教育部-华为智能基座课程

《人工智能基础与实践》

第12章：生成式人工智能

授课教师：丛润民

山东大学

控制科学与工程学院

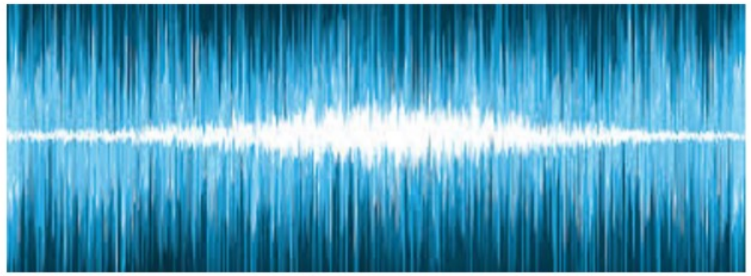
章节目录 CONTENTS

章节目录 CONTENTS

背景介绍

- 什么是生成式人工智能?

Computer Vision



Computational Speech

Natural Language Processing

Computer Vision/robotics

背景介绍

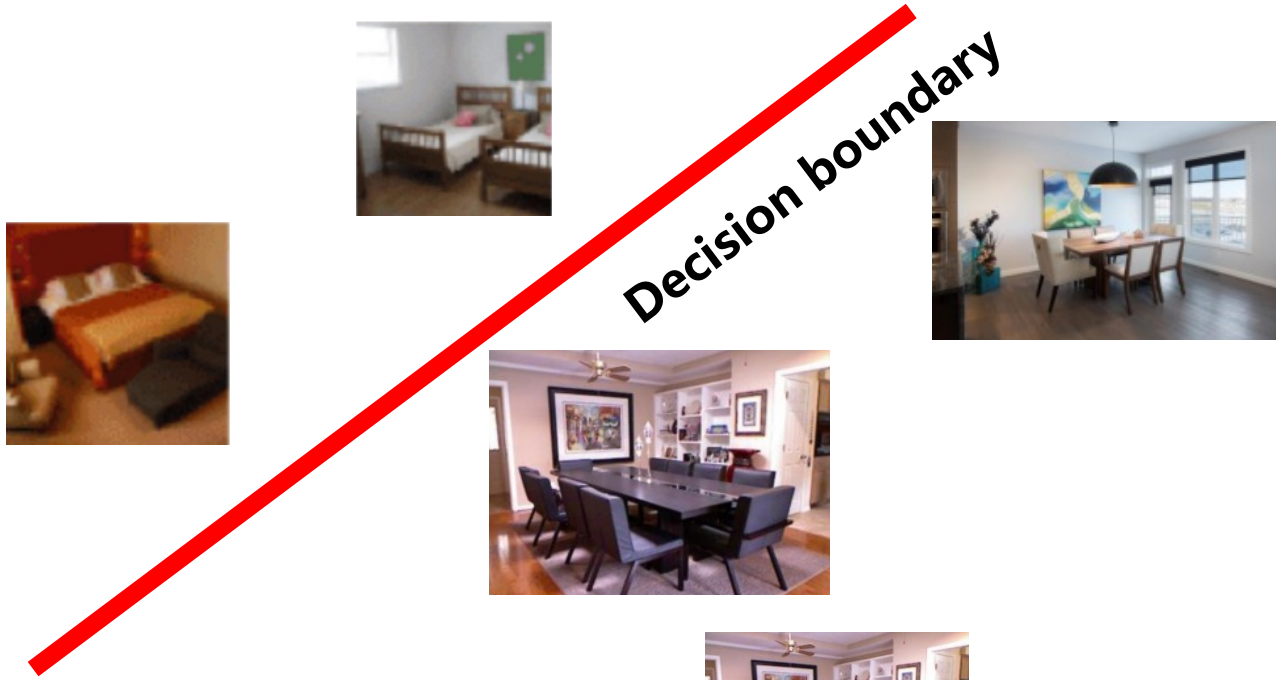
- 什么是生成式人工智能?

Richard Feynman: What I cannot create, I do not understand

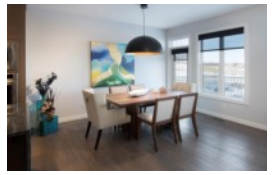
今天来说： *What I understand, I can create*

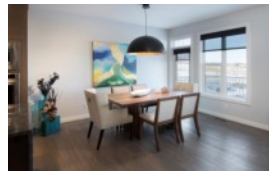
背景介绍

- 生成式 VS 判别式



$P(Y = \text{Bedroom} \mid X =$  $) = 0.01$

$Y = B,$ 

$Y = D,$ 

$Y = B,$ 

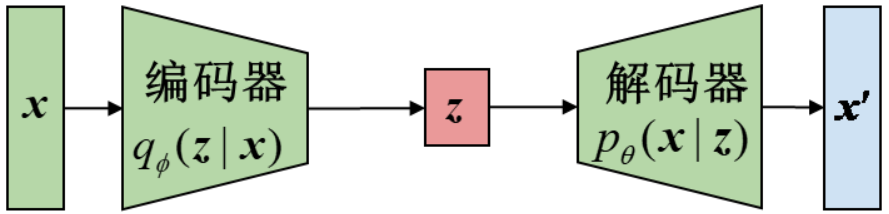
$Y = D,$ 

$P(Y = \text{Bedroom}, X =$  $)$

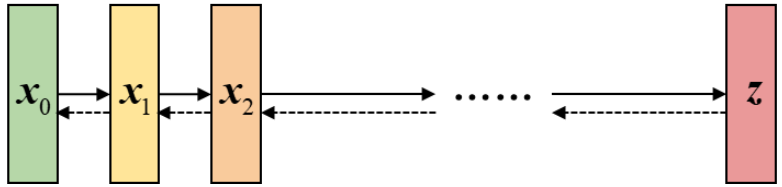
背景介绍

➢ 典型生成模型的结构示意图

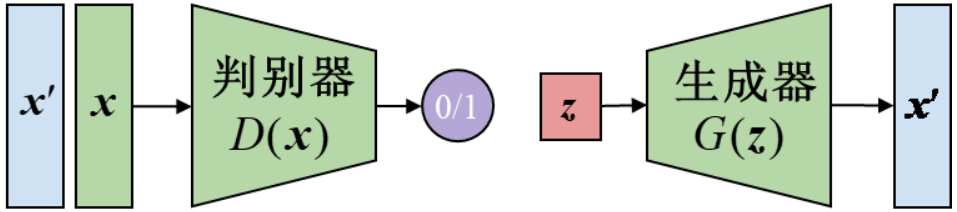
变分自编码器



扩散模型



生成对抗网络



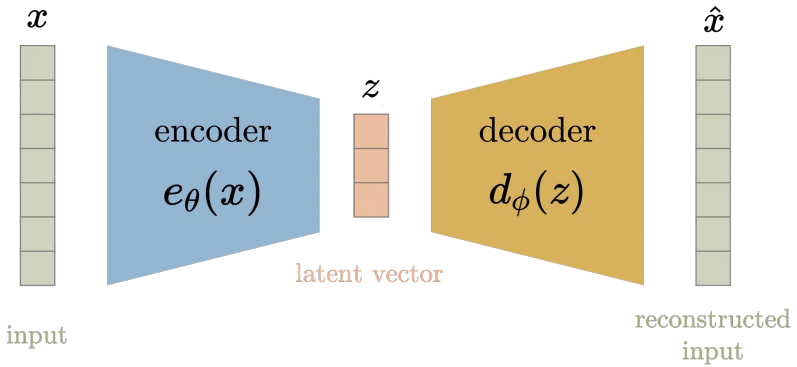
章节目录 CONTENTS

传统自编码器 AE (Auto Encoder)

➢ 工作机制

编码器 fenc 将高维输入 x ∈ $\mathbb{R}_n$ 映射为低维空间上的点 z

$$x \to z = f_{enc}(x) \to \hat{x} = f_{dec}(z)$$

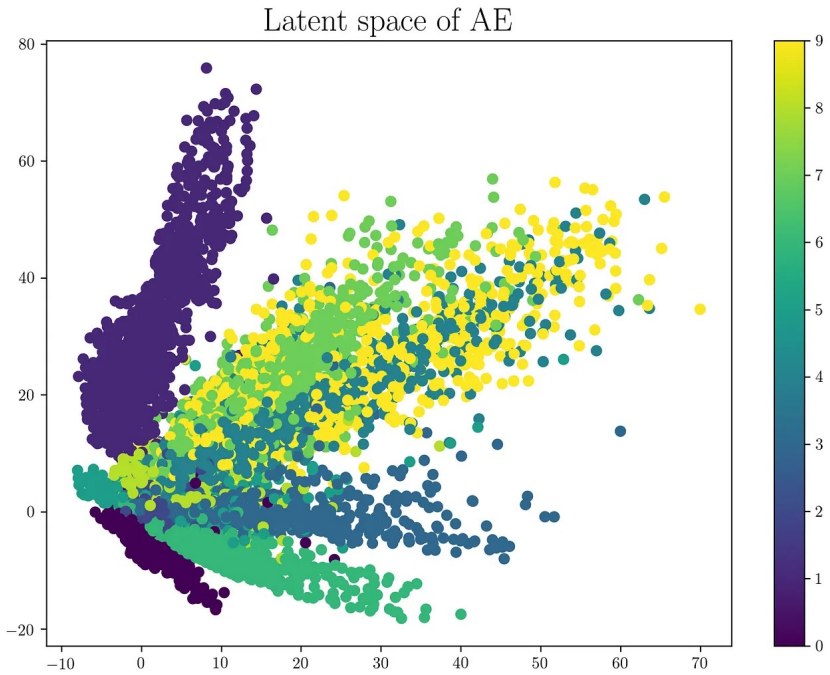


➢ AE的目标

仅仅是"记住"训练集数据的压缩表示，以最小化重构误差：

$$loss = ||x - \hat{x}||_2 = ||x - d_\phi(z)||_2 = ||x - d_\phi(e_\theta(x))||_2$$

➢ 几何可视化



高维空间中的数据流形与低维潜在空间映射

- AE是优秀的压缩器，但不是生成器：仅重构见过的数据，无法创造有意义的新样本

➢ 引入概率分布 (Probabilistic Mapping)

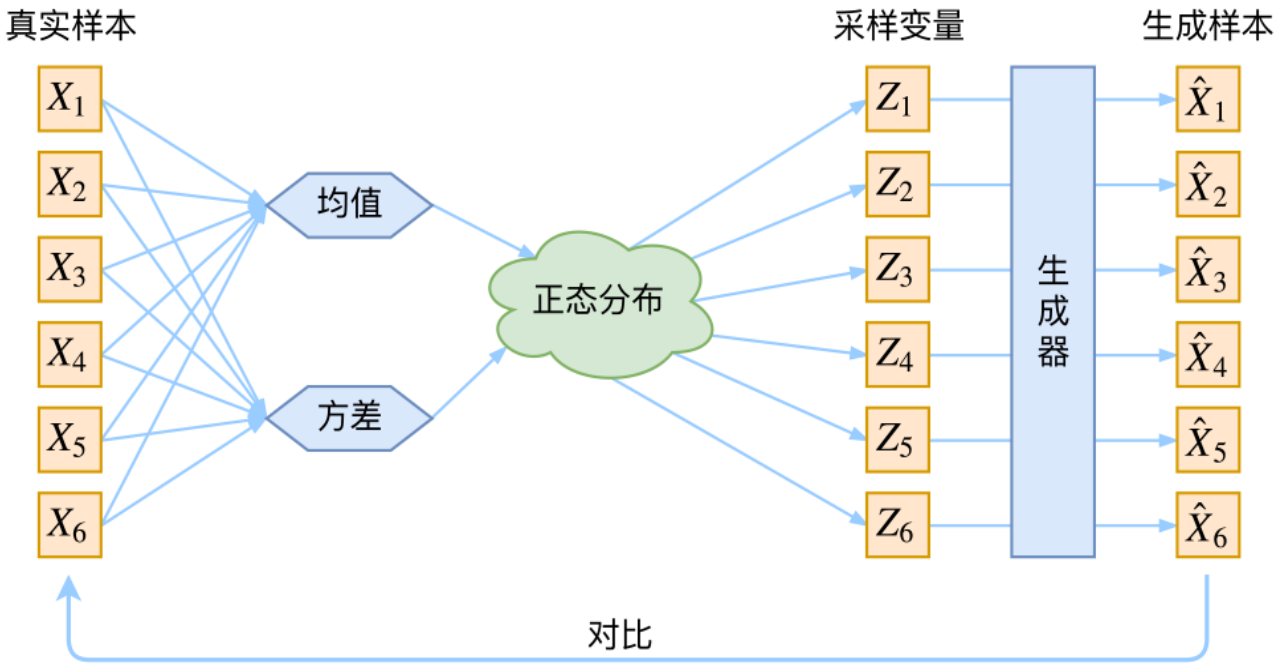
VAE核心思想

将"确定性的点"替换为"概率性的区域"

不再输出单一坐标，而是输出完整概率分布

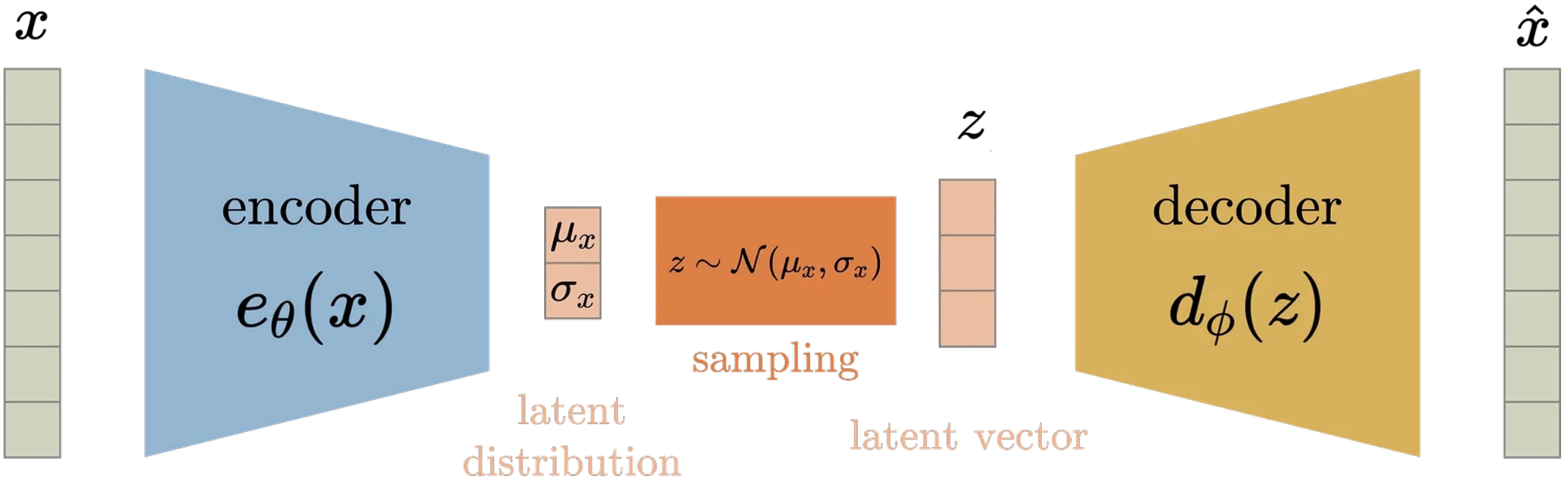
具体操作

1. **编码器输出分布参数**
   不再输出坐标 z，而是输出 $\mathbf{q\varphi(z|x)} = \mathbf{N(z;\ \mu, \sigma^2)}$
2. **映射为云团**
   输入 x 被映射为潜在空间中的概率性区域
3. **从分布中采样**
   从该分布中采样 $z \sim q\varphi(z|x)$，再由解码器生成



- **范式转变的意义：**通过引入概率分布，为潜在空间带来连续性和可采样性。

VAE 架构



编码器（Encoder）：编码器是一个神经网络，接收输入数据 x，输出的不是一个确定的潜在向量 z，而是潜在空间中一个概率分布的参数，通常是高斯分布的均值 μ 和对数方差 $\log(\sigma^2)$。这种将输入映射到分布参数的方式，是VAE概率特性的根本来源。

解码器（Decoder）：解码器是另一个神经网络，它接收从潜在空间采样得到的向量 z 作为输入，并尽力重构出原始的输入数据 x′。

变分推断

1. 生成过程假设

先验 (Prior)：潜在变量服从标准正态分布 $p(z) = N(0, I)$

生成 (Generation)：数据由解码器生成 $p(x|z)$

2. 推断难题 (The Inference Problem)

目标是求后验分布 $p(z|x)$。根据贝叶斯公式：

$$p(z|x) = p(x|z)p(z) / p(x)$$

⚠ **痛点：分母 p(x) 的积分在高维空间中计算不可行 (Intractable)。**

3. 解决方案：变分推断 (VI)

引入一个易于处理的近似分布 $q_\varphi(z/x)$ （即编码器神经网络）。

目标：优化参数 φ，使近似分布尽可能接近真实后验（最小化 KL 散度）。

重参数

➢ 重参数技巧（Reparameterization Trick)

VAE的目标是从编码器定义的分布 $q(z|x)$ 中采样。然而，直接从一个随机节点进行采样会导致梯度无法反向传播，从而无法训练整个模型。“重参数技巧”优雅地解决了这个问题。它将随机采样过程分离出来，**通过 $z = \mu + \sigma * \varepsilon$ 来生成潜在向量 z**，其中 ε 是从标准正态分布（均值为0，方差为1）中采样的噪声。如此一来，模型的随机性来自于外部的 ε，**而 μ 和 σ 仍然是编码器网络的可学习参数**，梯度得以顺利传播

损失函数

$$reconstruction\ loss = ||x - \hat{x}||_2 = ||x - d_\phi(z)||_2 = ||x - d_\phi(\mu_x + \sigma_x \epsilon)||_2$$

$$\mu_x, \sigma_x = e_\theta(x), \epsilon \sim \mathrm{N}(0, I)$$

$$similarity\ loss = KL\ Divergence = D_{KL}(\mathrm{N}(\mu_x, \sigma_x)||\mathrm{N}(0, I))$$

$$loss = reconstruction\ loss + similarity\ loss$$

损失函数

- **重构项 (Reconstruction)**

要求：生成的图像要像原图

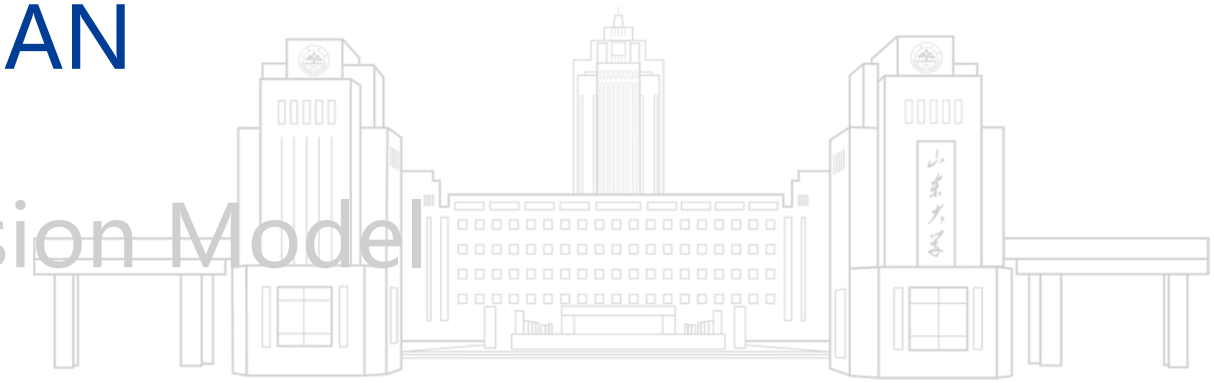
数学表达：

$$reconstruction\ loss = ||x - \hat{x}||_2 = ||x - d_\phi(z)||_2 = ||x - d_\phi(\mu_x + \sigma_x \epsilon)||_2$$

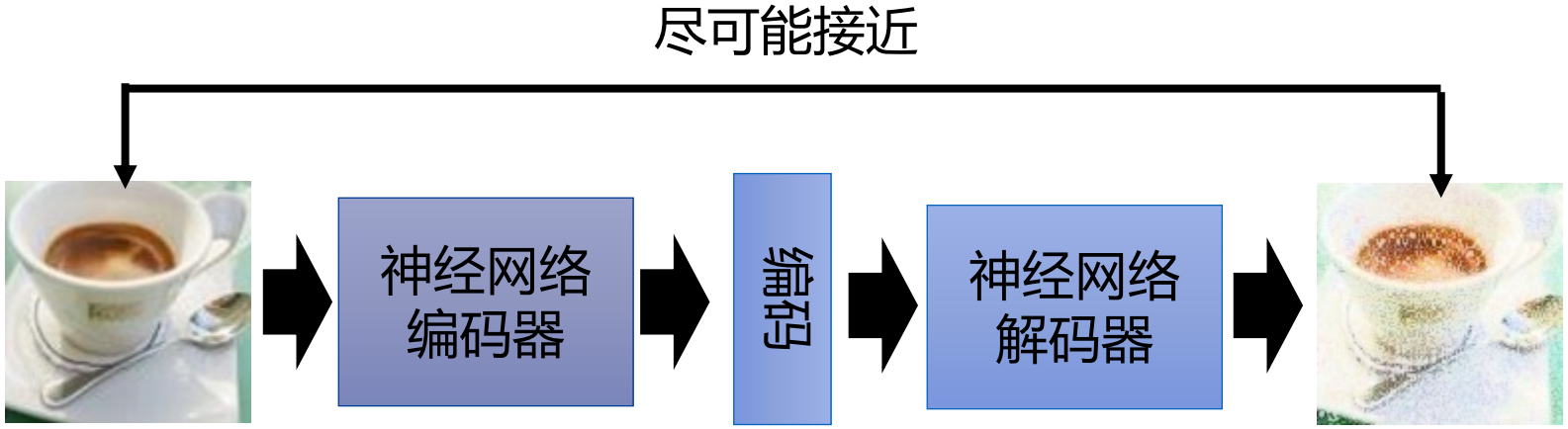
$$\mu_x, \sigma_x = e_\theta(x), \epsilon \sim \mathrm{N}(0, I)$$

倾向： 减小方差 σ^2，退化回AE

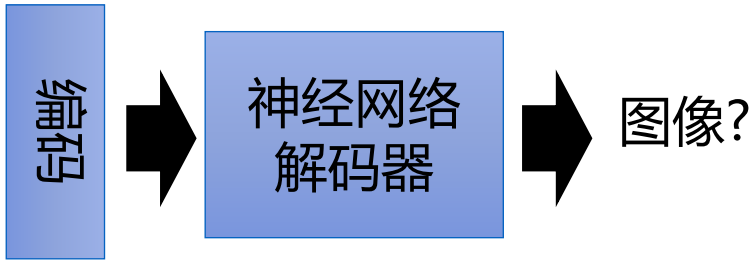
章节目录 CONTENTS



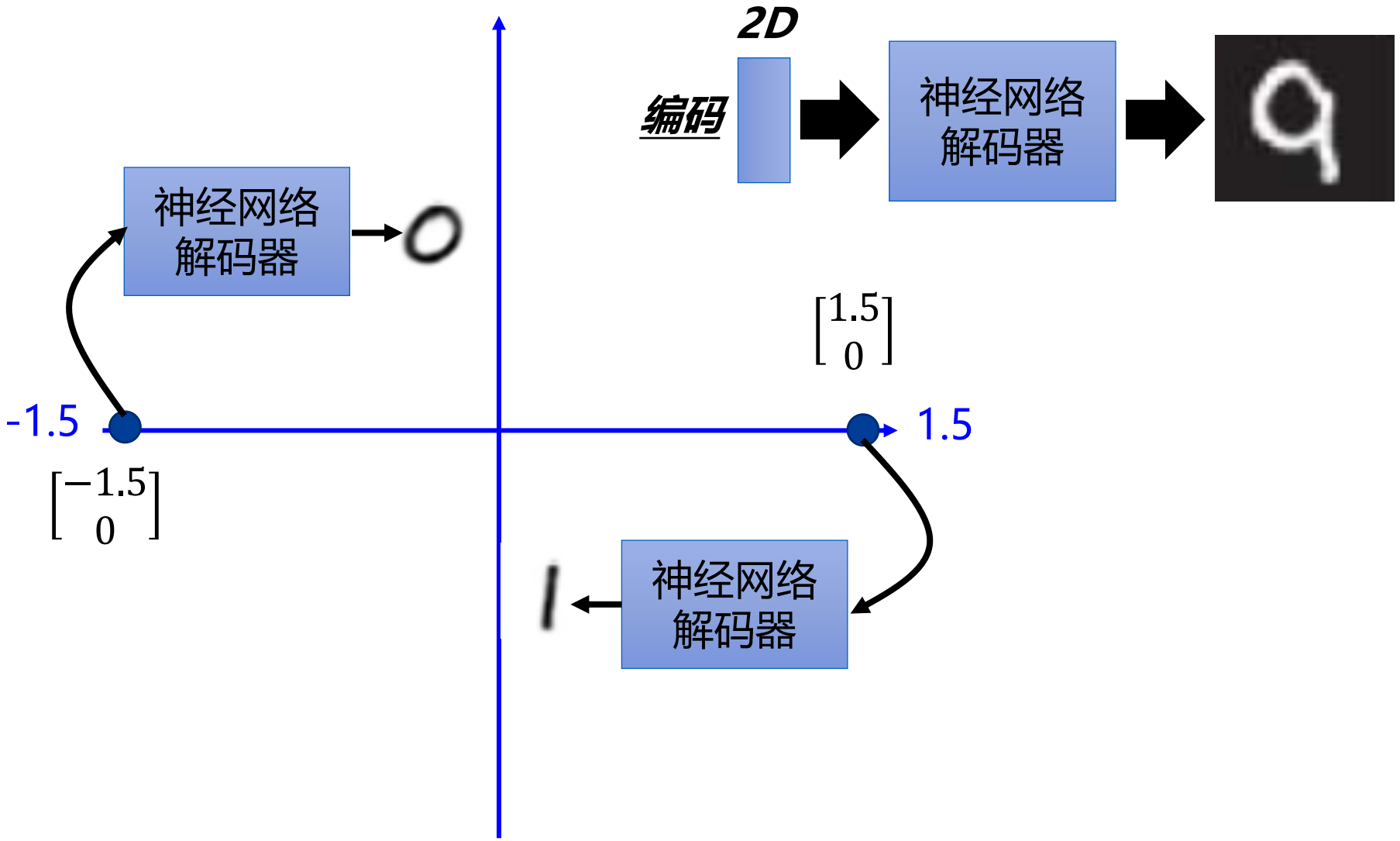
知识回顾：自编码器（Auto Encoder）



随机生成一个向量作为编码

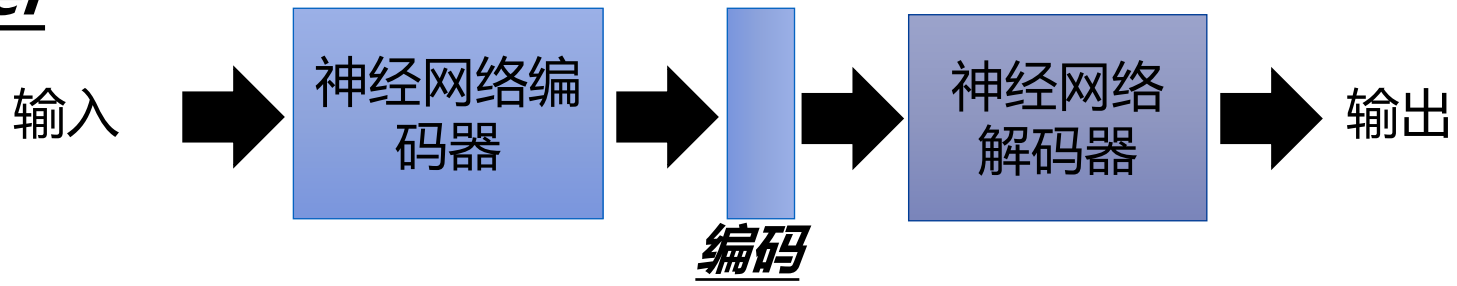


知识回顾：自编码器（Auto Encoder）

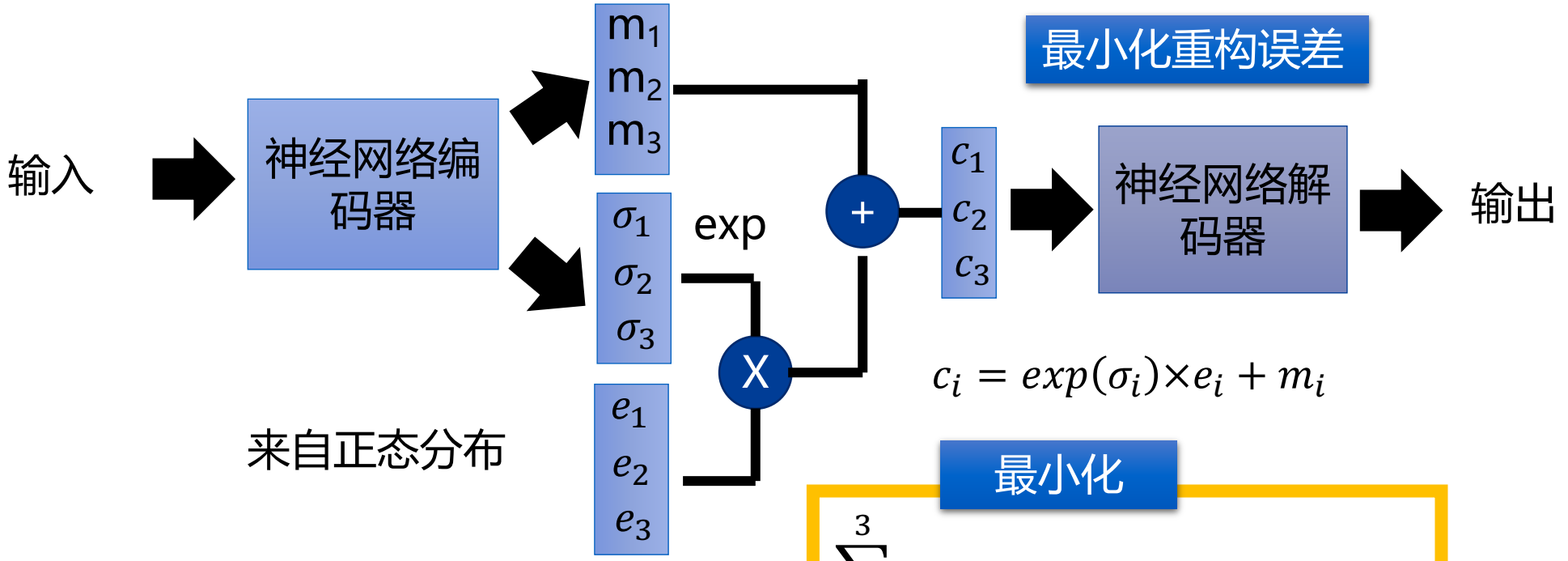


知识回顾：变分自编码器（VAE）

Auto-encoder

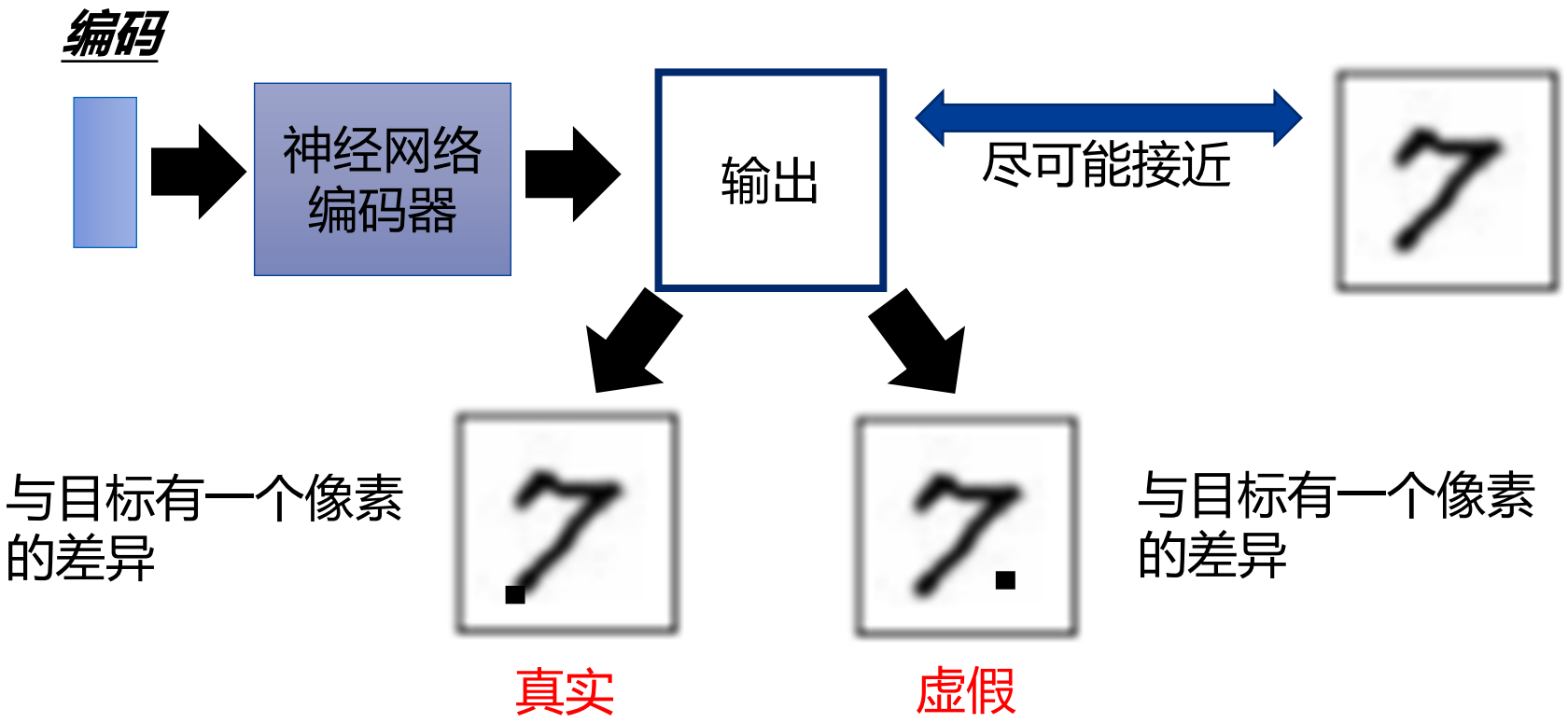


VAE



知识回顾：变分自编码器（VAE）的缺陷

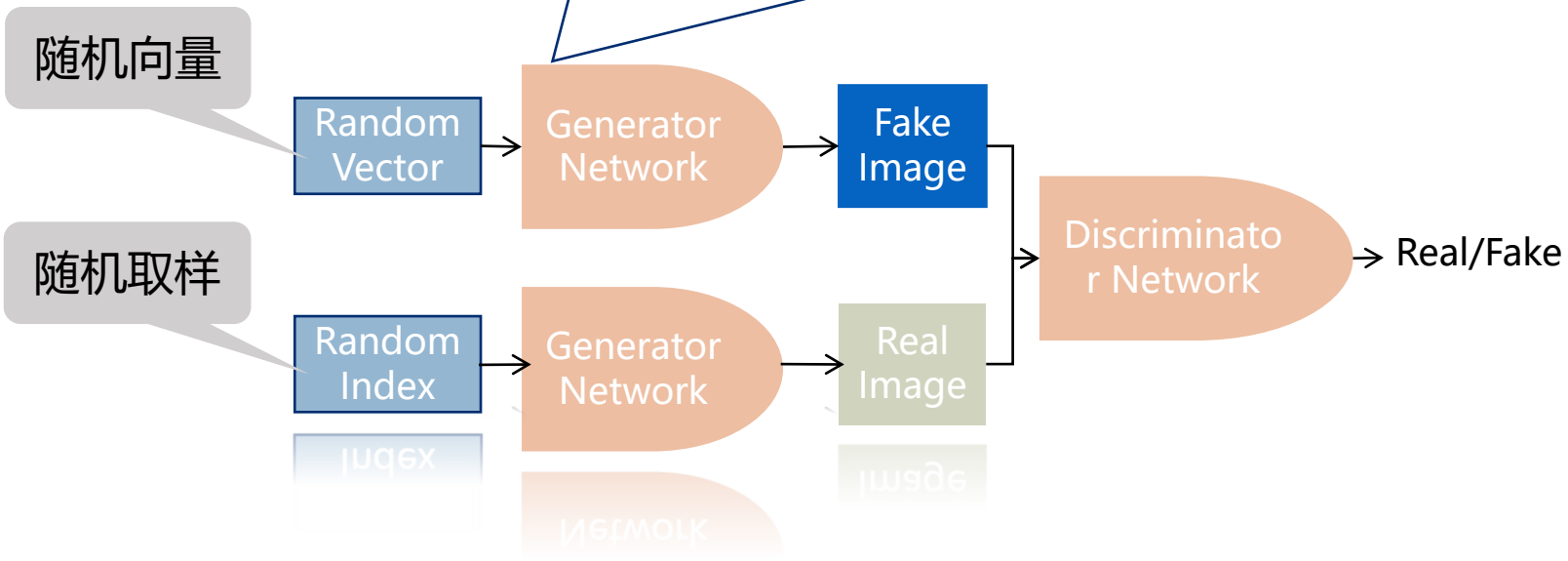
- VAE并没有真正尝试去模拟真实的图像



生成对抗网络

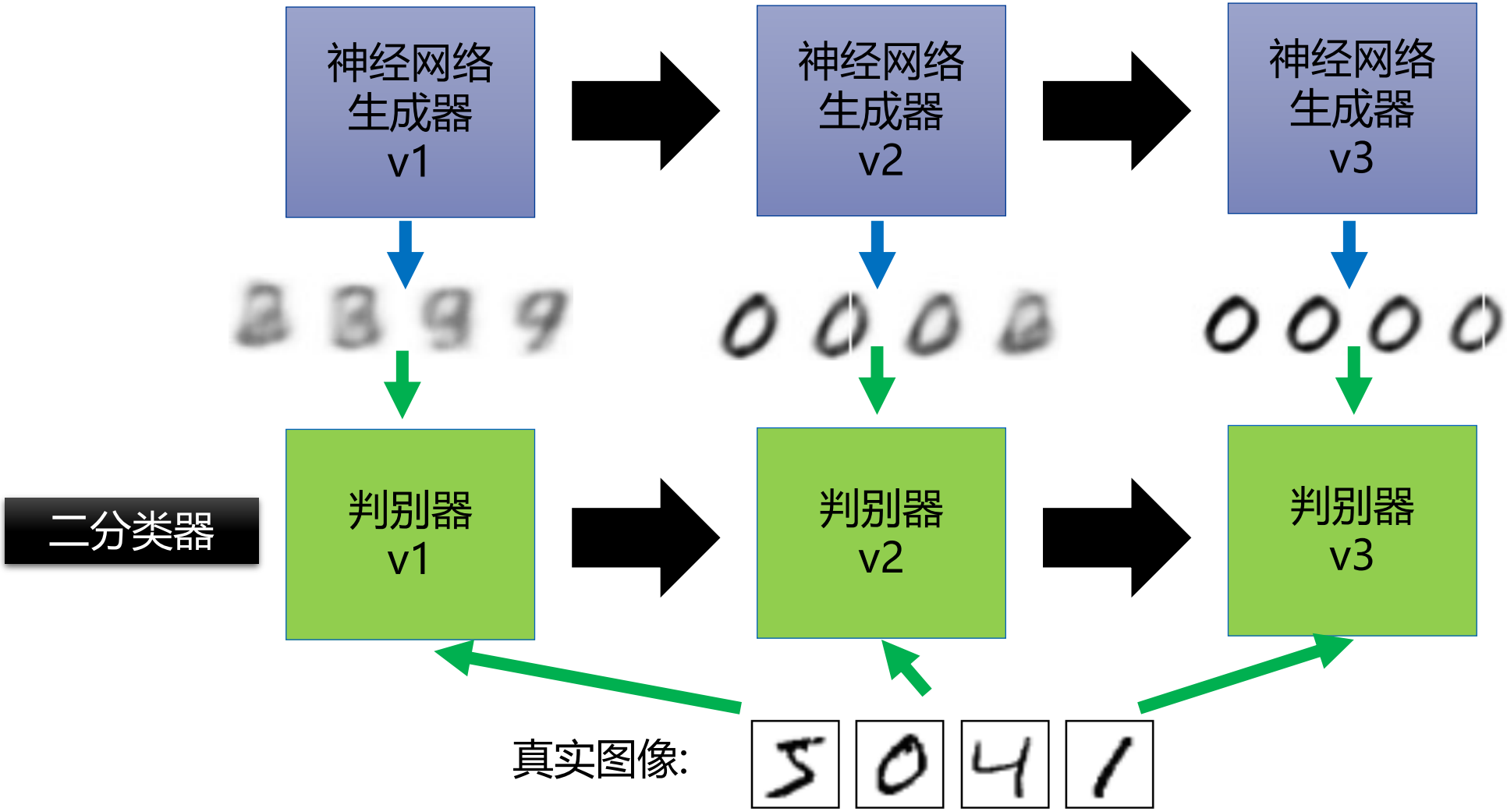
- 对抗网络：使用两个网络互相竞争，称之为对抗式结构

生成器G：通过一个参数化概率生成模型(通常用深度神经网络进行参数化)进行概率分布的逆变换采样，得到一个生成的概率分布。

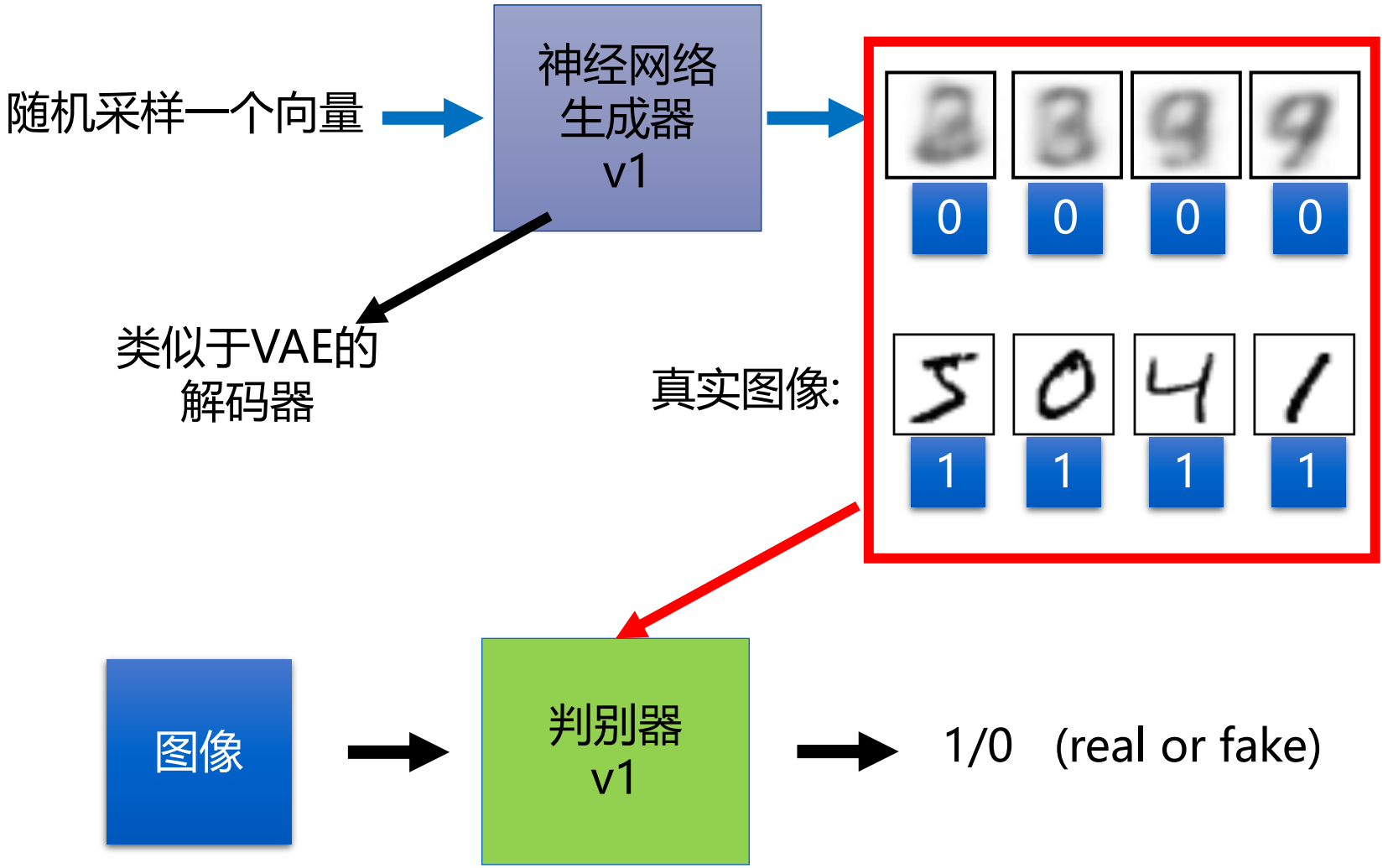


判别器D：给定样本，判断(通常也是深度卷积神经网络)这个样本来自真实数据还是伪造数据。

生成对抗网络（GAN）的迭代训练过程



生成对抗网络 —— 判别器


随机采样一个向量
神经网络
生成器
v1
类似于VAE的
解码器
真实图像:
0
0
0
0
1
1
1
1
图像
判别器
v1
1/0 (real or fake)

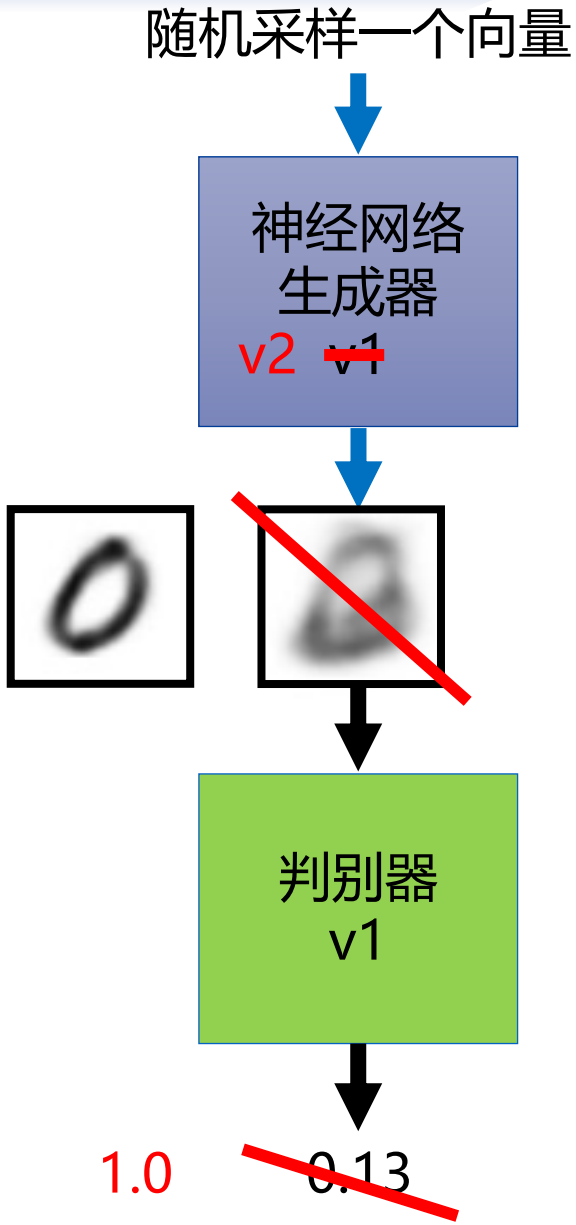
生成对抗网络 —— 生成器

更新生成器的参数

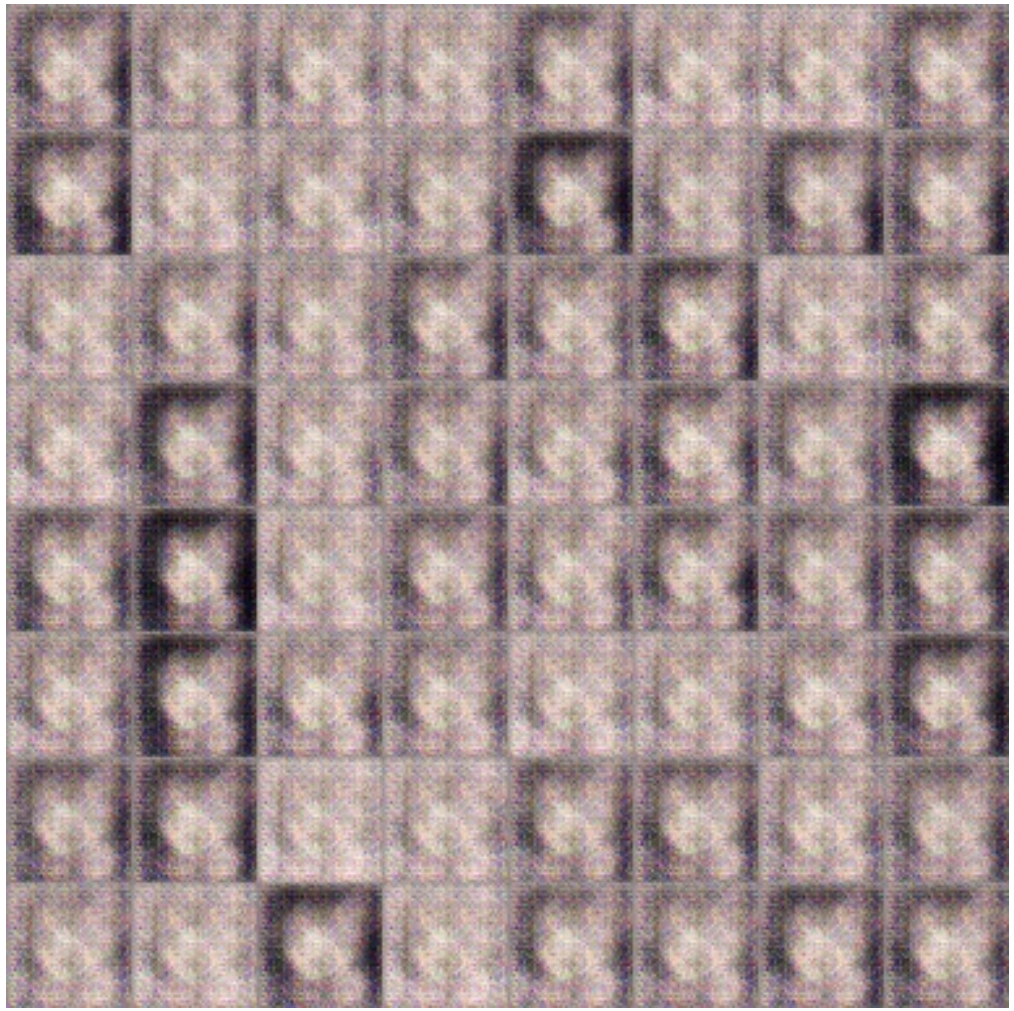
➡ 目的是让输出被归类为“真”

生成器 + 判别器 = 一个网络

使用梯度下降来更新生成器中的参数，
但是固定住判别器



GAN示例 —— 二次元人物头像生成



100 轮次

1000 轮次

GAN示例 —— 二次元人物头像生成

2000 轮次

5000 轮次

GAN示例 —— 二次元人物头像生成

10000 轮次

20000 轮次

GAN示例 —— 二次元人物头像生成

50000 轮次

生成对抗网络

- 损失函数

$$L(G,D) = -E_{x\sim P_r} \log D(x) - E_{z\sim P_z} \log\left(1 - D(G(z))\right)$$

这个Loss其实就是交叉熵。对于判别器D，它的任务是最小化$-L(G,D)$，即

$$L(G,D) = -E_{x\sim P_r} \log D(x) - E_{x\sim P_g} \log\left(1 - D(G(z))\right)$$

如果采用零和博弈，生成器G的目标是最小化$L(G,D)$，而实际操作发现零和博弈训练效果并不好，G的目标一般采用最小化

$$-E_{z\sim P_z} \log D(G(z)) \qquad -E_{z\sim P_z} \log\left(1 - D(G(z))\right)$$

一般来说，更新D是，G是固定的；更新G时，D是固定的。

极大似然估计（Maximum Likelihood Estimation）

- 给定一个数据分布：$P_{data}(x)$

- 我们可以得到一个由参数 θ 控制的分布：$P_G(x;\theta)$
  - 例如：$P_G(x;\theta)$ 是一个高斯混合模型，θ 指的是高斯分布的均值和方差
  - 我们想找到一组参数 θ，使得生成分布 $P_G(x;\theta)$ 尽可能接近真实分布 $P_{data}(x)$

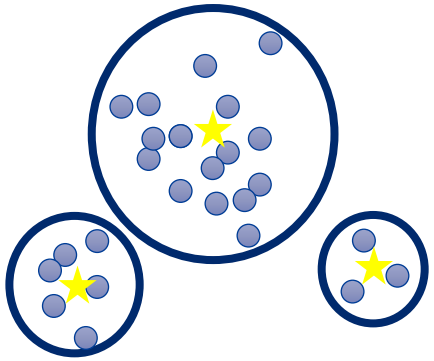
从 $P_{data}(x)$ 中采样得到 m 个样本 $\{x^1, x^2, \dots, x^m\}$

我们可以计算样本在生成分布中出现的概率$P_G(x^i;\theta)$

生成这些样本的Likelihood：

$$L = \prod_{i=1}^{m} P_G(x^i;\theta)$$

找到一个最优参数 θ^*，使得L最大



极大似然估计（Maximum Likelihood Estimation）

$$\theta^* = arg\max_{\theta}\prod_{i=1}^{m} P_G(x^i;\theta) = arg\max_{\theta} log\prod_{i=1}^{m} P_G(x^i;\theta)$$

$$= arg\max_{\theta}\sum_{i=1}^{m} logP_G(x^i;\theta)$$

来自 $P_{data}(x)$ 的采样 $\{x^1, x^2, \dots, x^m\}$

$$\approx arg\max_{\theta} E_{x\sim P_{data}}[logP_G(x;\theta)]$$

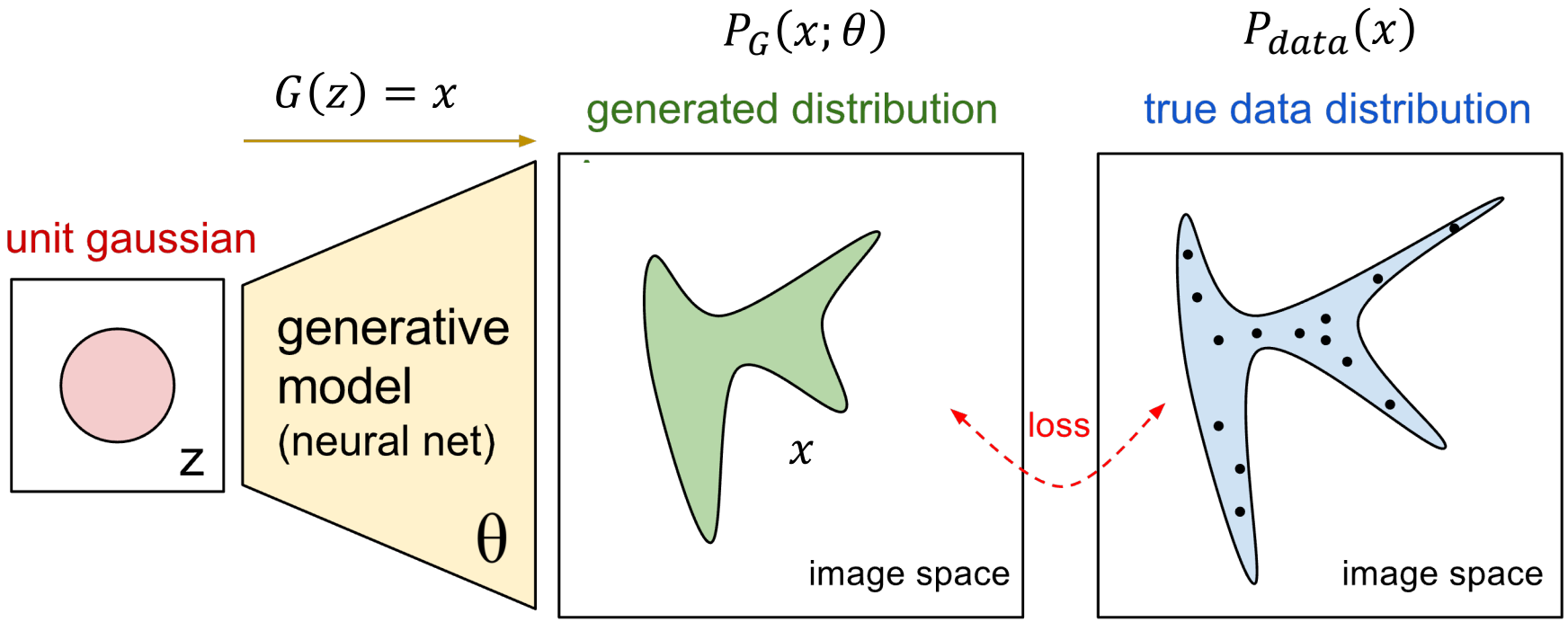
$$= arg\max_{\theta}\int_x P_{data}(x)logP_G(x;\theta)dx - \int_x P_{data}(x)logP_{data}(x)dx$$

$$= arg\min_{\theta} KL\big(P_{data}(x)||P_G(x;\theta)\big)$$

如何得到一个非常通用的 $P_G(x;\theta)$?

极大似然估计（Maximum Likelihood Estimation）

现在 $P_G(x;\theta)$ 是一个神经网络：



$$P_G(x) = \int_z P_{prior}(z) I_{[G(z)=x]} dz$$

计算似然函数是非常困难的。

GAN的基本概念

- 生成器 G　　难以通过最大似然估计进行学习
  - G 是一个函数, 输入为 z, 输出为 x
  - 给定一个先验分布 $P_{prior}(z)$，通过函数 G 定义了一个概率分布 $P_G(x)$
- 判别器 D
  - D 是一个函数, 输入为 x, 输出为一个标量
  - 评估 $P_G(x)$ 与 $P_{data}(x)$ 之间的“差异”
- 存在一个函数 V(G,D).

$$G^* = arg \min_G \max_D V(G, D)$$

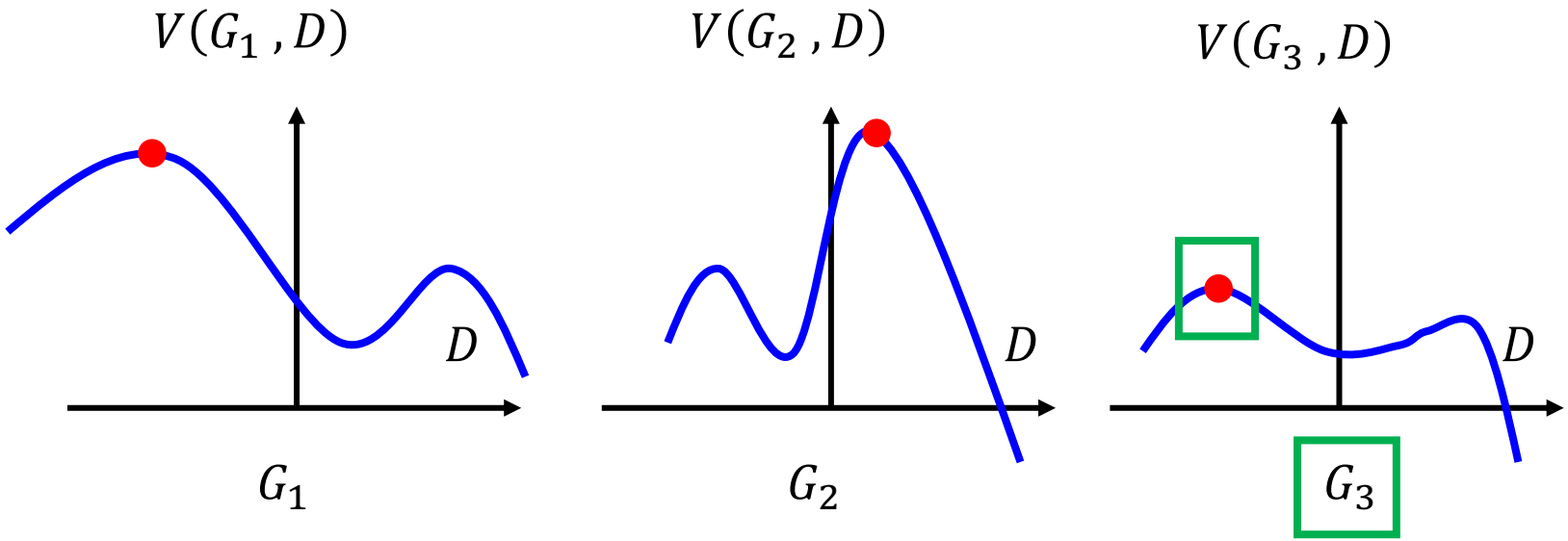
GAN的基本概念

$$G^* = arg \min_G \max_D V(G,D)$$

$$V = E_{x\sim P_{data}}[logD(x)] + E_{x\sim P_G}[log(1-D(x))]$$

给定一个生成器 G，$\max_D V(G,D)$ 评估 P_G 和 P_{data} 之间的差异

选择能使定义的 P_G 与 P_{data} 最为相似的生成器 G



GAN的算法总结

初始化辨别器 D 的参数 θ_d 以及生成器 G 的参数 θ_g

只能找到下界 $\max_D V(G,D)$

- 在每一次训练迭代中:

学习判别器 D

重复 k 次

- 从数据分布 $P_{data}(x)$ 中采样 m 个样本 $\{x^1, x^2, \dots, x^m\}$
- 从先验分布 $P_{prior}(z)$ 中采样 m 个噪声样本$\{z^1, z^2, \dots, z^m\}$
- 获得生成数据$\{\tilde{x}^1, \tilde{x}^2, \dots, \tilde{x}^m\}$, $\tilde{x}^i = G(z^i)$
- 更新判别器参数 θ_d 以最大化以下目标函数:
  - $\tilde{V} = \frac{1}{m}\sum_{i=1}^{m} logD(x^i) + \frac{1}{m}\sum_{i=1}^{m} log\left(1 - D(\tilde{x}^i)\right)$
  - $\theta_d \leftarrow \theta_d + \eta \nabla \tilde{V}(\theta_d)$

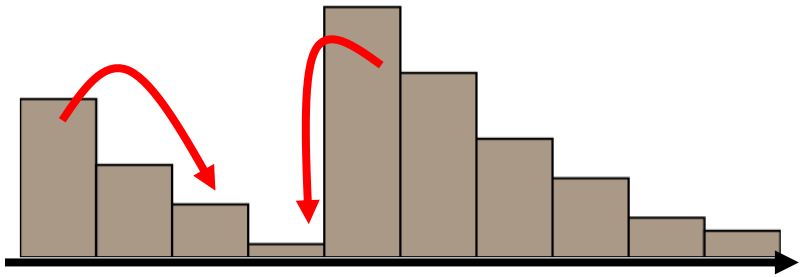
学习生成器 G

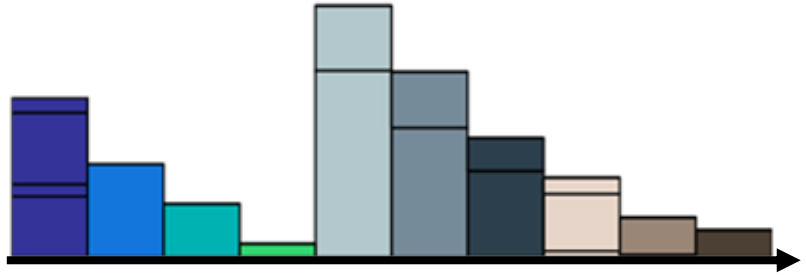
仅执行一次

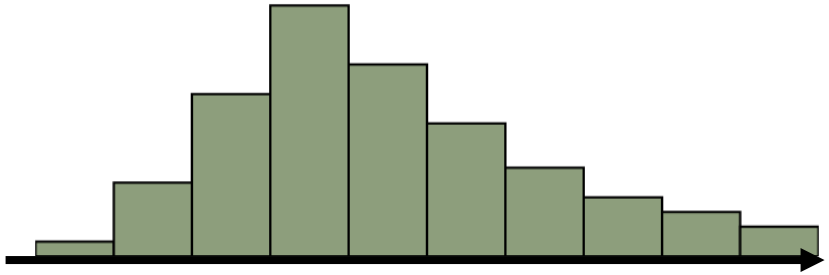
- 从先验分布 $P_{prior}(z)$ 中采样 m 个噪声样本$\{z^1, z^2, \dots, z^m\}$
- 更新生成器参数 θ_g 以最小化以下目标函数:
  - $\tilde{V} = \cancel{\frac{1}{m}\sum_{i=1}^{m} logD(x^i)} + \frac{1}{m}\sum_{i=1}^{m} log\left(1 - D\left(G(z^i)\right)\right)$
  - $\theta_g \leftarrow \theta_g - \eta \nabla \tilde{V}(\theta_g)$

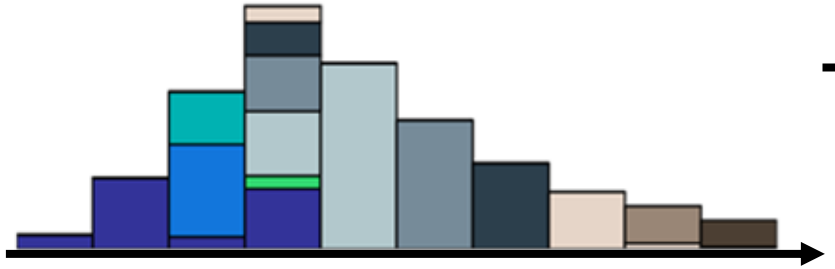
推土机距离

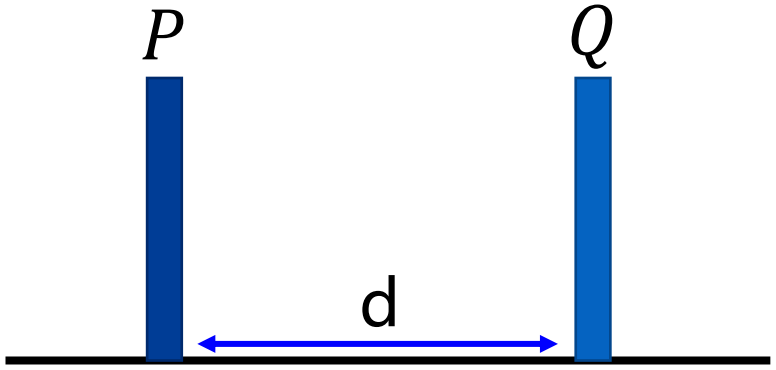
- 将一个分布 P 视为一堆土, 将另一个分布 Q 视为目标
- 搬运工将这些图移动到实现目标形状所需要移动的平均距离











$$W(P,Q) = d$$

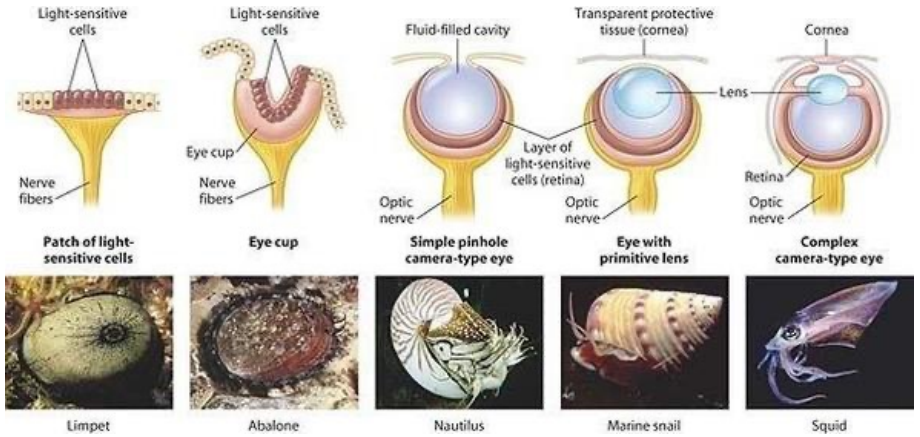
为什么用推土机距离?

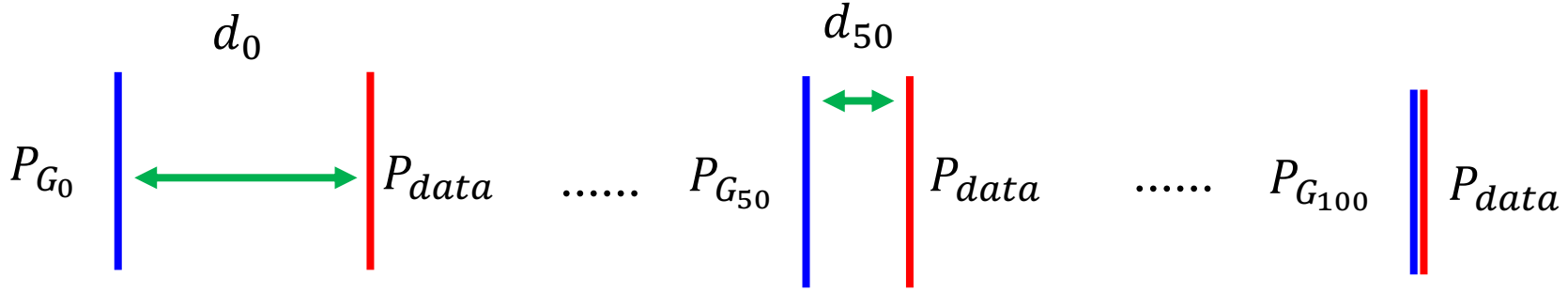
推土机距离 = Wasserstein距离

$$D_f(P_{data}||P_G)$$

$$\Downarrow$$

$$W(P_{data}, P_G)$$





$$JS(P_{G_0}, P_{data}) = log2$$

$$JS(P_{G_{50}}, P_{data}) = log2$$

$$JS(P_{G_{100}}, P_{data}) = 0$$

$$W(P_{G_0}, P_{data}) = d_0$$

$$W(P_{G_{50}}, P_{data}) = d_{50}$$

$$W(P_{G_{100}}, P_{data}) = 0$$

WGAN的算法

初始化辨别器 D 的参数 θ_d 以及生成器 G 的参数 θ_g

D 的输出不使用sigmoid

- 在每一次训练迭代中:

学习判别器 D

重复 k 次

- 从数据分布 $P_{data}(x)$ 中采样 m 个样本 $\{x^1, x^2, \dots, x^m\}$
- 从先验分布 $P_{prior}(z)$ 中采样 m 个噪声样本$\{z^1, z^2, \dots, z^m\}$
- 获得生成数据$\{\tilde{x}^1, \tilde{x}^2, \dots, \tilde{x}^m\}$, $\tilde{x}^i = G(z^i)$
- 更新判别器参数 θ_d 以最大化以下目标函数:
  - $\tilde{V} = \frac{1}{m}\sum_{i=1}^{m} D(x^i) - \frac{1}{m}\sum_{i=1}^{m} D(\tilde{x}^i)$
  - $\theta_d \leftarrow \theta_d + \eta \nabla \tilde{V}(\theta_d)$

权重裁剪

学习生成器 G

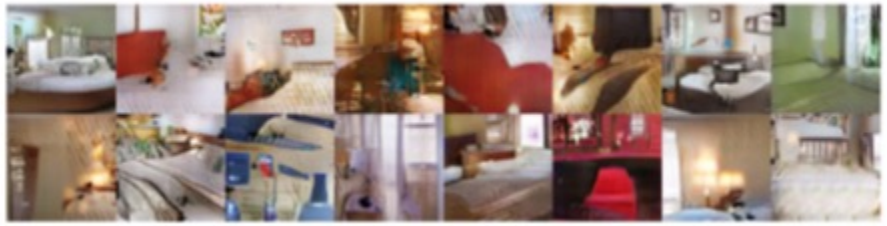
仅执行一次

- 从先验分布 $P_{prior}(z)$ 中采样 m 个噪声样本$\{z^1, z^2, \dots, z^m\}$
- 更新生成器参数 θ_g 以最小化以下目标函数:
  - $\tilde{V} = \frac{1}{m}\sum_{i=1}^{m} \log D(x^i) - \frac{1}{m}\sum_{i=1}^{m} D\left(G(z^i)\right)$
  - $\theta_g \leftarrow \theta_g - \eta \nabla \tilde{V}(\theta_g)$

WGAN的算法

CNN 生成器:

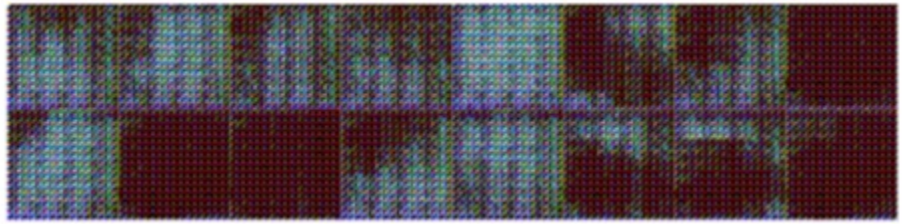
W-GAN



GAN

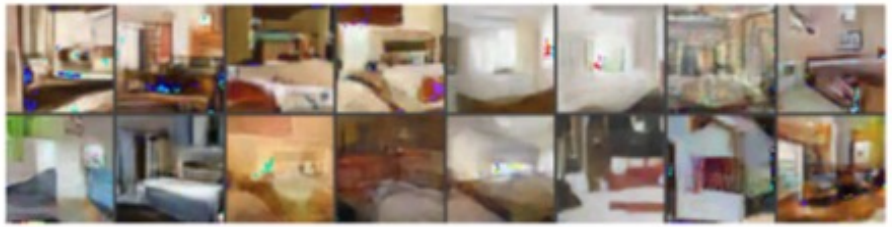
CNN 生成器 (未使用batch归一化，结构不佳):

W-GAN



GAN

MLP 生成器:

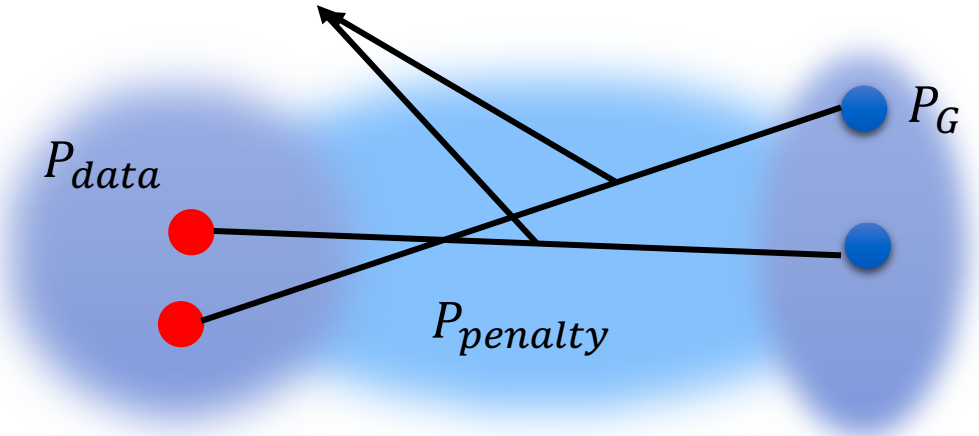


W-GAN

GAN

改进型WGAN

$$W(P_{data}, P_G) \approx \max_D \{E_{x\sim P_{data}}[D(x)] - E_{x\sim P_G}[D(x)] - \lambda E_{x\sim P_{penalty}}[max(0, \|\nabla_x D(x)\| - 1)]\}$$



“鉴于在所有地方都强制执行 Lipschitz 约束是难以实现的，仅在**这些直线（连接真实样本和生成样本的直线）上**执行约束似乎就足够了，并且实验结果表明其性能良好。”

仅对P_{data} 和 P_G 之间的区域施加梯度约束，因为这些区域直接影响了P_G 如何向 P_{data} 移动。

改进型WGAN、WGAN等模型效果的比较

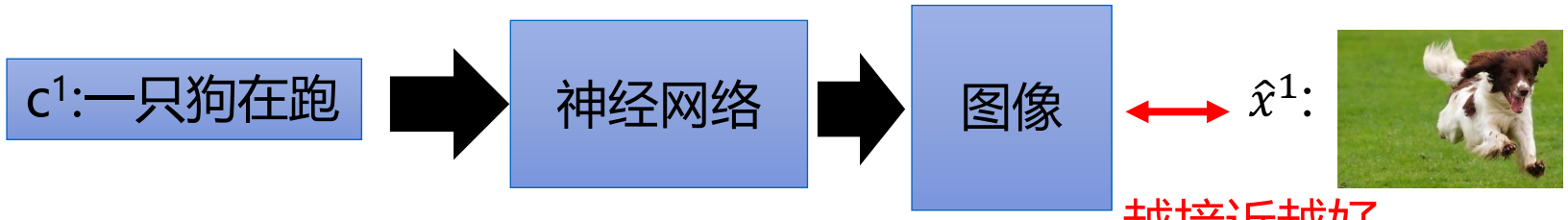
DCGAN	LSGAN	原始WGAN	改进型WGAN
G: CNN, D: CNN			
			
G: CNN (no normalization), D: CNN (no normalization)			
			
G: CNN (tanh), D: CNN(tanh)			
			

条件生成对抗网络（cGAN）

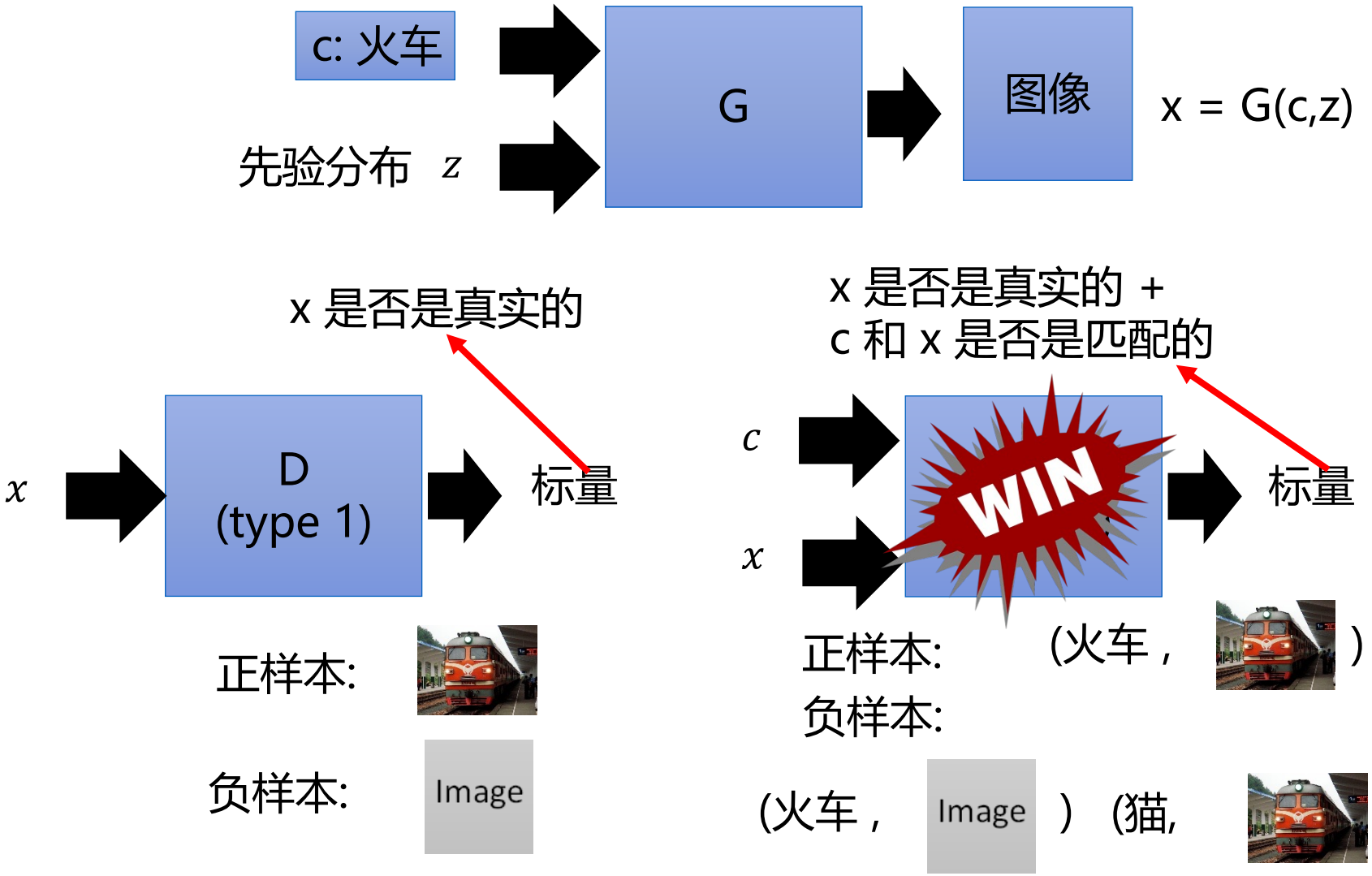
c^1: 一只狗在跑 $\hat{x}^1$: 

c^2: 一只鸟在飞 $\hat{x}^2$: 

- 传统监督学习下的文本转图像



条件生成对抗网络（cGAN）



Cycle GAN

https://arxiv.org/abs/1703.10593
https://junyanz.github.io/CycleGAN/

领域 X

领域 Y

领域 X

忽略输入

$G_{X \to Y}$

变成和领域 Y相似的图像

不是我们想要的

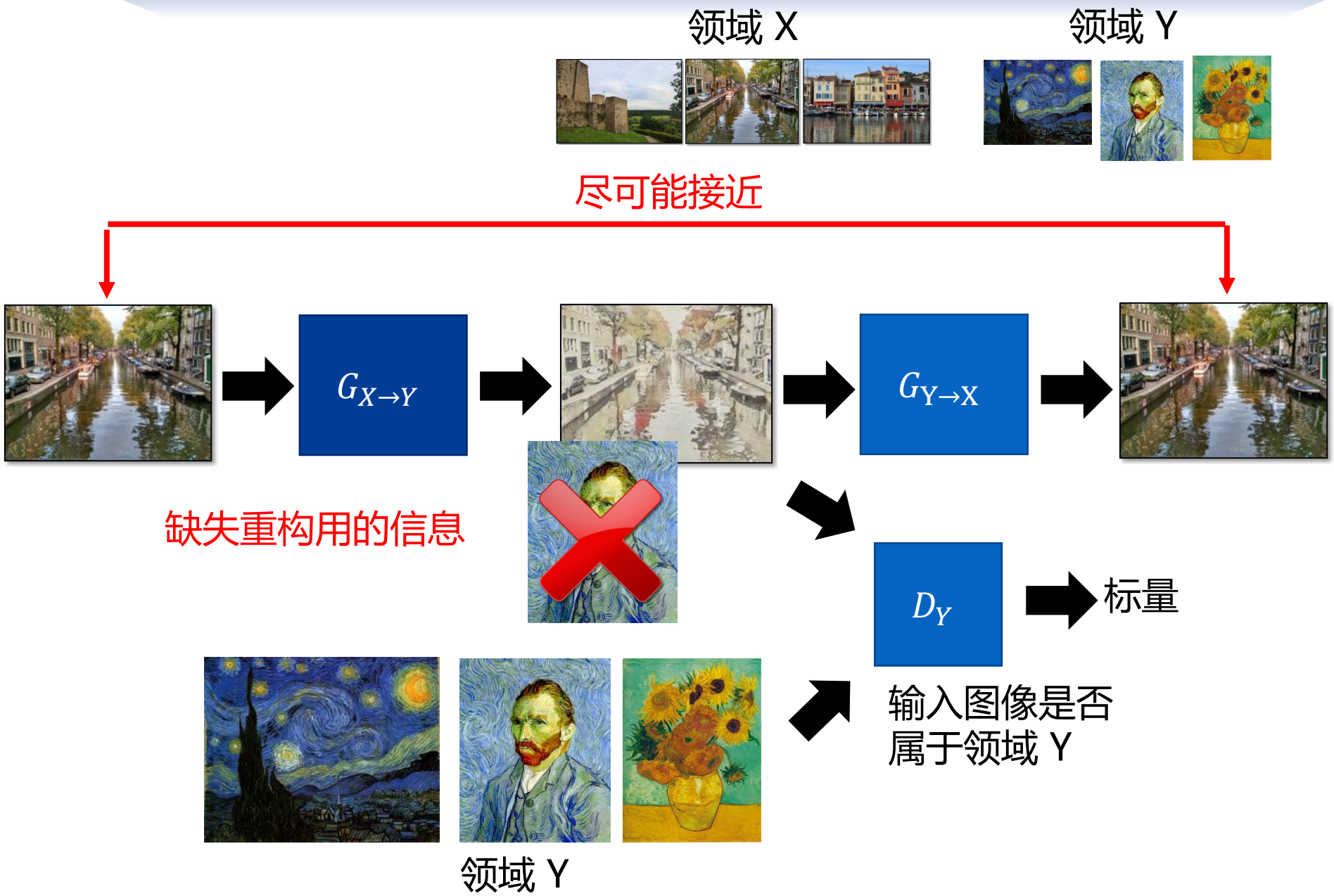
D_Y

标量

输入图像是否属于领域 Y

领域 Y

Cycle GAN



章节目录 CONTENTS

理解扩散模型的两种视角

- **变分扩散模型**
  - Variational Diffusion Model（VDM）
  - 涉及：马尔科夫链、极大似然估计、变分推断等
- **基于分数的生成模型**
  - Score-based Generative Model（SGM）
  - 涉及：朗之万动力学、分数匹配、随机微分方程、常微分方程等

理解扩散模型的两种视角

- **变分扩散模型**
  - Variational Diffusion Model（VDM）
  - 涉及：马尔科夫链、极大似然估计、变分推断等
- **基于分数的生成模型**
  - Score-based Generative Model（SGM）
  - 涉及：朗之万动力学、分数匹配、随机微分方程、常微分方程等

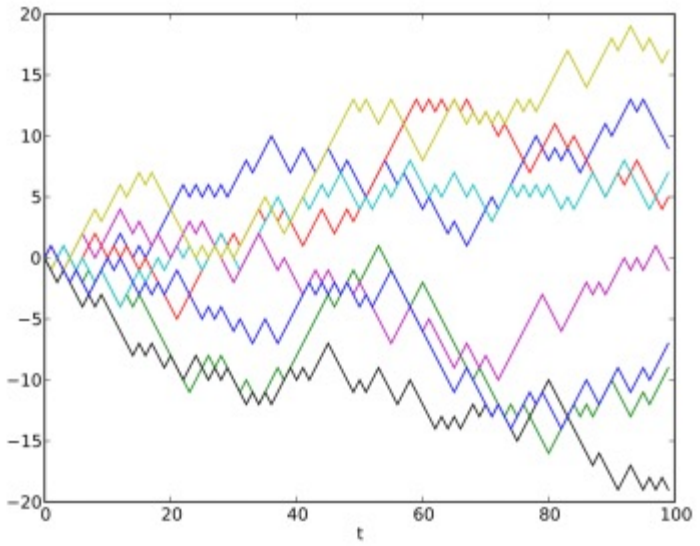
随机过程

时间t可以是离散的
也可以是连续的！

- 随机过程（Stochastic Process）$\{X_t, t \in T\}$是一组依赖时间t变化的随机变量/向量。
- 状态空间（State Space）是X_t所有可能取值构成的集合。
- X_t表示随机变量/向量，x_t表示具体的取值
- 边际分布、条件分布、联合分布

随机过程

- **例1（随机游动）：** 一个醉汉在路上行走，以概率p前进一步，以概率$1-p$后退一步（假定其步长相同）。以X_t记他在路上的位置，则$\{X_t\}$就是直线上的随机游动。



- **例2（布朗运动）**：飘浮在液面上的微小粒子会不断进行无规则的运动。若记(X_t, Y_t)为粒子在平面坐标上的位置，则它是平面上的布朗运动。

一个计算机模拟的布朗运动

马尔科夫性质

- 假设$\{X_t\}_{t>0}$是一个离散时间的随机过程，X_t表示系统在t时刻的状态。记t为当前时刻，若对任意$t>0$和$x_0,\ldots,x_{t+1}$，都有

$$\Pr(X_{t+1}=x_{t+1}\mid X_t=x_t,X_{t-1}=x_{t-1},\ldots,X_0=x_0)$$
$$=\Pr(X_{t+1}=x_{t+1}\mid X_t=x_t),$$

则称$\{X_t\}_{t>0}$具有马尔科夫性质。

马尔科夫性质

未来　　当前　　过去

$$\Pr(X_{t+1}=x_{t+1}\mid X_t=x_t, X_{t-1}=x_{t-1},\ldots,X_0=x_0)$$
$$=\Pr(X_{t+1}=x_{t+1}\mid X_t=x_t)$$

未来的状态只依赖于当前的状态，

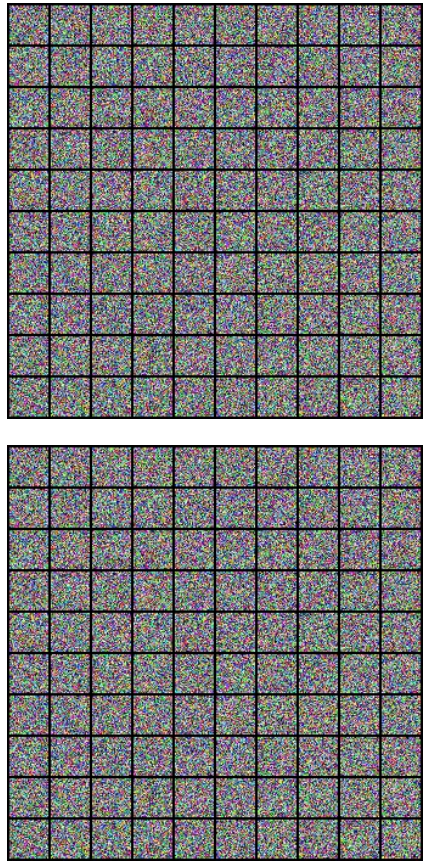
而与过去的状态无关

马尔科夫性质

- 马尔科夫链（Markov Chain）是一类具有马尔科夫性质的随机过程。
- 即未来的状态只依赖于当前的状态，而与过去的状态无关。

经典模型：DDPM

- DDPM简介
- 正向与逆向扩散过程
- 损失函数（无推导）
- 网络结构
- 训练与采样算法
- 实验效果：Fashion MNIST

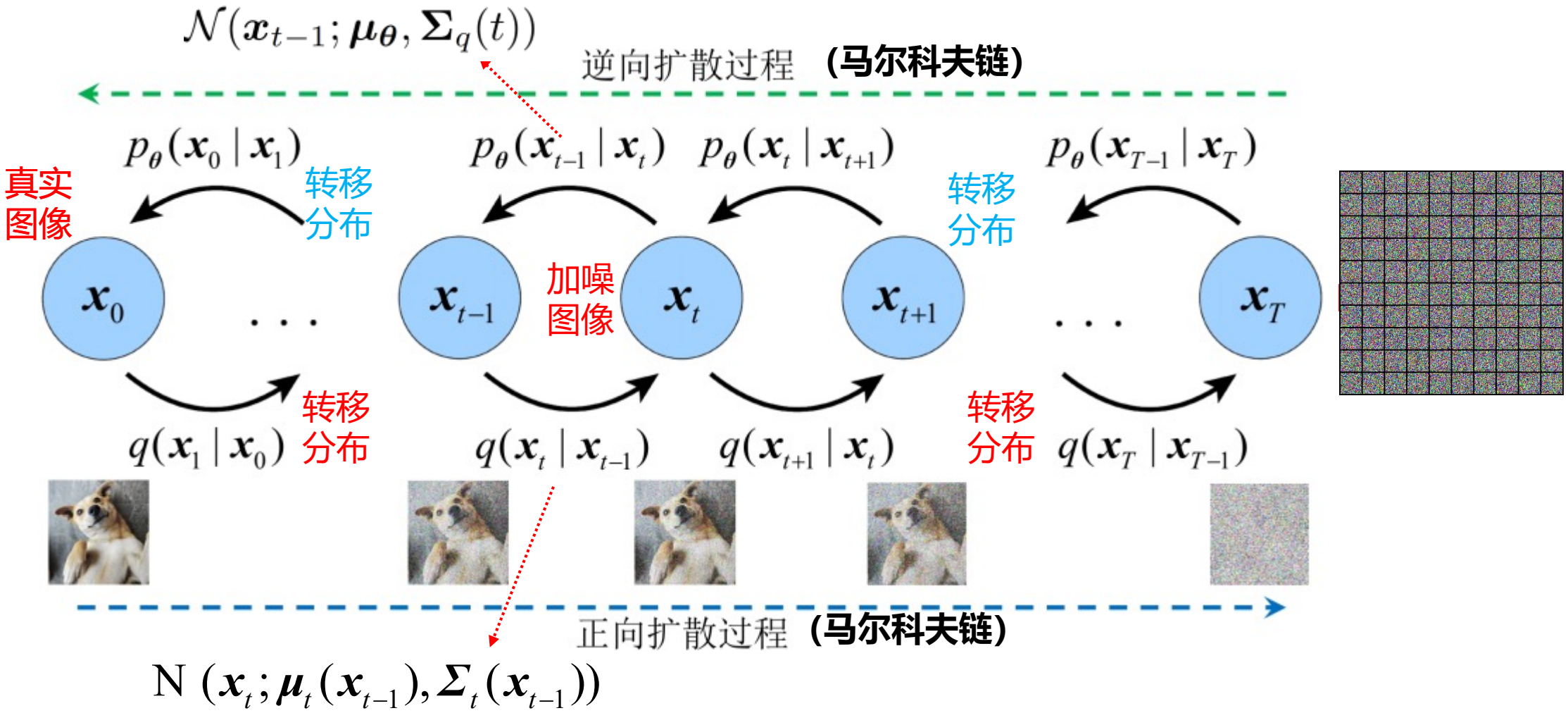


DDPM简介

- 去噪扩散概率模型
- Denoising Diffusion Probabilistic Models
- 由Jonathan Ho等人于2020年提出
- 属于变分扩散模型的一种！

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." ***NeurIPS 2020***

正向/逆向扩散过程


$\mathcal{N}(x_{t-1}; \mu_\theta, \Sigma_q(t))$
逆向扩散过程 (马尔科夫链)
$p_\theta(x_0 \mid x_1)$
$p_\theta(x_{t-1} \mid x_t)$
$p_\theta(x_t \mid x_{t+1})$
$p_\theta(x_{T-1} \mid x_T)$
真实图像
转移分布
转移分布
转移分布
转移分布
加噪图像
x_0
x_{t-1}
x_t
x_{t+1}
x_T
$q(x_1 \mid x_0)$
$q(x_t \mid x_{t-1})$
$q(x_{t+1} \mid x_t)$
$q(x_T \mid x_{T-1})$
正向扩散过程 (马尔科夫链)
$\mathrm{N}(x_t; \mu_t(x_{t-1}), \Sigma_t(x_{t-1}))$

正向扩散过程

- Forward Diffusion
- 从$\boldsymbol{x}_0 \sim p_{data}(\boldsymbol{x}_0)$到$\boldsymbol{x}_T \sim N(0, I)$的加噪过程
- 无可学习参数
- 状态转移分布为

$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = N(\boldsymbol{x}_t; \boldsymbol{\mu}_t(\boldsymbol{x}_{t-1}), \boldsymbol{\Sigma}_t(\boldsymbol{x}_{t-1}))$$

- 满足马尔科夫性质

逆向扩散过程

- Reverse Diffusion
- 从$\boldsymbol{x}_T \sim N(0, I)$到$\boldsymbol{x}_0 \sim p_{data}(\boldsymbol{x}_0)$的去噪过程
- 含有可学习参数$\boldsymbol{\theta}$
- 状态转移分布为

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = N(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t), \boldsymbol{\Sigma}_{\boldsymbol{\theta}}(\boldsymbol{x}_t, t))$$

- 满足马尔科夫性质

损失函数

- **本质**：需要极大化对数似然的证据下界（ELBO）

$$\log p_\theta(\boldsymbol{x}) = \log \int p_\theta(x_{0:T}) dx_{1:T} \geq \mathbb{E}_{q(x_{1:T}|x_0)}[\frac{p(x_{0:T})}{q(x_{1:T}|x_0)}]$$

- 另外，在正向扩散过程中，可推导出

$$\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\varepsilon}_0$$

- **可证明**：最大化ELBO近似等价于**噪声预测**损失：

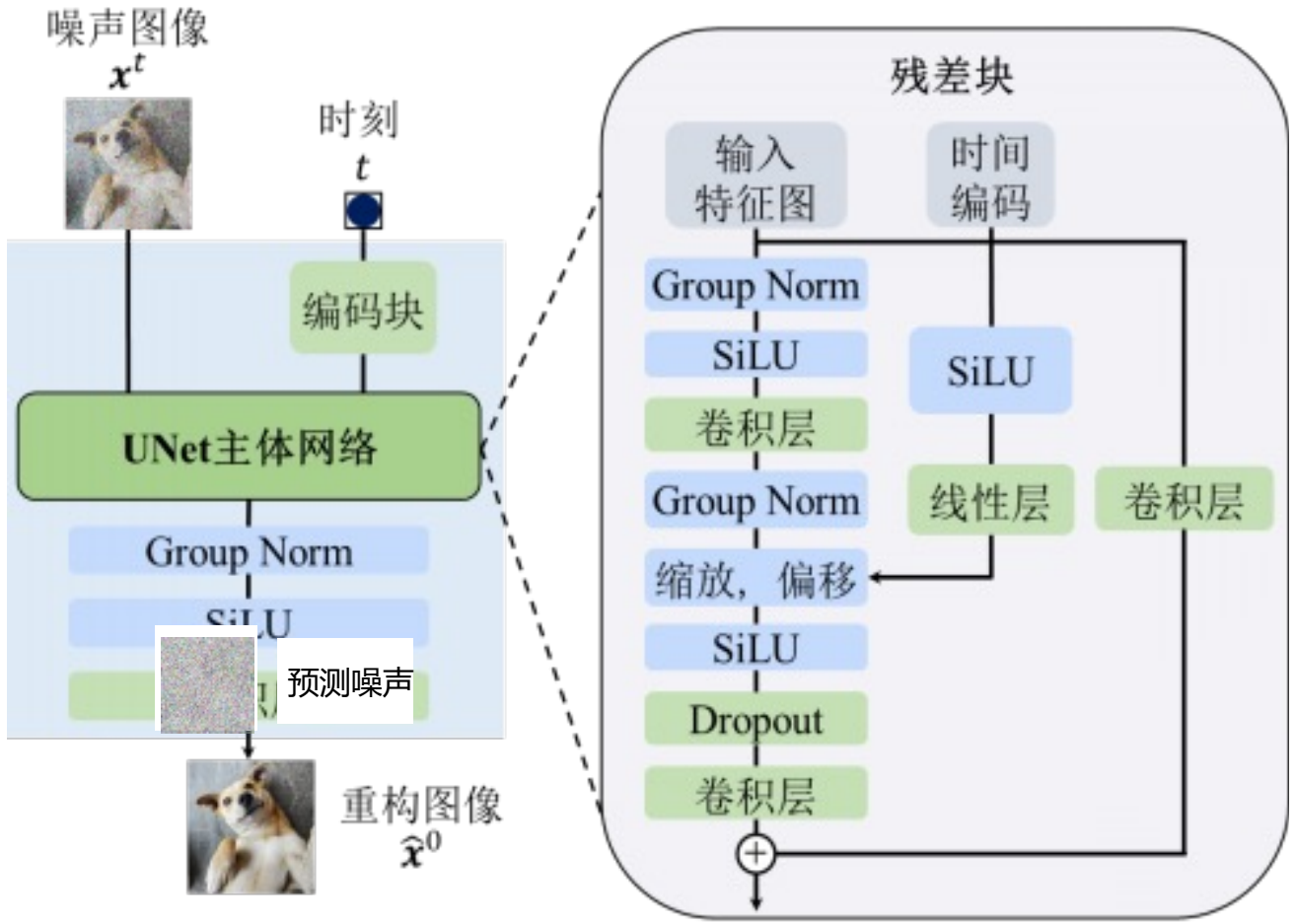
$$L^{\text{DDPM}}(\boldsymbol{\theta}) = \mathrm{E}_{\boldsymbol{\varepsilon}_0 \sim \mathrm{N}(\boldsymbol{0},\boldsymbol{I}),\, t\sim DU(0,T),\, \boldsymbol{X}_0 \sim p(\boldsymbol{x}_0)} \left[\left\| \boldsymbol{\varepsilon}_0 - \hat{\boldsymbol{\varepsilon}}_{\boldsymbol{\theta}}(\sqrt{\bar{\alpha}_t}\boldsymbol{X}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\varepsilon}_0, t) \right\|_2^2 \right]$$

源噪声　　神经网络　　X_t

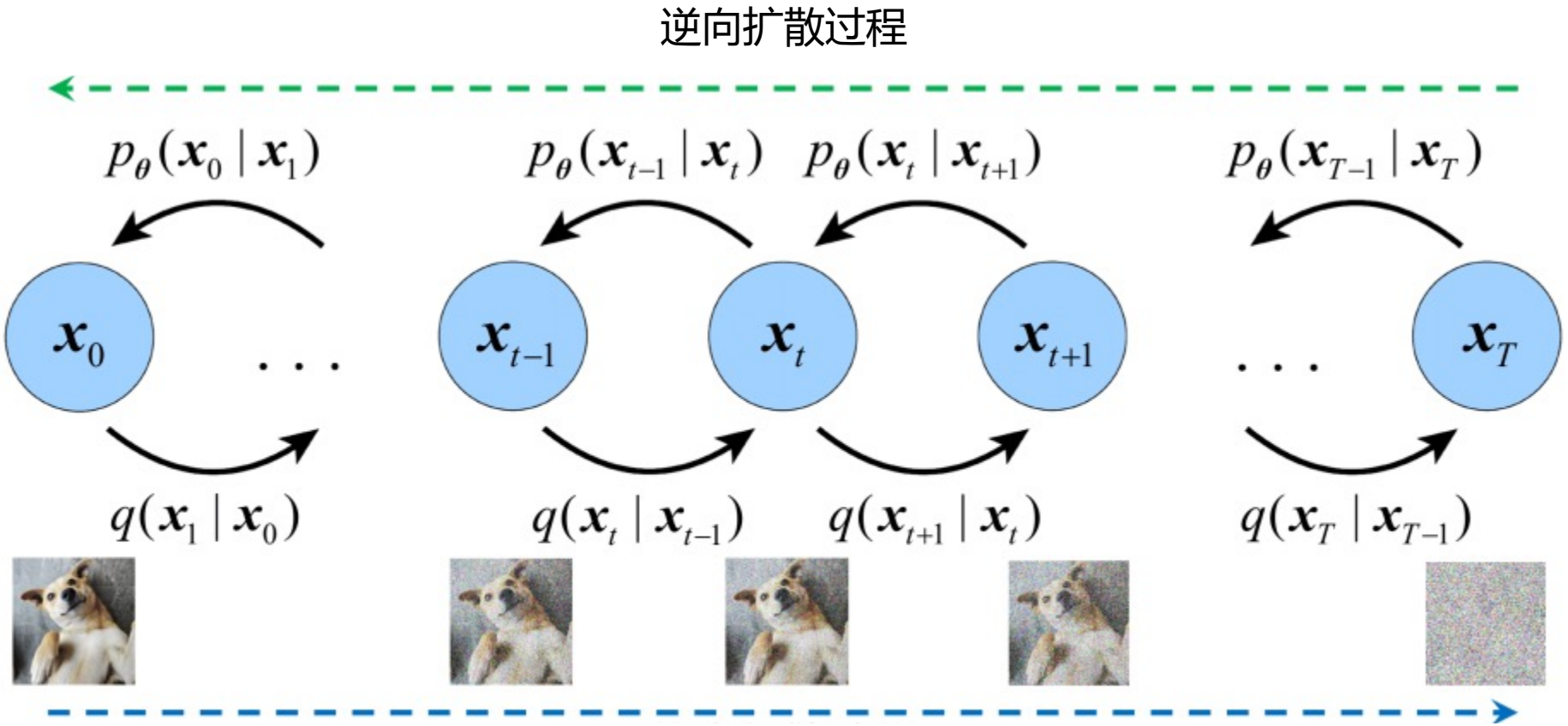
网络结构：U-Net

相比于传统的U-Net结构

- 组归一化（GN）替代批量归一化（BN）
- 引入了残差连接和注意力机制
- 将时间t编码成一个嵌入向量（如正弦位置嵌入）



训练与采样算法


逆向扩散过程
$p_\theta(x_0 \mid x_1)$
$p_\theta(x_{t-1} \mid x_t)$
$p_\theta(x_t \mid x_{t+1})$
$p_\theta(x_{T-1} \mid x_T)$
x_0
x_{t-1}
x_t
x_{t+1}
x_T
$q(x_1 \mid x_0)$
$q(x_t \mid x_{t-1})$
$q(x_{t+1} \mid x_t)$
$q(x_T \mid x_{T-1})$
正向扩散过程

训练与采样算法

算法6-1：DDPM的训练算法[Ho et al., 2020]

while 不收敛

　　从训练集中随机抽取样本 x_0；

　　从离散均匀分布采样时刻 $t \sim DU(2,T)$；

　　从标准高斯分布采样源噪声 $\varepsilon_0 \sim \mathcal{N}(0,I)$；

　　根据以下梯度进行梯度下降（Gradient Descent）

$$\nabla_\theta \| \varepsilon_0 - \hat{\varepsilon}_\theta(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\varepsilon_0, t) \|_2^2$$

end while

训练与采样算法

算法6-2： DDPM的训练算法[Ho et al., 2020]

采样高斯白噪声 $x_T \sim \mathcal{N}(0, I)$；

// 从时刻T出发，构建一条逆向马尔科夫链

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

  if $t > 1$

    采样高斯白噪声 $z \sim \mathcal{N}(0, I)$；

  else

    z = 0;

// 根据公式(6-54)以及再参数化技巧，从 x_t生成 x_{t-1}

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\hat{\varepsilon}_\theta(x_t, t)\right) + \sigma_q(t)\cdot z$$

end for

return x_0

实验效果：Fashion MNIST

VAE

DCGAN

DDPM

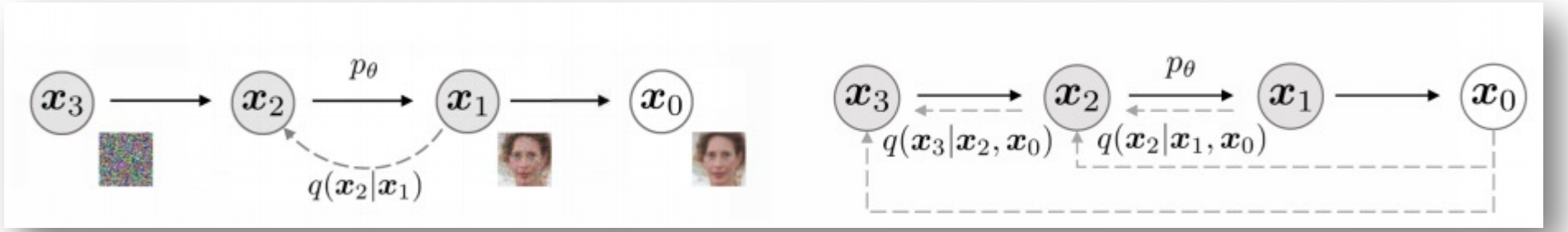
实验效果：Fashion MNIST

表 6-1 Fashion MNIST 数据集上不同生成模型的 FID 与 IS 分数

模型	类型	FID 分数（↓）	IS 分数（↑）	采样速度（张/秒） （↑）
测试集	—	0.011	9.389	—
VAE	变分自编码器	3.186	5.799	3.65×10^4
DCGAN	生成对抗网络	2.379	7.525	4.92×10^4
SNGAN	生成对抗网络	0.487	8.499	3.52×10^4
RealNVP	标准化流	1.385	7.540	3.32×10^4
DDPM	扩散模型	0.647	8.672	7.18
SGM	扩散模型	0.525	9.058	10.10

实验效果：Fashion

- 去噪扩散隐模型
- Denoising Diffusion Implicit Models（DDIM）
- DDIM和DDPM有相同的训练目标，但是它不再限制扩散过程必须是一个马尔卡夫链，这使得DDIM可以采用更小的采样步数来加速生成过程
- 牺牲多样性换取更快的推断速度



加速采样仍然值得研究

- 在某个实验中，采样179800张64×64的RGB图片
  - 基于GAN：0.012~0.031小时（43~112秒）
  - 基于DDIM：27.2小时

此实验中，GAN的采样速度是DDIM的1295倍！

条件扩散模型

- 分类器引导（Classifier Guidance）
- 无分类器引导（Classifier-Free Guidance）

目的：给扩散模型加入条件！

分类器引导

- Ablated Diffusion Model（ADM, 2021）
- 相对于DDPM有两方面改进：
  - 扩散模型结构的改进
  - 利用一个**预训练的分类器**，从预训练的DDPM采样
- **意义**：性能媲美甚至超过GAN、进行基于类别标签的图像生成

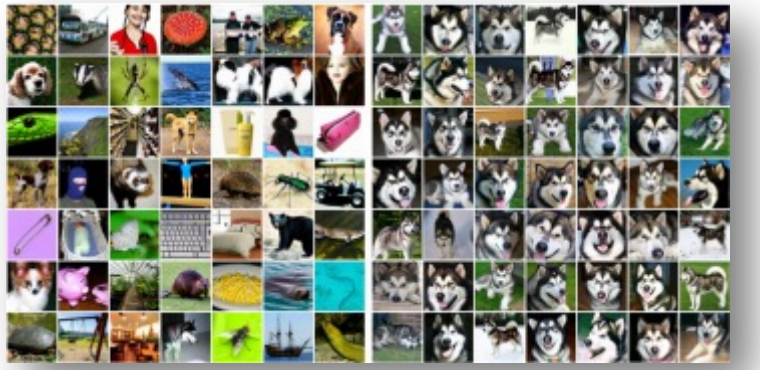
Dhariwal, Prafulla, and Alexander Nichol. "Diffusion models beat GANs on image synthesis." ***NeurIPS 2021***. [**OpenAI**]

无分类器引导

- **不再依赖预训练分类器**也可以进行基于条件的生成式建模
- 基于对**Stein’ s分数函数**的分解

$$\nabla \log p(\boldsymbol{x}_t \mid \boldsymbol{y}) = \gamma \cdot \underbrace{\nabla \log p(\boldsymbol{x}_t \mid \boldsymbol{y})}_{\text{舫锄技竦}} + (1-\gamma) \cdot \underbrace{\nabla \log p(\boldsymbol{x}_t)}_{\text{蚯舫锄技竦}}$$

- **意义：**是DALL·E 2、Imagen、Stable Diffusion等大模型的组成部分。



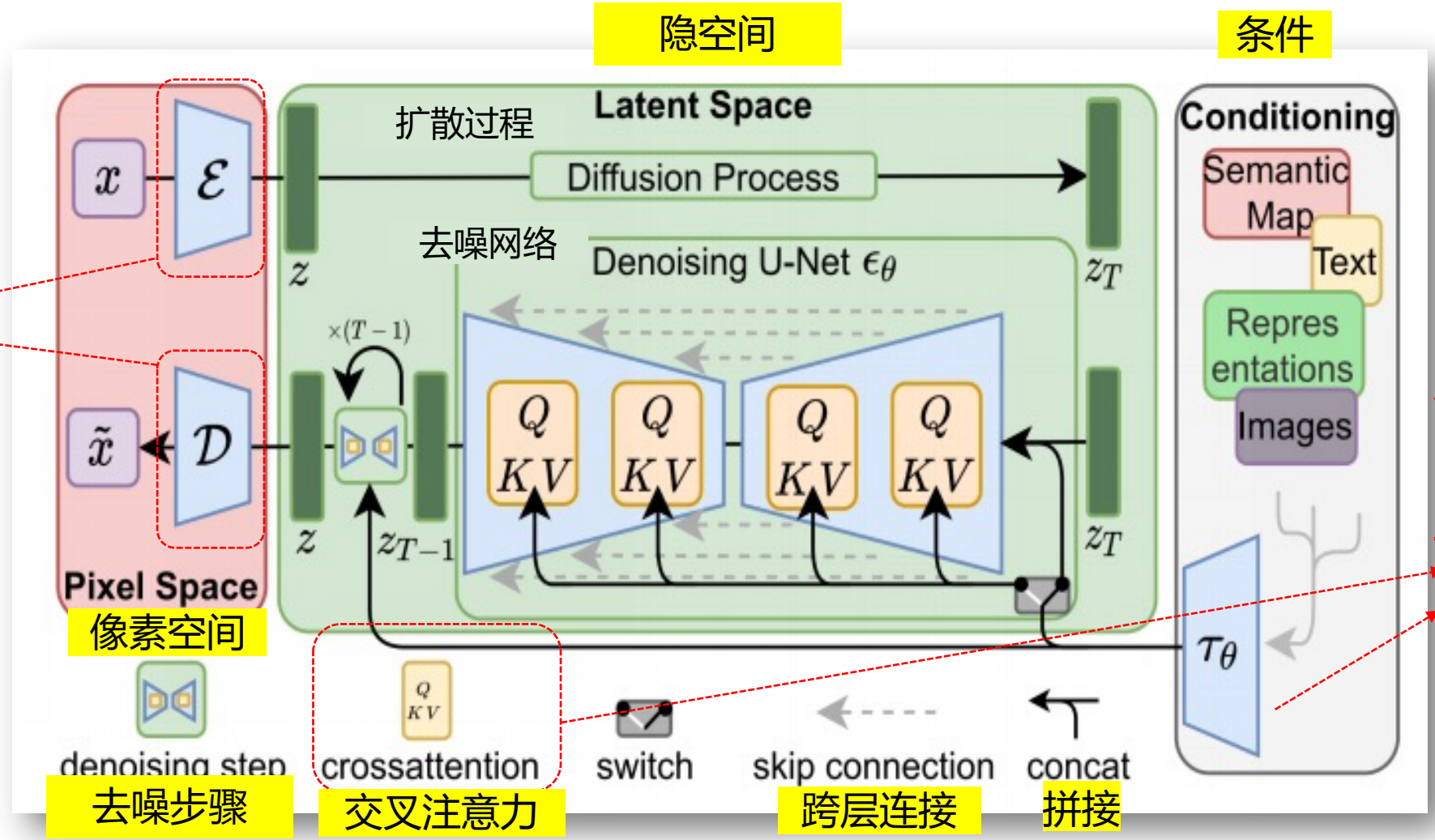
Ho, Jonathan, and Tim Salimans. "Classifier-Free Diffusion Guidance." NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications. 2021.

隐扩散模型

- Latent Diffusion Models （LDM, 2022）
- **将扩散过程放在低维度的隐空间**（latent space），可降低训练和推理代价。
- **新的条件输入机制**：类别标签、文字、布局等。
- 可完成无条件图片生成、文生图、图片修复、图片超分等任务。
- 可生成更细致、更高分辨率的图像。

Rombach, Robin, et al. “High-resolution image synthesis with latent diffusion models.” ***CVPR 2022***.

隐扩散模型


隐空间
条件
语义图
文本
表征
图片
.....
编码器降维
解码器升维
扩散过程
Latent Space
Diffusion Process
去噪网络
Denoising U-Net ϵ_θ
Conditioning
Semantic Map
Text
Repres entations
Images
Pixel Space
像素空间
领域专用编码器与交叉注意力
denoising step
crossattention
switch
skip connection
concat
去噪步骤
交叉注意力
跨层连接
拼接

隐扩散模型

隐扩散模型

提示词：一幅具有毕加索风格的最后的晚餐

'A painting of the last supper by Picasso.'