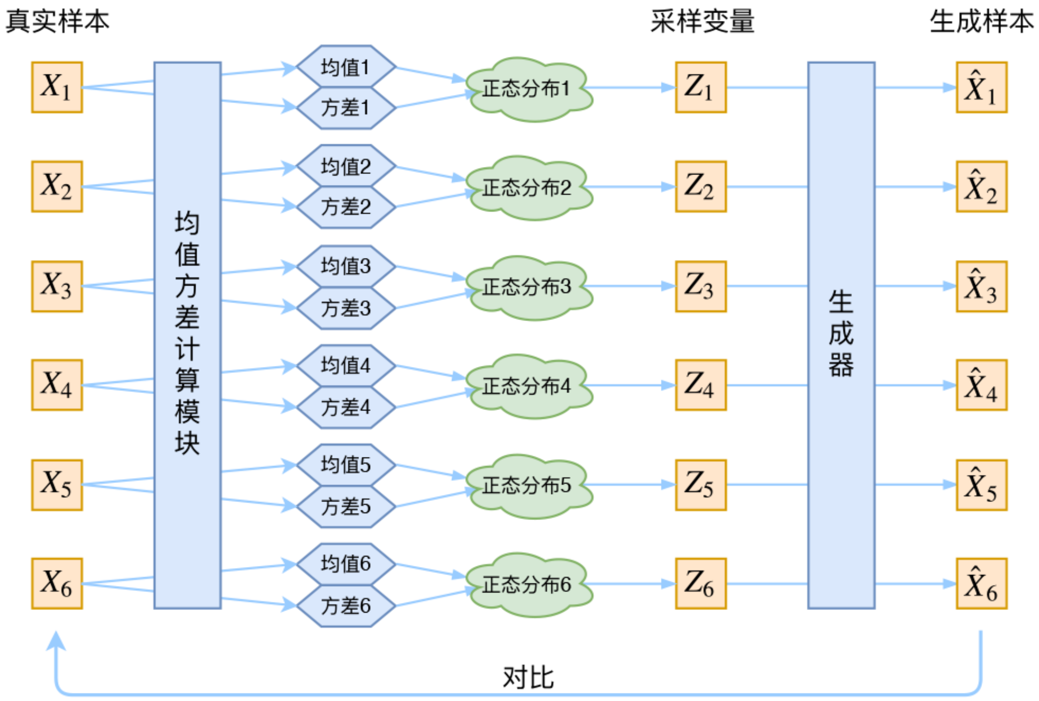
VAE – 分布变换

- 如何让生成样本的分布符合真实样本的分布？
  - 对于一个真实样本X_k，假设$p(Z|X_k)$符合正态分布
  - 最小化$\hat{X}_k = g(Z)$与X_k的重构误差
  - 相当于对于每一个样本都有一个自己的专属正态分布$p(Z|X_k)$

VAE – 分布变换

- 如何得到正态分布的参数（均值、方差）？
  - ➢ 直接用两个神经网络去拟合!

$$\mu_k = f_1(X_k), \log \sigma_k^2 = f_2(X_k)$$



VAE – 分布标准化

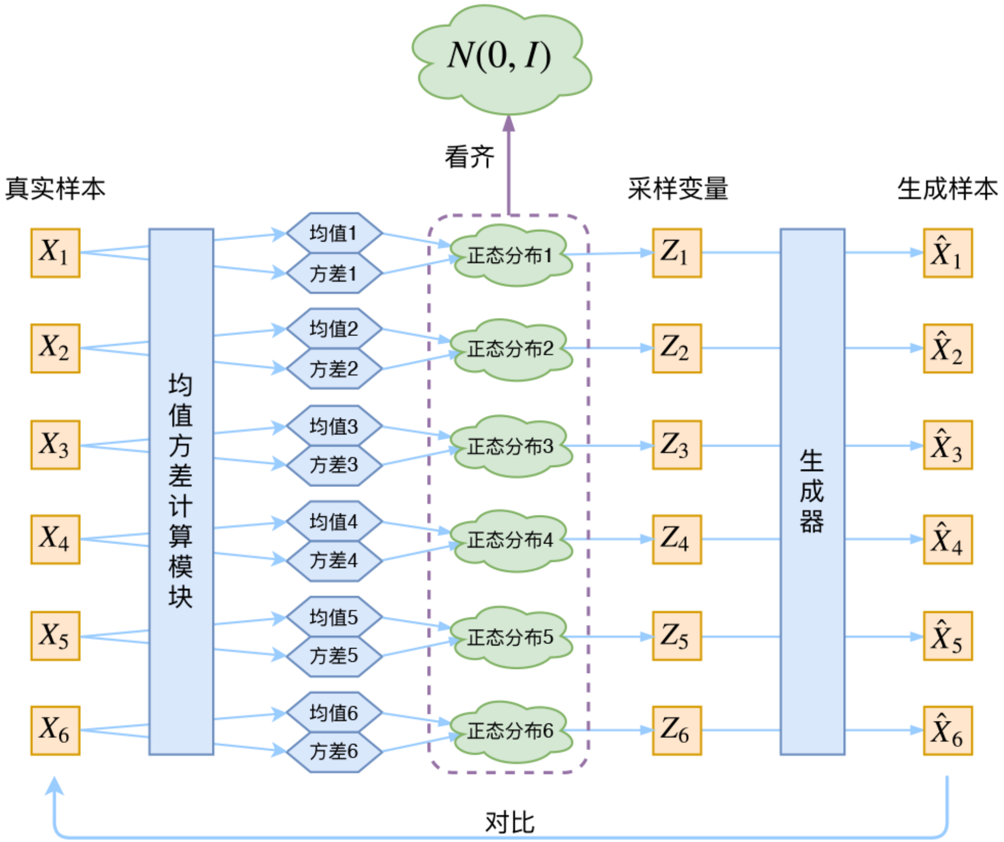
- 流程：$X_k \rightarrow \mu_k, \sigma_k \rightarrow Z_k \rightarrow \hat{X}_k \rightarrow D\left(X_k, \hat{X}_k\right)^2$
  - 采样Z_k的过程包含噪声σ_k，噪声太大会增加重构难度，噪声太小会让生成模型丧失随机性

➢ VAE让所有的$p(Z|X_k)$都向标准正态分布看齐，利用KL散度来监督f_1, f_2：

$$\mathcal{L}_{\mu,\sigma^2} = \frac{1}{2}\sum_{i=1}^{d}\left(\mu_{(i)}^2 + \sigma_{(i)}^2 - \log\sigma_{(i)}^2 - 1\right)$$

VAE – 分布标准化

$$p(Z)=\sum_X p(Z|X)p(X)=\sum_X \mathcal{N}(0,I)p(X)=\mathcal{N}(0,I)\sum_X p(X)=\mathcal{N}(0,I)$$

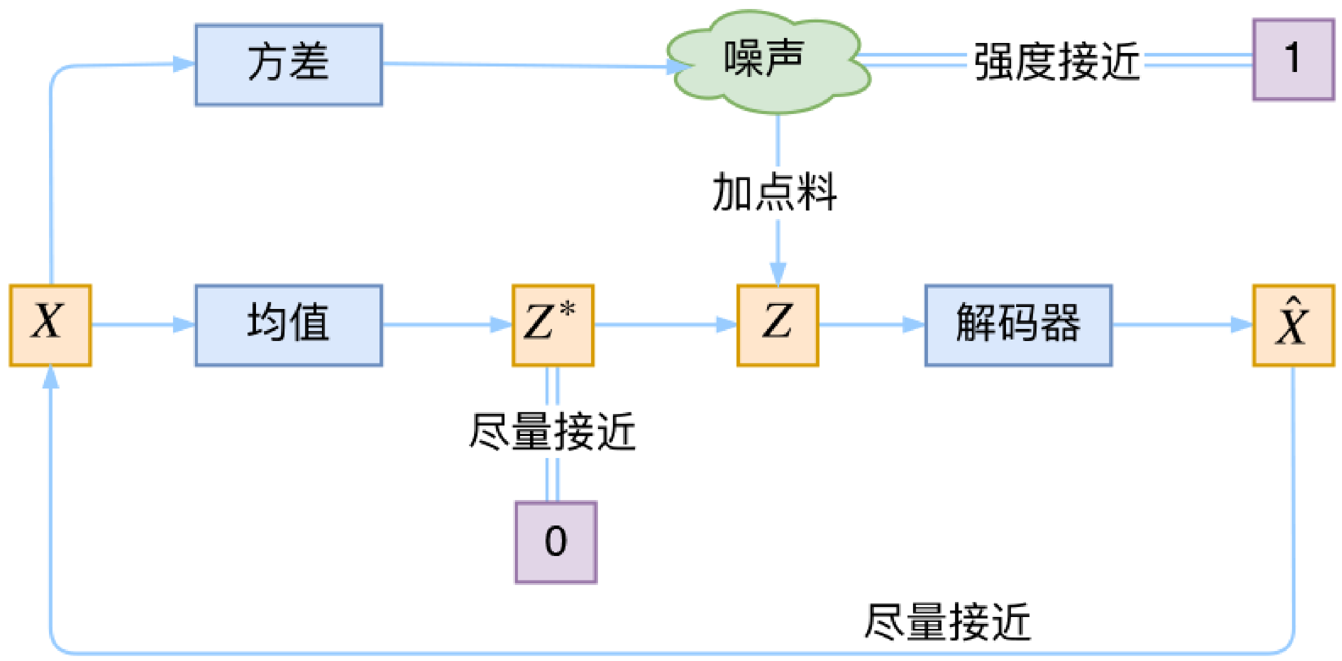


VAE – 训练技巧

- 流程：$X_k \rightarrow \mu_k, \sigma_k \rightarrow Z_k \rightarrow \hat{X}_k \rightarrow D\left(X_k, \hat{X}_k\right)^2$
  - 问题：采样过程不可导
  - 重参数技巧（reparameterization trick）
    - 从$\mathcal{N}(\mu, \sigma^2)$中采样一个$Z$，相当于从$\mathcal{N}(0, \boldsymbol{I})$中采样一个$\varepsilon$，然后让$Z = \mu + \varepsilon * \sigma$。
    - 可使采样过程下放到标准正态分布，让全流程可导。

Variational Auto-Encoder 总结

- 与以往的AE不同的是，VAE的Encoder是用与计算均值和方差
- 重构的过程是希望没噪声的，而KL loss则希望有高斯噪声的



Denoising Diffusion Probabilistic Models

Jonathan Ho
UC Berkeley
jonathanho@berkeley.edu

Ajay Jain
UC Berkeley
ajayj@berkeley.edu

Pieter Abbeel
UC Berkeley
pabbeel@cs.berkeley.edu

DDPM

Denoising Diffusion Probabilistic Models NeruIPS 2020.06

Deep Unsupervised Learning using Nonequilibrium Thermodynamics ICML 2015.03

苏剑林. (Jun. 13, 2022). 《生成扩散模型漫谈（一）：DDPM = 拆楼 + 建楼》[Blog post]. Retrieved from https://kexue.fm/archives/9119

DDPM 生成效果

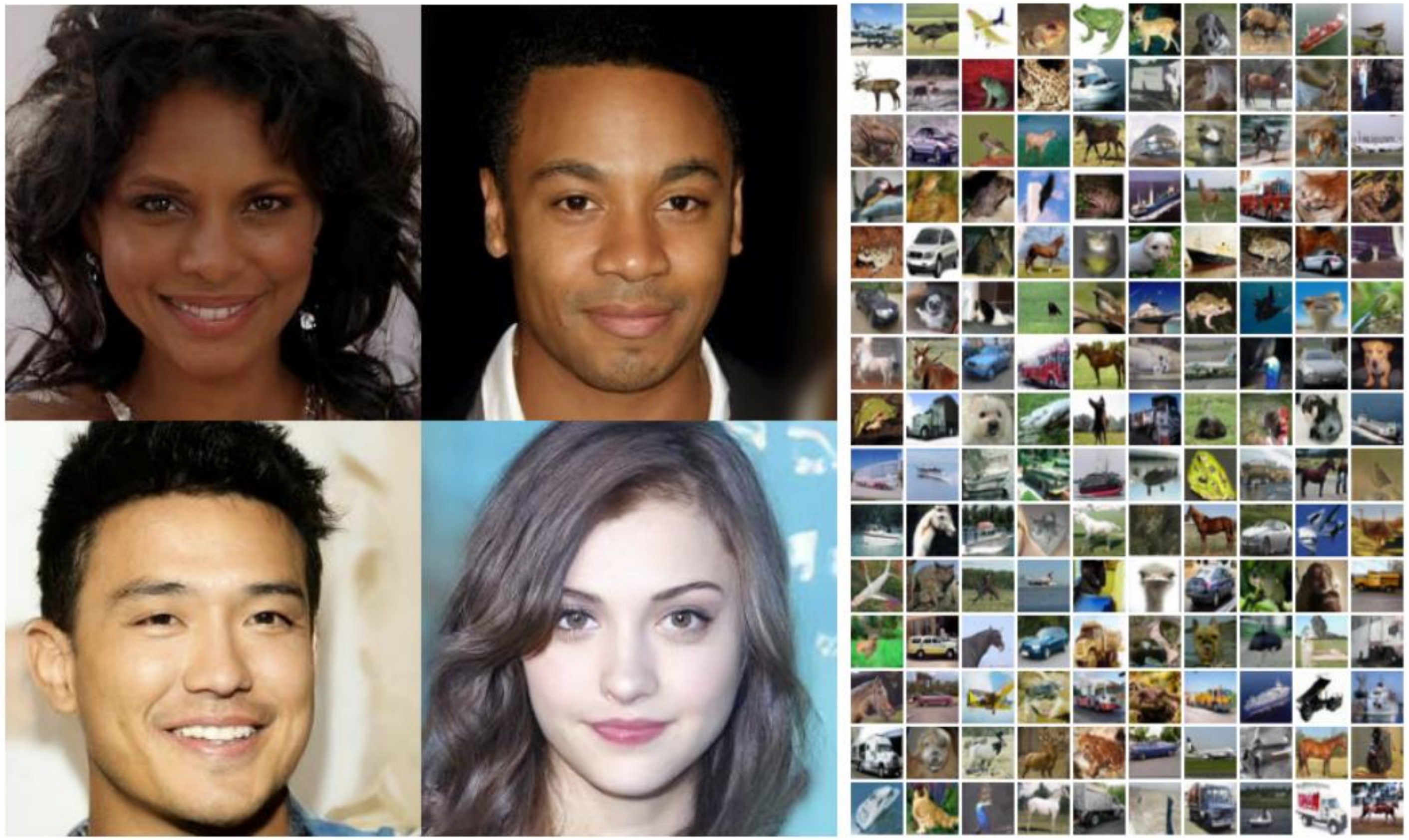
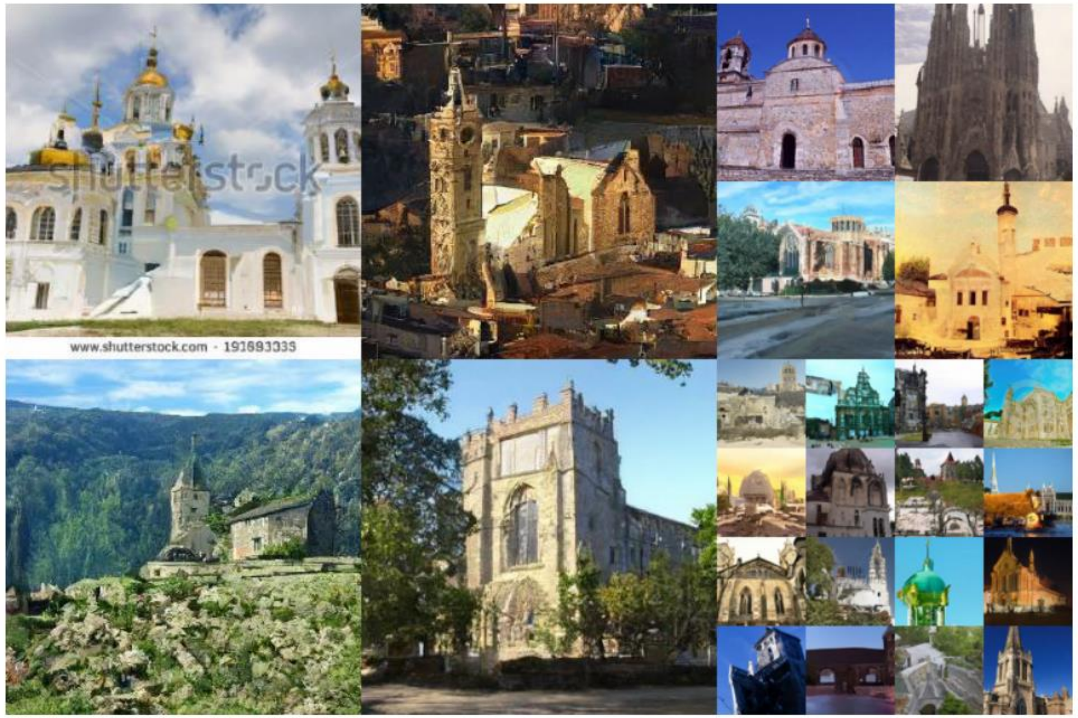
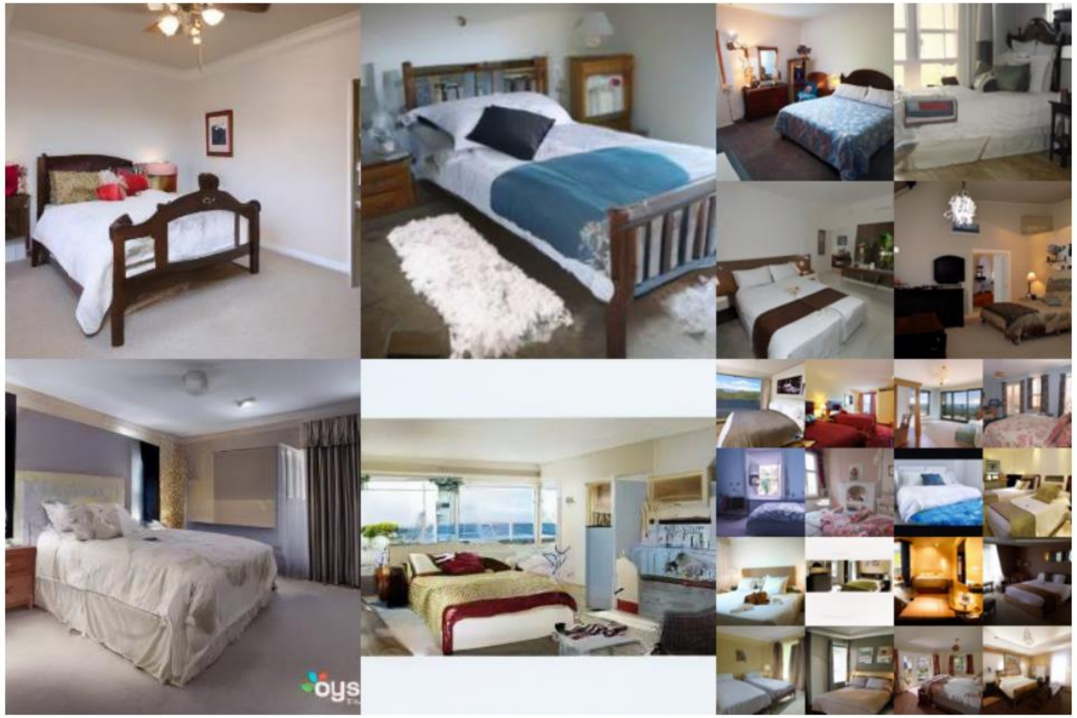


Figure 1: Generated samples on CelebA-HQ 256 × 256 (left) and unconditional CIFAR10 (right)

DDPM 生成效果



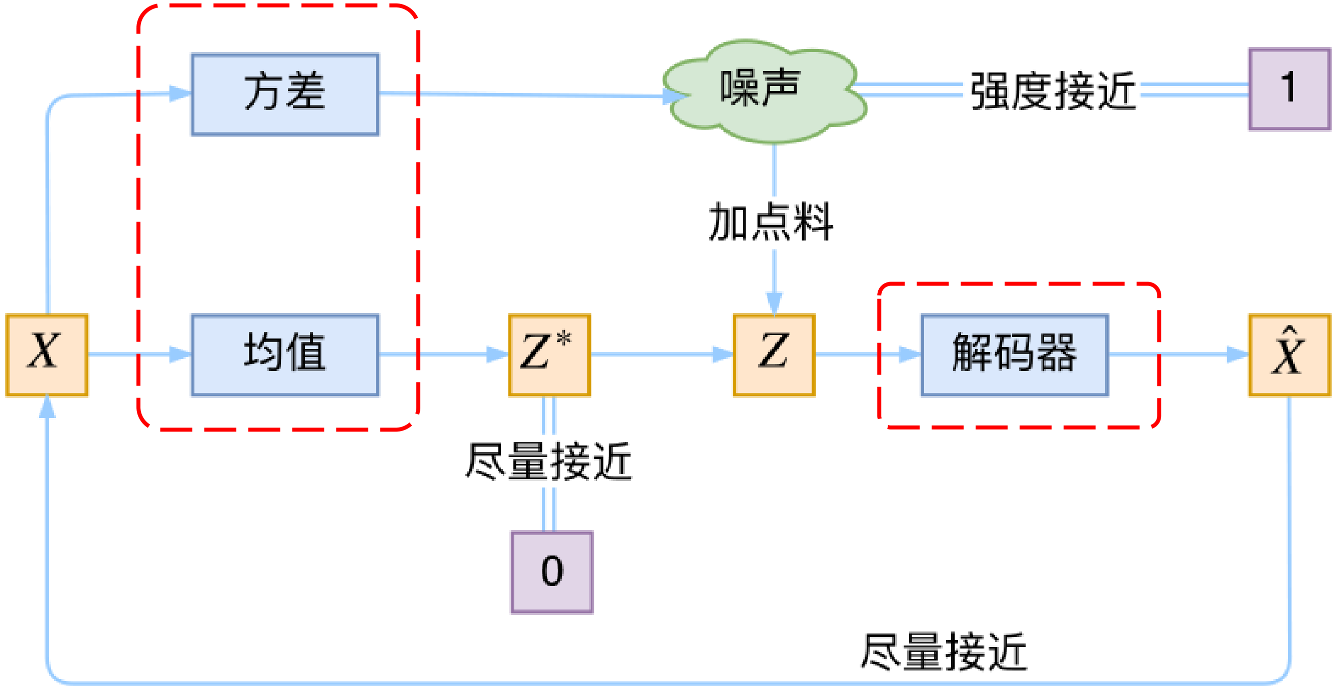


VAE存在的问题

- 从贝叶斯观点来看，VAE同时优化条件分布$p_\theta(x|z)$和变分后验$p(z|x)$

  生成器　　均值方差估计

- 同时优化两个目标导致候选搜索空间极大，难以优化。



VAE存在的问题

- 怎么把变分后验的搞掉?
  - 生成器的目的是为了将标准正态分布的Z变成数据分布的X，变分后验的目的是为了将数据X_k映射到标准正态分布。

? 有没有什么操作能够在不需要优化的情况下，将某个数据分布映射到标准正态分布?

➢ 利用马尔科夫链的平稳分布性质，构造一个平稳分布为标准正态分布的马尔科夫链即可。

$$x = x_0 \to x_1 \to x_2 \to \cdots \to x_{T-1} \to x_T = z$$

- 其中每一小步都可以近似为高斯分布

DDPM 拆楼-建楼

- 生成模型类比

$$
\begin{tikzcd}
\text{随机噪声}\boldsymbol{z} \arrow[r, "\text{变换}"] \arrow[d, "\text{类比}"'] & \text{样本数据}\boldsymbol{x} \arrow[d, "\text{类比}"] \\
\text{砖瓦水泥} \arrow[r, "\text{建设}"] & \text{高楼大厦}
\end{tikzcd}
$$

拆楼： $x = x_0 \to x_1 \to x_2 \to \cdots \to x_{T-1} \to x_T = z$

DDPM 拆楼-建楼

- VAE中的变分后验优化采用了概率论的采样给替代了

$$x = x_0 \rightarrow x_1 \rightarrow x_2 \rightarrow \cdots \rightarrow x_{T-1} \rightarrow x_T = z$$

- 相当于我们得到了很多个pair：$\{x_{t-1}, x_t\}$，如何构造生成器?

➢ 用生成器模拟拆楼的每一步的逆过程，即建楼的每一步!

$$x_t \rightarrow x_{t-1}$$
$$x_{t-1} = \mu(x_t)$$

- 重复这一过程就能够从z一步步采样得到x

DDPM 拆楼

$$x = x_0 \to x_1 \to x_2 \to \cdots \to x_{T-1} \to x_T = z$$

- 对于每一个$x_{t-1} \to x_t$:

$$x_t = \alpha_t x_{t-1} + \beta_t \varepsilon_t, \qquad \varepsilon_t \sim \mathcal{N}(0, \boldsymbol{I})$$

$$\alpha_t, \beta_t > 0 且 \alpha_t^2 + \beta_t^2 = 1$$

$$\begin{aligned}
\boldsymbol{x}_t &= \alpha_t \boldsymbol{x}_{t-1} + \beta_t \boldsymbol{\varepsilon}_t \\
&= \alpha_t \left(\alpha_{t-1} \boldsymbol{x}_{t-2} + \beta_{t-1} \boldsymbol{\varepsilon}_{t-1}\right) + \beta_t \boldsymbol{\varepsilon}_t \\
&= \cdots \\
&= (\alpha_t \cdots \alpha_1) \boldsymbol{x}_0 + \underbrace{(\alpha_t \cdots \alpha_2) \beta_1 \boldsymbol{\varepsilon}_1 + (\alpha_t \cdots \alpha_3) \beta_2 \boldsymbol{\varepsilon}_2 + \cdots + \alpha_t \beta_{t-1} \boldsymbol{\varepsilon}_{t-1} + \beta_t \boldsymbol{\varepsilon}_t}_{\text{多个相互独立的正态噪声之和}}
\end{aligned}$$

DDPM 拆楼

$$\begin{aligned}\boldsymbol{x}_t &= \alpha_t \boldsymbol{x}_{t-1} + \beta_t \boldsymbol{\varepsilon}_t \\ &= \alpha_t\big(\alpha_{t-1}\boldsymbol{x}_{t-2} + \beta_{t-1}\boldsymbol{\varepsilon}_{t-1}\big) + \beta_t \boldsymbol{\varepsilon}_t \\ &= \cdots \\ &= (\alpha_t \cdots \alpha_1)\boldsymbol{x}_0 + \underbrace{(\alpha_t \cdots \alpha_2)\beta_1 \boldsymbol{\varepsilon}_1 + (\alpha_t \cdots \alpha_3)\beta_2 \boldsymbol{\varepsilon}_2 + \cdots + \alpha_t \beta_{t-1}\boldsymbol{\varepsilon}_{t-1} + \beta_t \boldsymbol{\varepsilon}_t}_{\text{多个相互独立的正态噪声之和}}\end{aligned}$$

$$(\alpha_t \cdots \alpha_1)^2 + (\alpha_t \cdots \alpha_2)^2 \beta_1^2 + (\alpha_t \cdots \alpha_3)^2 \beta_2^2 + \cdots + \alpha_t^2 \beta_{t-1}^2 + \beta_t^2 = 1$$

$$\boldsymbol{x}_t = \underbrace{(\alpha_t \cdots \alpha_1)}_{\text{记为}\bar{\alpha}_t}\boldsymbol{x}_0 + \underbrace{\sqrt{1 - (\alpha_t \cdots \alpha_1)^2}}_{\text{记为}\bar{\beta}_t}\bar{\boldsymbol{\varepsilon}}_t, \quad \bar{\boldsymbol{\varepsilon}}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$$

DDPM 建楼

- 学模型$x_{t-1} = \mu(x_t)$，目标为

$$||x_{t-1} - \mu(x_t)||^2$$

- 重参数技巧：根据$x_t = \alpha_t x_{t-1} + \beta_t \varepsilon_t$，将$\mu(x_t)$表示为$\frac{1}{\alpha_t}\left(x_t - \beta_t \epsilon_\theta(x_t, t)\right)$

$$\left\|\boldsymbol{x}_{t-1} - \boldsymbol{\mu}(\boldsymbol{x}_t)\right\|^2 = \frac{\beta_t^2}{\alpha_t^2}\left\|\boldsymbol{\varepsilon}_t - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)\right\|^2$$

- 带入$x_t = \alpha_t x_{t-1} + \beta_t \varepsilon_t = \bar{\alpha}_t x_0 + \alpha_t \bar{\beta}_{t-1} \bar{\varepsilon}_{t-1} + \beta_t \varepsilon_t$

$$\left\|\boldsymbol{\varepsilon}_t - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\bar{\alpha}_t \boldsymbol{x}_0 + \alpha_t \bar{\beta}_{t-1} \bar{\boldsymbol{\varepsilon}}_{t-1} + \beta_t \boldsymbol{\varepsilon}_t, t)\right\|^2$$

DDPM 建楼

$$\|\boldsymbol{x}_{t-1} - \boldsymbol{\mu}(\boldsymbol{x}_t)\|^2 = \frac{\beta_t^2}{\alpha_t^2}\|\boldsymbol{\varepsilon}_t - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t)\|^2$$

- 推导:

$$||x_{t-1} - \mu(x_t)||^2 \qquad (1)$$

由$x_t = \alpha_t x_{t-1} + \beta_t \varepsilon_t$，我们可以设计$\mu(x_t) = \frac{1}{\alpha_t}\left(x_t - \beta_t \varepsilon_\theta(x_t, t)\right)$

将$x_{t-1} = \frac{1}{\alpha_t}(x_t - \beta_t \varepsilon_t)$和$\mu(x_t) = \frac{1}{\alpha_t}\left(x_t - \beta_t \varepsilon_\theta(x_t, t)\right)$代入优化目标(1)，得

$$\left\|\frac{1}{\alpha_t}(x_t - \beta_t \varepsilon_t) - \frac{1}{\alpha_t}\left(x_t - \beta_t \varepsilon_\theta(x_t, t)\right)\right\|^2$$

$$= \left\|\frac{\beta_t}{\alpha_t}\left(\varepsilon_\theta(x_t, t) - \varepsilon_t\right)\right\|^2$$

$$= \frac{\beta_t^2}{\alpha_t^2}||\varepsilon_t - \varepsilon_\theta(x_t, t)||^2$$

DDPM 建楼

- 带入$x_t = \alpha_t x_{t-1} + \beta_t \varepsilon_t = \bar{\alpha}_t x_0 + \alpha_t \bar{\beta}_{t-1} \bar{\varepsilon}_{t-1} + \beta_t \varepsilon_t$，得

$$\left\| \boldsymbol{\varepsilon}_t - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\bar{\alpha}_t \boldsymbol{x}_0 + \alpha_t \bar{\beta}_{t-1} \bar{\boldsymbol{\varepsilon}}_{t-1} + \beta_t \boldsymbol{\varepsilon}_t, t) \right\|^2$$

- 降低方差
  - 上式需要采样4个变量$x_0, \bar{\varepsilon}_{t-1}, \varepsilon_t, t$，会影响优化效果
  - 经过一系列的数学技巧，最终目标函数为

$$\left\| \boldsymbol{\varepsilon} - \frac{\bar{\beta}_t}{\beta_t} \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\bar{\alpha}_t \boldsymbol{x}_0 + \bar{\beta}_t \boldsymbol{\varepsilon}, t) \right\|^2$$

DDPM 建楼

- 一系列的数学技巧： $\left\|\boldsymbol{x}_{t-1}-\boldsymbol{\mu}(\boldsymbol{x}_t)\right\|^2=\frac{\beta_t^2}{\alpha_t^2}\left\|\boldsymbol{\varepsilon}_t-\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t)\right\|^2$ (2)

继续简化(2)式：

利用22页的计算展开$x_t=\bar{\alpha}_t x_0+\bar{\beta}_t\bar{\varepsilon}_t$，但(2)中已采样了$\varepsilon_t$，与$\bar{\varepsilon}_t$不独立，将$x_t$倒退一步：

$$x_t=\alpha_t x_{t-1}+\beta_t\varepsilon_t=\bar{\alpha}_t x_0+\alpha_t\bar{\beta}_{t-1}\bar{\varepsilon}_{t-1}+\beta_t\varepsilon_t$$

代入(2)，忽略常数项，得：

$$\left\|\boldsymbol{\varepsilon}_t-\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\bar{\alpha}_t\boldsymbol{x}_0+\alpha_t\bar{\beta}_{t-1}\bar{\boldsymbol{\varepsilon}}_{t-1}+\beta_t\boldsymbol{\varepsilon}_t,t)\right\|^2$$

DDPM 建楼

- 一系列的数学技巧： $\left\|\boldsymbol{\varepsilon}_t - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\bar{\alpha}_t \boldsymbol{x}_0 + \alpha_t \bar{\beta}_{t-1} \bar{\boldsymbol{\varepsilon}}_{t-1} + \beta_t \boldsymbol{\varepsilon}_t, t)\right\|^2$ (3)

如何降低采样方差?

目的是希望将采样的两个随机变量$\varepsilon_t, \bar{\varepsilon}_{t-1}$变成一个
根据正态分布的叠加性，我们有以下结论：

$$\alpha_t \bar{\beta}_{t-1} \bar{\varepsilon}_{t-1} + \beta_t \varepsilon_t \text{等价于} \bar{\beta}\varepsilon, \varepsilon \sim \mathcal{N}(0, \boldsymbol{I})$$

$$\beta_t \bar{\varepsilon}_{t-1} - \alpha_t \bar{\beta}_{t-1} \varepsilon_t \text{等价于} \bar{\beta}\omega, \omega \sim \mathcal{N}(0, \boldsymbol{I})$$

证明：
先看第一个结论，两边都是正态分布，且均值为0，计算方差是否相等：

$$(\alpha_t \bar{\beta}_{t-1})^2 + \beta_t^2 = \alpha_t^2(1 - \bar{\alpha}_{t-1}^2) + \beta_t^2 = \alpha_t^2 - \alpha_t^2 \bar{\alpha}_{t-1}^2 + \beta_t^2 = 1 - \bar{\alpha}_t^2 = \bar{\beta}_t^2$$

意味着方差也相同，结论得证。第二个结论类似。

DDPM 建楼

- 一系列的数学技巧： $\left\|\boldsymbol{\varepsilon}_t - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\bar{\alpha}_t \boldsymbol{x}_0 + \alpha_t \bar{\beta}_{t-1} \bar{\boldsymbol{\varepsilon}}_{t-1} + \beta_t \boldsymbol{\varepsilon}_t, t)\right\|^2$ (3)

根据上一页的结论，我们可以得到一个二元一次方程组：

$$\begin{cases} \alpha_t \bar{\beta}_{t-1} \bar{\varepsilon}_{t-1} + \beta_t \varepsilon_t = \bar{\beta} \varepsilon \\ \beta_t \bar{\varepsilon}_{t-1} - \alpha_t \bar{\beta}_{t-1} \varepsilon_t = \bar{\beta} \omega \end{cases}$$

将ε_t解出来，得：

$$\varepsilon_t = \frac{\beta_t \varepsilon - \alpha_t \bar{\beta}_{t-1} \omega}{\bar{\beta}_t}$$

带入式(3)，由于训练过程是不断sample随机变量，所以我们将总的损失函数用期望表示：

$$E_{\varepsilon_t, \bar{\varepsilon}_{t-1} \sim \mathcal{N}(0, \boldsymbol{I})} \left[\left\| \varepsilon_t - \varepsilon_\theta \left(\bar{\alpha}_t x_0 + \alpha_t \bar{\beta}_{t-1} \bar{\varepsilon}_{t-1} + \beta_t \varepsilon_t, t \right) \right\|^2 \right]$$

$$= E_{\varepsilon_t, \bar{\varepsilon}_{t-1} \sim \mathcal{N}(0, \boldsymbol{I})} \left[\left\| \frac{\beta_t \varepsilon - \alpha_t \bar{\beta}_{t-1} \omega}{\bar{\beta}_t} - \varepsilon_\theta \left(\bar{\alpha}_t x_0 + \bar{\beta}_t \varepsilon_t, t \right) \right\|^2 \right]$$

DDPM 建楼

- 一系列的数学技巧： $\left\|\boldsymbol{\varepsilon}_t - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\bar{\alpha}_t\boldsymbol{x}_0 + \alpha_t\bar{\beta}_{t-1}\bar{\boldsymbol{\varepsilon}}_{t-1} + \beta_t\boldsymbol{\varepsilon}_t, t)\right\|^2$ (3)

由于$\omega\sim\mathcal{N}(0,\boldsymbol{I})$，且与$\theta$模型无关，可以展开直接算出$\omega$相关项的期望，是一些常数。所以有

$$E_{\varepsilon_t,\bar{\varepsilon}_{t-1}\sim\mathcal{N}(0,\boldsymbol{I})}\left[\left\|\frac{\beta_t\varepsilon - \alpha_t\bar{\beta}_{t-1}\omega}{\bar{\beta}_t} - \varepsilon_\theta\left(\bar{\alpha}_t x_0 + \bar{\beta}_t\varepsilon, t\right)\right\|^2\right]$$

$$= E_{\varepsilon\sim\mathcal{N}(0,\boldsymbol{I})}\left[\left\|\beta_t\varepsilon - \varepsilon_\theta\left(\bar{\alpha}_t x_0 + \bar{\beta}_t\varepsilon, t\right)\right\|^2\right]$$

变化一下就得到了，再去除常数系数，可得最终损失函数：

$$E_{\varepsilon\sim\mathcal{N}(0,\boldsymbol{I})}\left[\left\|\varepsilon - \frac{\bar{\beta}_t}{\beta_t}\varepsilon_\theta\left(\bar{\alpha}_t x_0 + \bar{\beta}_t\varepsilon, t\right)\right\|^2\right]$$

DDPM 损失函数直观理解

- 初始损失函数：$\|\boldsymbol{x}_{t-1}-\boldsymbol{\mu}(\boldsymbol{x}_t)\|^2=\frac{\beta_t^2}{\alpha_t^2}\|\boldsymbol{\varepsilon}_t-\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t)\|^2$ (2)
- 递推到x_0的损失函数：$\left\|\boldsymbol{\varepsilon}_t-\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\bar{\alpha}_t\boldsymbol{x}_0+\alpha_t\bar{\beta}_{t-1}\bar{\boldsymbol{\varepsilon}}_{t-1}+\beta_t\boldsymbol{\varepsilon}_t,t)\right\|^2$ (3)
- 简化后的损失函数：$\left\|\boldsymbol{\varepsilon}-\frac{\bar{\beta}_t}{\beta_t}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\bar{\alpha}_t\boldsymbol{x}_0+\bar{\beta}_t\boldsymbol{\varepsilon},t)\right\|^2$ (4)

- 直观的理解还是需要带入“拆楼+建楼”的过程，对于拆楼的每一对$\{x_{t-1}\rightarrow x_t\}$，我们希望用生成器$x_{t-1}=\mu(x_t)$模拟逆过程$\{x_t\rightarrow x_{t-1}\}$。
- 但根据$x_t=\alpha_t x_{t-1}+\beta_t\varepsilon_t$，这里面$x_t,\alpha_t,\beta_t$都是已知的，$\varepsilon_t$是未知的。直接让模型重构图片也许不太容易，但让模型拟合一个服从正态分布的噪声是不是容易得多?
- 我们知道x_t可以理解为x_{t-1}添加了一小点“料”（随机噪声ε_t），只要我们能拟合第t步采样的噪声ε_t，就可以根据公式还原$\{x_t\rightarrow x_{t-1}\}$这一建楼过程。相当于我们期望模型拟合的是第$t$步加了什么“料”，然后可以知道我们在第$t$步去噪（建楼）应该减去什么噪声。于是可以理解式(3)中重参数技巧的含义：
  - 尽管式(3)中的ε_t是随机采样得到的噪声，但它的含义应该是模型在拆楼的第t步中加的那个特定的“料”，模型想要去拟合的恰好是第t步的噪声ε_t。

DDPM 损失函数直观理解

- 初始损失函数：$\left\|\boldsymbol{x}_{t-1}-\boldsymbol{\mu}(\boldsymbol{x}_t)\right\|^2=\frac{\beta_t^2}{\alpha_t^2}\left\|\boldsymbol{\varepsilon}_t-\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{x}_t,t)\right\|^2$ (3)
- 递推到x_0的损失函数：$\left\|\boldsymbol{\varepsilon}_t-\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\bar{\alpha}_t\boldsymbol{x}_0+\alpha_t\bar{\beta}_{t-1}\bar{\boldsymbol{\varepsilon}}_{t-1}+\beta_t\boldsymbol{\varepsilon}_t,t)\right\|^2$ (4)
- 简化后的损失函数：$\left\|\boldsymbol{\varepsilon}-\frac{\bar{\beta}_t}{\beta_t}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\bar{\alpha}_t\boldsymbol{x}_0+\bar{\beta}_t\boldsymbol{\varepsilon},t)\right\|^2$ (5)

- 至于为什么可以去sample任意的t，因为任意一个x_t的表达式都可以推导为仅依赖x_0的形式。
- 于是每一步的训练我们只需要去采样t,x_0,ε即可。

DDPM 生成流程

- 从一个随机噪声$x_T \sim \mathcal{N}(0, \boldsymbol{I})$出发执行$T$步

$$x_{t-1} = \frac{1}{\alpha_t}\left(x_t - \beta_t \epsilon_\theta(x_t, t)\right)$$

- Random Sampling:

$$x_{t-1} = \frac{1}{\alpha_t}\left(x_t - \beta_t \epsilon_\theta(x_t, t)\right) + \sigma_t z, z \sim \mathcal{N}(0, \boldsymbol{I})$$

- 通常$\sigma_t = \beta_t$，使正向和反向的方差保持一致

DDPM 超参设置

- $T = 1000$，生成过程是串行的，采样速度是DDPM的一个瓶颈
- α_t: $\sqrt{1 - \frac{0.02t}{T}}$ 单调递减

$$\log \bar{\alpha}_T = \sum_{t=1}^{T} \log \alpha_t = \frac{1}{2} \sum_{t=1}^{T} \log\left(1 - \frac{0.02t}{T}\right) < \frac{1}{2} \sum_{t=1}^{T} \left(-\frac{0.02t}{T}\right) = -0.005(T+1)$$

  - $T = 1000, \bar{\alpha}_t \approx e^{-5}$，符合之前所说的要求

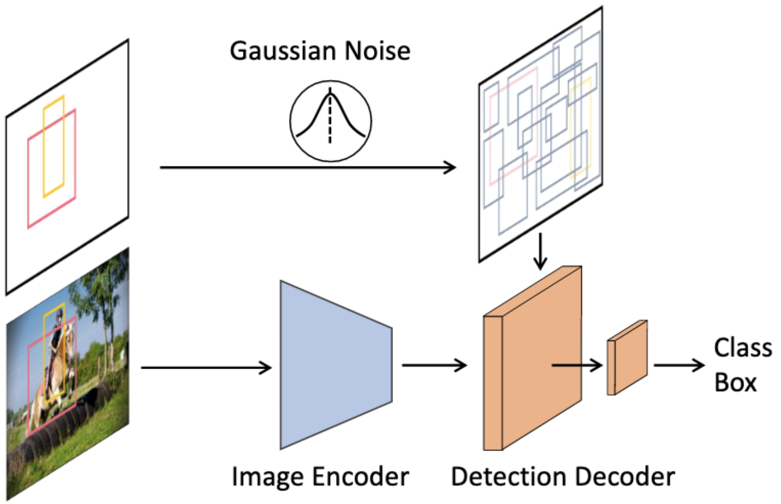
DDPM 总结

- 优点：训练简单
  - 从VAE中难以优化的变分后验出发，利用马尔科夫过程直接减少了一个优化维度。
  - 每一步“拆楼”的过程近似为一个高斯分布采样，让模型学习逆过程的难度大大降低。
  - 过程中仍然会遇到优化困难的点，但可以借助概率推导降低优化复杂度。
- 缺点是采样时间过长

Detection -> DiffusionDet

Q: Is there a simpler approach that does not even need the surrogate of learnable queries
A: Detect objects from random boxes
noise-to-box
Training:
gaussian noise added to gt -> noise boxes -> crop ROI features -> decoder for predict
Inference:
random boxes

1. Dynamic boxes
2. Progressive refinement



Detection -> DiffusionDet

Algorithm 1 DiffusionDet Training

```
def train_loss(images, gt_boxes):
  """
  images: [B, H, W, 3]
  gt_boxes: [B, *, 4]
  # B: batch
  # N: number of proposal boxes
  """

  # Encode image features
  feats = image_encoder(images)

  # Pad gt_boxes to N
  pb = pad_boxes(gt_boxes) # padded boxes: [B, N, 4]

  # Signal scaling
  pb = (pb * 2 - 1) * scale

  # Corrupt gt_boxes
  t = randint(0, T)           # time step
  eps = normal(mean=0, std=1) # noise: [B, N, 4]
  pb_crpt = sqrt(    alpha_cumprod(t)) * pb +
            sqrt(1 - alpha_cumprod(t)) * eps

  # Predict
  pb_pred = detection_decoder(pb_crpt, feats, t)

  # Set prediction loss
  loss = set_prediction_loss(pb_pred, gt_boxes)

  return loss
```

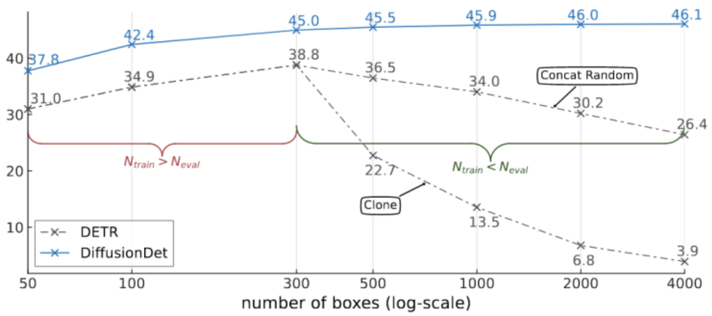
`alpha_cumprod(t)`: cumulative product of α_i, *i.e.*, $\prod_{i=1}^{t} \alpha_i$

Algorithm 2 DiffusionDet Sampling

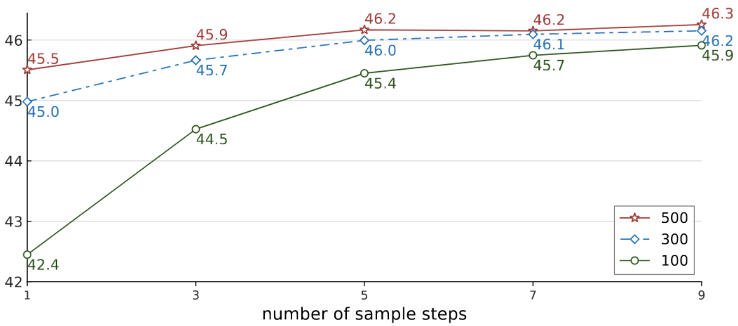
```
def infer(images, steps, T):
  """
  images: [B, H, W, 3]
  # steps: number of sample steps
  # T: number of time steps
  """

  # Encode image features
  feats = image_encoder(images)

  # noisy boxes: [B, N, 4]
  pb_t = normal(mean=0, std=1)

  # uniform sample step size
  times = reversed(linespace(-1, T, steps))

  # [(T-1, T-2), (T-2, T-3), ..., (1, 0), (0, -1)]
  time_pairs = list(zip(times[:-1], times[1:])

  for t_now, t_next in zip(time_pairs):
    # Predict pb_0 from pb_t
    pb_pred = detection_decoder(pb_t, feats, t_now)

    # Estimate pb_t at t_next
    pb_t = ddim_step(pb_t, pb_pred, t_now, t_next)

    # Box renewal
    pb_t = box_renewal(pb_t)

  return pb_pred
```

`linespace`: generate evenly spaced values

Detection -> DiffusionDet



(a) **Dynamic boxes.** Both DETR and DiffusionDet are trained with 300 object queries or proposal boxes. More proposal boxes in inference bring accuracy improvement on DiffusionDet, while degenerate DETR.



(b) **Progressive refinement.** DiffusionDet is trained with 300 proposal boxes and evaluated with different numbers of proposal boxes. For all cases, the accuracy increases with refinement times.

Method	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
ResNet-50 [34]						
RetinaNet [93]	38.7	58.0	41.5	23.3	42.3	50.3
Faster R-CNN [93]	40.2	61.0	43.8	24.2	43.5	52.0
Cascade R-CNN [93]	44.3	62.2	48.0	26.6	47.7	57.7
DETR [10]	42.0	62.4	44.2	20.5	45.8	61.1
Deformable DETR [102]	43.8	62.6	47.7	26.4	47.1	58.0
Sparse R-CNN [81]	45.0	63.4	48.2	26.9	47.2	59.5
DiffusionDet (1 step)	45.5	65.1	48.7	27.5	48.1	61.2
DiffusionDet (4 step)	46.1	66.0	49.2	28.6	48.5	61.3
DiffusionDet (8 step)	**46.2**	**66.4**	**49.5**	**28.7**	**48.5**	**61.5**
ResNet-101 [34]						
RetinaNet [93]	40.4	60.2	43.2	24.0	44.3	52.2
Faster R-CNN [93]	42.0	62.5	45.9	25.2	45.6	54.6
Cascade R-CNN [11]	45.5	63.7	49.9	27.6	49.2	59.1
DETR [10]	43.5	63.8	46.4	21.9	48.0	61.8
Sparse R-CNN [81]	46.4	64.6	49.5	28.3	48.3	61.6
DiffusionDet (1 step)	46.6	66.3	50.0	30.0	49.3	62.8
DiffusionDet (4 step)	46.9	66.8	50.4	**30.6**	49.5	62.6
DiffusionDet (8 step)	**47.1**	**67.1**	**50.6**	30.2	**49.8**	**62.7**
Swin-Base [54]						
Cascade R-CNN [54]	51.9	70.9	56.5	35.4	55.2	67.4
Sparse R-CNN	52.0	72.2	57.0	35.8	55.1	68.2
DiffusionDet (1 step)	52.3	72.7	56.3	34.8	56.0	68.5
DiffusionDet (4 step)	52.7	73.5	56.8	36.1	56.0	**68.9**
DiffusionDet (8 step)	**52.8**	**73.6**	**56.8**	**36.1**	**56.2**	68.8

Method	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l	AP_r	AP_c	AP_f
ResNet-50 [34]									
Faster R-CNN†	22.5	37.1	23.6	16.5	29.6	34.9	9.9	21.1	29.7
Cascade R-CNN†	26.3	37.8	27.8	18.4	34.4	41.9	12.3	24.9	34.1
Faster R-CNN	25.2	40.6	26.9	18.5	32.2	37.7	16.4	23.4	31.1
Cascade R-CNN	29.4	41.4	30.9	20.6	37.5	44.3	20.0	27.7	35.4
Sparse R-CNN	29.2	41.0	30.7	20.7	36.9	44.2	20.6	27.7	34.6
DiffusionDet (1 step)	30.4	42.8	31.8	20.6	38.6	47.6	23.5	28.1	36.0
DiffusionDet-(4 step)	31.8	45.0	33.2	22.5	39.9	48.3	24.8	29.3	37.6
DiffusionDet-(8 step)	**31.9**	**45.3**	**33.1**	**22.8**	**40.2**	**48.1**	**24.0**	**29.5**	**38.1**
ResNet-101 [34]									
Faster R-CNN†	24.8	39.8	26.1	17.9	32.2	36.9	13.7	23.1	31.5
Cascade R-CNN†	28.6	40.1	30.1	19.8	37.1	43.8	15.3	27.3	35.9
Faster R-CNN	27.2	42.9	29.1	20.3	35.0	40.4	18.8	25.4	33.0
Cascade R-CNN	31.6	43.8	33.4	22.3	39.7	47.3	23.9	29.8	37.0
Sparse R-CNN	30.1	42.0	31.9	21.3	38.5	45.6	23.5	27.5	35.9
DiffusionDet (1 step)	31.9	44.6	33.1	21.6	40.3	49.0	23.4	30.5	37.1
DiffusionDet-(4 step)	32.9	46.5	34.3	23.3	41.1	49.9	24.2	31.3	38.6
DiffusionDet-(8 step)	**33.5**	**47.3**	**34.7**	**23.6**	**41.9**	**49.8**	**24.8**	**32.0**	**39.0**
Swin-Base [54]									
DiffusionDet-(1 step)	40.6	54.8	42.7	28.3	50.0	61.6	33.6	39.8	44.6
DiffusionDet-(4 step)	41.9	57.1	44.0	30.3	50.6	62.3	34.9	40.7	46.3
DiffusionDet-(8 step)	**42.1**	**57.8**	**44.3**	**31.0**	**51.3**	**62.5**	**34.3**	**41.0**	**46.7**

Table 2. **Comparisons with different object detectors on LVIS v1.0 `val` set.** We re-implement all detectors using federated loss [100] except for the rows in light gray (with †).

scale	AP	AP_{50}	AP_{75}
0.1	38.5	54.2	41.4
1.0	44.3	63.2	47.6
2.0	45.0	64.3	48.1
3.0	44.8	63.9	48.2

(a) **Signal scale.** A large scaling factor can improve detection performance.

case	AP	AP_{50}	AP_{75}
Repeat	43.7	62.6	47.0
Cat Gaussian	45.0	64.3	48.1
Cat Uniform	44.7	63.7	48.3
Cat Full	44.8	63.9	47.9

(b) **GT boxes padding.** Concatenating Gaussian boxes works best.

DDIM	box renewal	step 1	step 4	step 8
		45.0	43.4	43.4
✓		45.0	45.5	45.4
	✓	45.0	45.6	45.6
✓	✓	45.0	45.8	46.2

(c) **Sampling strategy.** Using both DDIM and box renewal works best.

score thresh.	AP	AP_{50}	AP_{75}
0.0	45.4	65.2	48.8
0.3	45.9	65.7	49.2
0.5	46.2	66.1	49.4
0.7	46.0	66.2	49.0

(d) **Box renewal** at evaluation of step 8. The threshold of 0.5 works best.

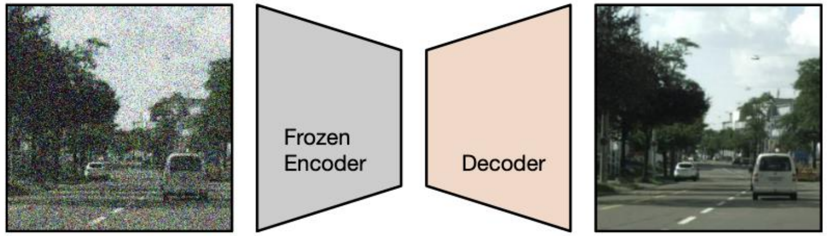
eval \ train	100	300	500
100	<u>42.5</u>	42.4	41.1
300	43.8	<u>45.0</u>	44.8
500	44.2	45.5	<u>45.6</u>
1000	44.5	45.9	<u>46.1</u>

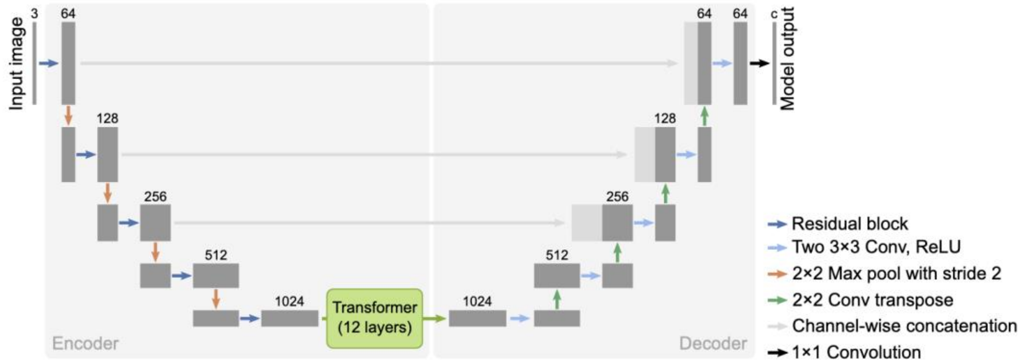
(e) **Matching between N_{train} and N_{eval}.** The best for each row is underlined.

# boxes	step	AP	AP_{50}	AP_{75}	FPS
[81]	1	45.0	63.4	48.2	31.4
100	1	42.5	60.3	45.9	31.6
300	1	45.0	64.3	48.1	31.3
300	4	45.8	65.7	49.2	12.4

(f) **Accuracy vs. speed.** Using more boxes bring performance gain at the cost of latency.

Seg - > denoising pretrain for seg





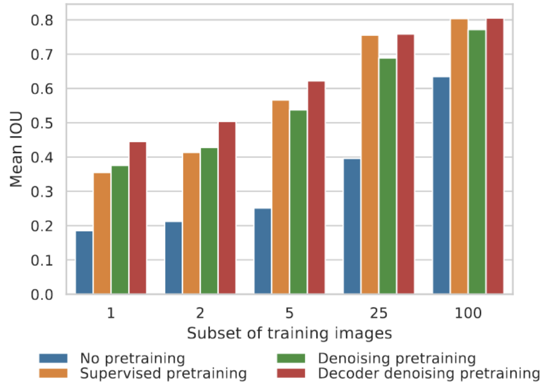


Table 2. Comparison with the state-of-the-art on Cityscapes. The result of [32] is reproduced by [108] based on DeepLab-v3+, while the results of [29, 44, 58, 63] are based on DeepLab-v2. All of the baselines except ours make use of a ResNet-101 backbone.

Method	full (2,975)	1/4 (744)	1/8 (372)	1/30 (100)
AdvSemSeg [44]	-	62.3	58.8	-
s4GAN [58]	65.8	61.9	59.3	-
DMT [29]	68.16	-	63.03	54.80
ClassMix [63]	-	63.63	61.35	-
CutMix [32]	-	68.33	65.82	55.71
PseudoSeg [108]	-	72.36	69.81	60.96
Sup. baseline [104]	74.88	73.31	68.72	56.09
PC^2Seg [104]	75.99	75.15	72.29	62.89
DDeP (Ours)	**80.62**	**76.26**	**72.99**	**63.25**

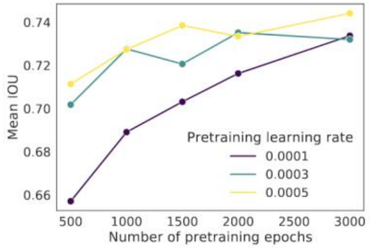
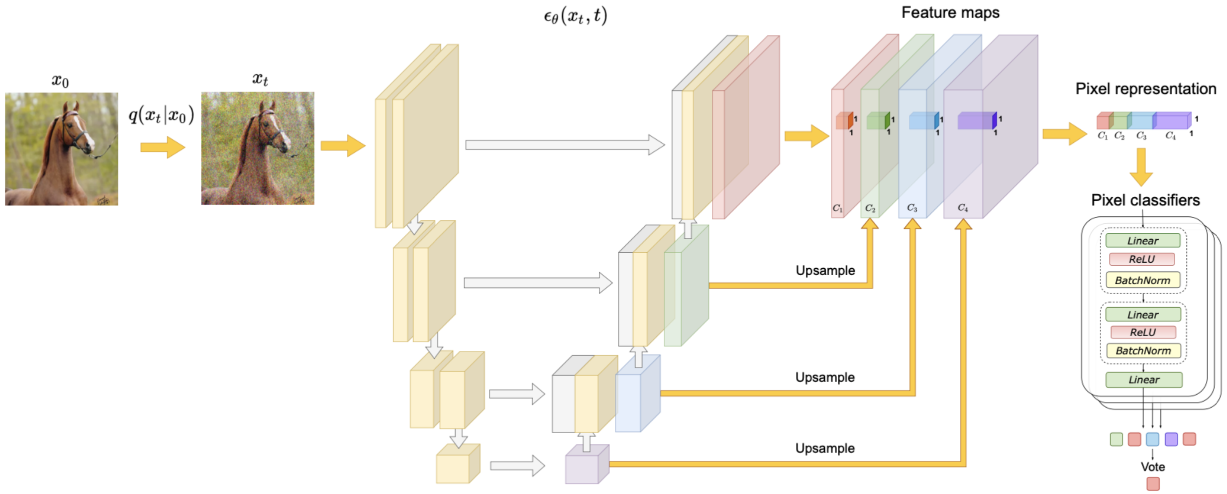


Figure 8. Effect of length of pretraining duration on downstream performance.

Seg - > ICLR 22



Few-shot segmentation

1. Train unsupervised diffusion model
2. Extract pixel-level representation of labeled
3. Concat reshape features
4. MLP for predict

DDPM 总结

- 目前主要研究方向：
  - 综述：
    - Diffusion Models: A Comprehensive Survey of Methods and Applications. https://arxiv.org/abs/2209.00796
  - 降低diffusion模型的采样步长
    - Analytic-DPM. https://arxiv.org/abs/2201.06503
    - DPM-Solver. https://arxiv.org/abs/2206.00927
    - On Fast Sampling of Diffusion Probabilistic Models. https://arxiv.org/pdf/2106.00132.pdf
  - 探索guided diffusion
    - Diffusion Models Beat GANs on Image Synthesis. https://proceedings.neurips.cc/paper/2021/file/49ad23d1ec9fa4bd8d77d02681df5cfa-Paper.pdf
    - More Control for Free! Image Synthesis with Semantic Diffusion Guidance. https://arxiv.org/pdf/2112.05744
    - Classifier-Free Diffusion Guidance. https://arxiv.org/pdf/2207.12598.pdf
  - 大模型、大语料
    - GLIDE. https://arxiv.org/pdf/2112.10741.pdf
  - 用于NLP生成
    - Diffusion-LM Improves Controllable Text Generation. https://arxiv.org/abs/2205.14217