

生成模型与图像生成

陆轶 & 杜亚磊

- ▶ 生成模型(Generative Models)简介
- ▶ 变分自编码器(Variational Auto-Encoder, VAE)
- ▶ 生成对抗神经网络(Generative Adversarial Nets, GANs)
- ▶ 图片生成案例

生成模型可用于图像，语言和文本的修饰，补全，和生成等。

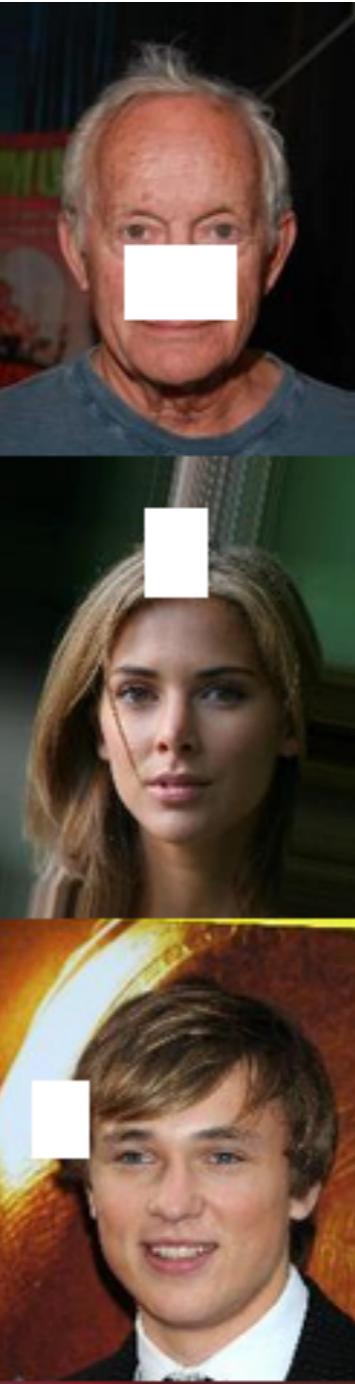
- ▶ 图像补全
- ▶ 超分辨率图像
- ▶ 图像转换
- ▶ 图像生成
- ▶ 语音生成

图像补全 [DCGAN]

原图



模型输入

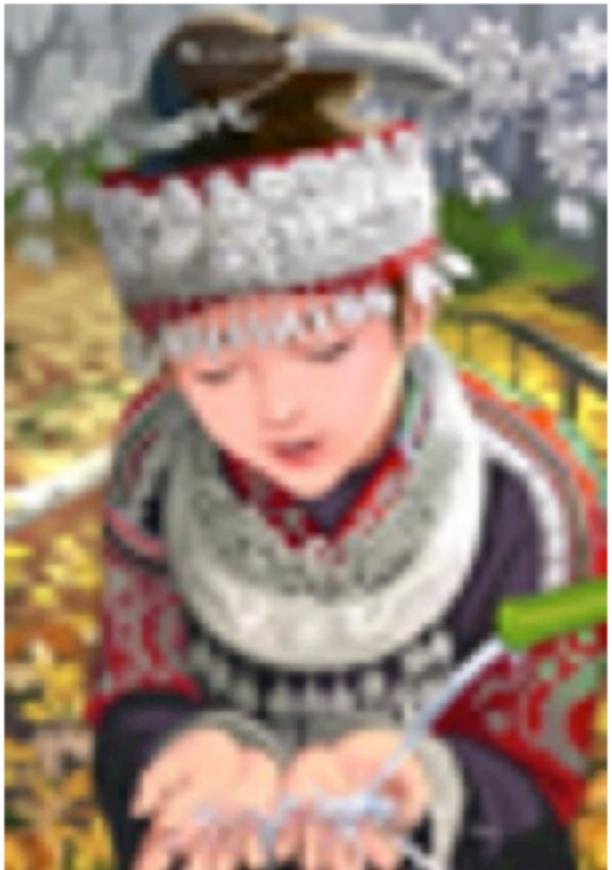


模型输出



超分辨率 [GAN]

bicubic
(21.59dB/0.6423)



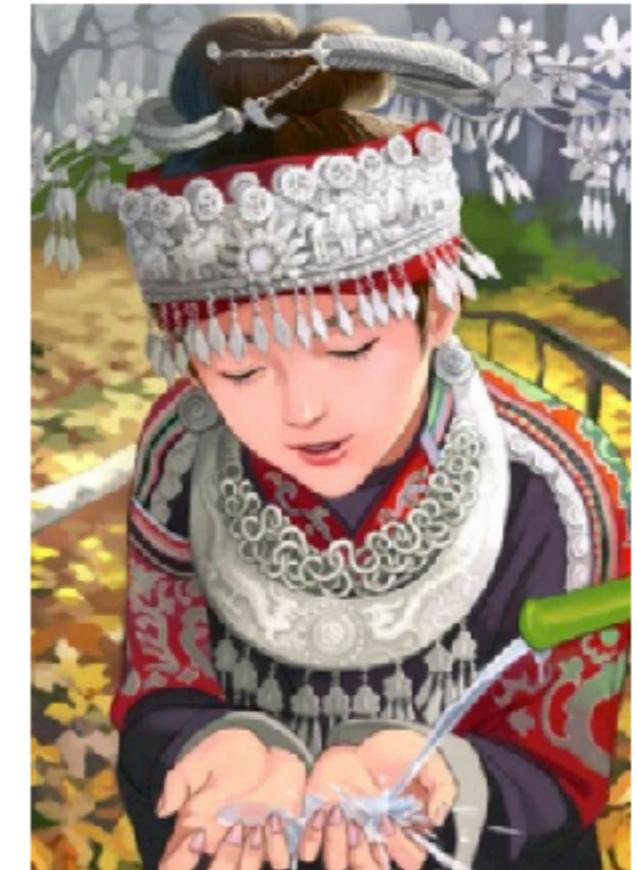
SRResNet
(23.53dB/0.7832)



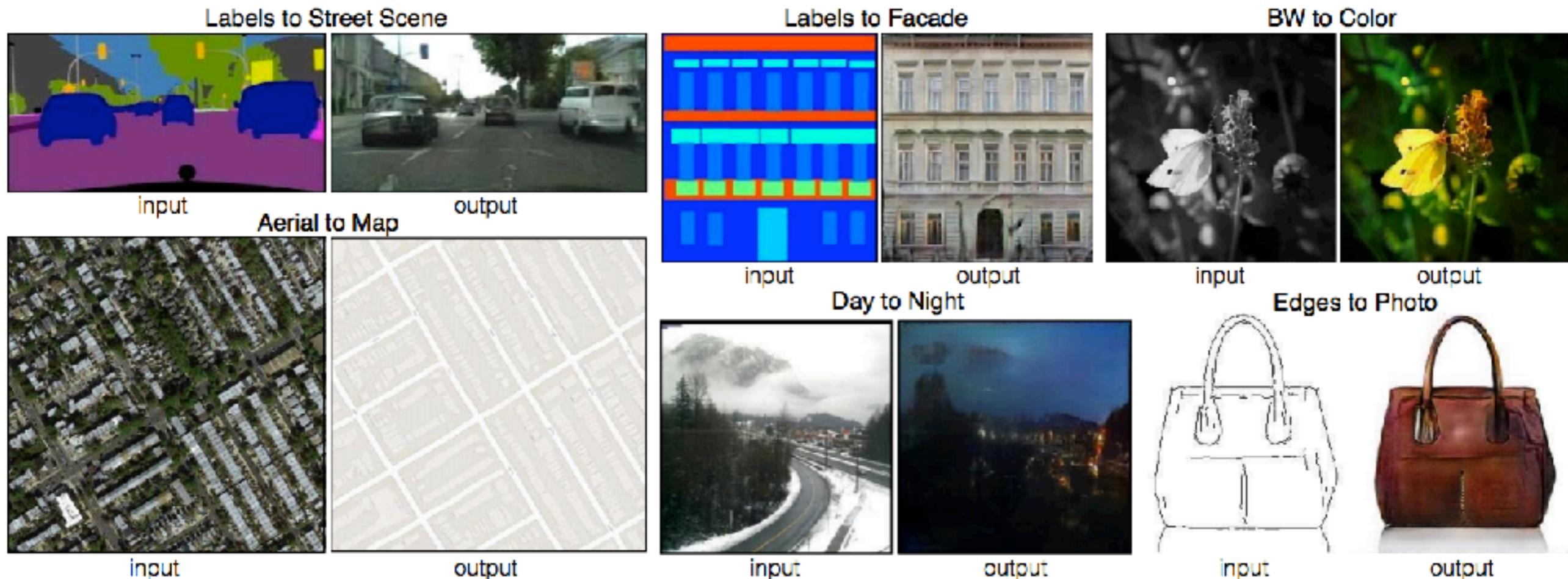
SRGAN
(21.15dB/0.6868)



original



风格转换(图像翻译?) [CONDITIONAL ADVERSARIAL NETWORKS]

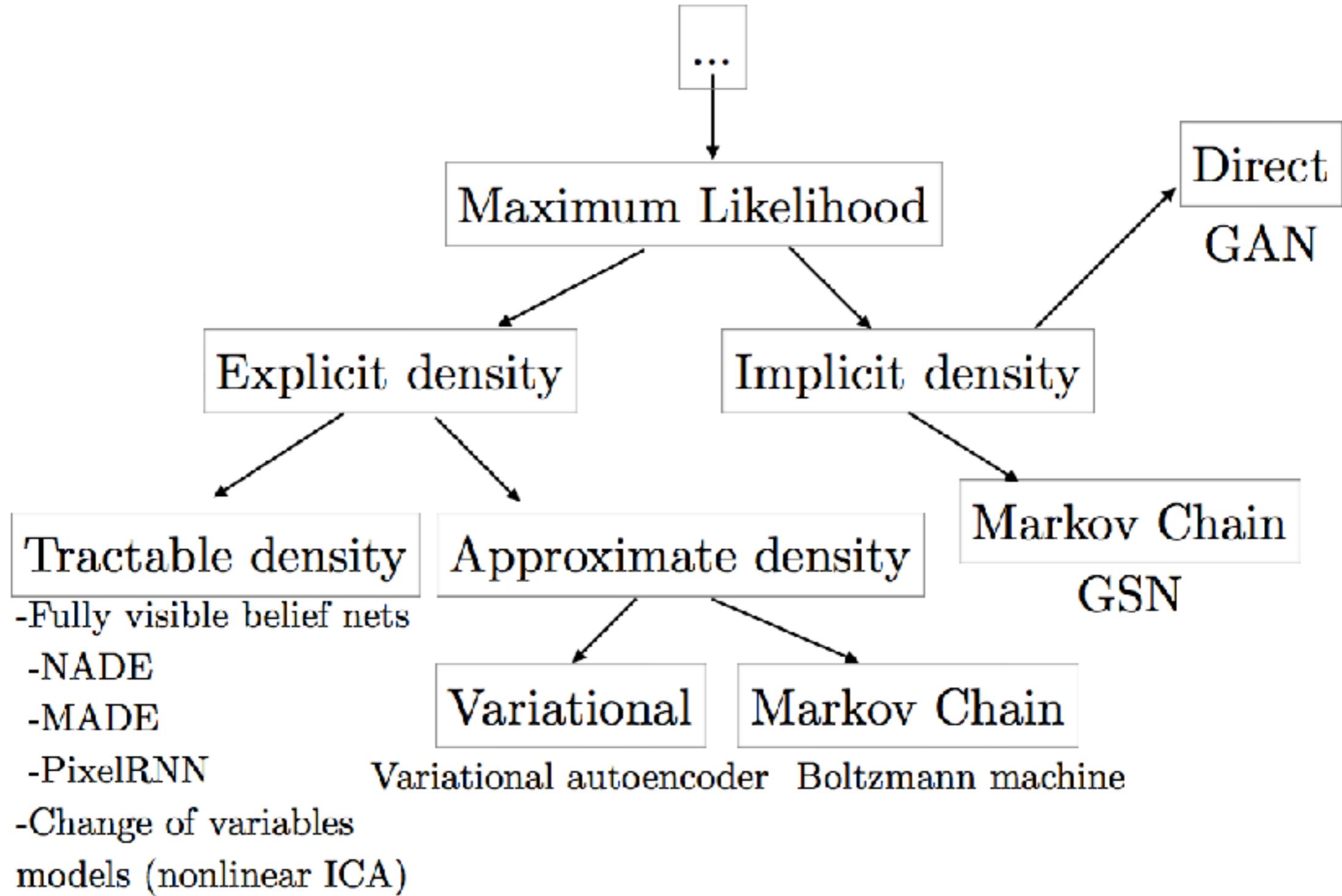


GANS是生成模型，但
生成模型不只只有GAN

杜亚磊

核心：最大似然估计

- ▶ 似然函数的表现形式(或逼近形式)
 - ▶ 似然函数
 - ▶ 似然函数有明确的形式且易于处理(DBN, PixelRNN, etc)
 - ▶ 构造似然函数，想办法逼近密度函数(VAE, Markov Chain)
- ▶ 不直接最大化似然函数
 - ▶ 马尔可夫链模拟(Generative Stochastic Networks, GSN)
 - ▶ GAN(Generative Adversarial Networks)



最大似然估计

- ▶ 给定分布形式，基于事实样本推断分布参数的一种方法
- ▶ 思想：寻找参数使得在随机事件中，样本出现的概率最大

最大似然估计

- ▶ 问题：给定一枚不均匀硬币，投十次九正一反，估计投币出现正反的概率。
- ▶ 给定分布形式(二项分布)，基于事实样本(投十次九正一反)推断分布参数(出现正反的概率)的一种方法。

最大化 $p^9 * (1-p)$

其中 $0 \leq p \leq 1$

最大似然估计

- ▶ 问题：给定一组图片，推断图片中像素的概率分布

最大似然估计

- ▶ 问题：给定一组图片，推断图片中像素的概率分布



像素的分布未知

VARIATIONAL AUTO-ENCODER

变分自编码

模型思想：

- ▶ 假设像素分布可以通过 $g(z)$ 来表示， z 是已知分布的随机变量
- ▶ 在计算出 $g(z)$ 的情况下，从 z 中取样，送入函数 $g(z)$ 生成新样本
- ▶ z 通常取多维正态分布

变分自编码

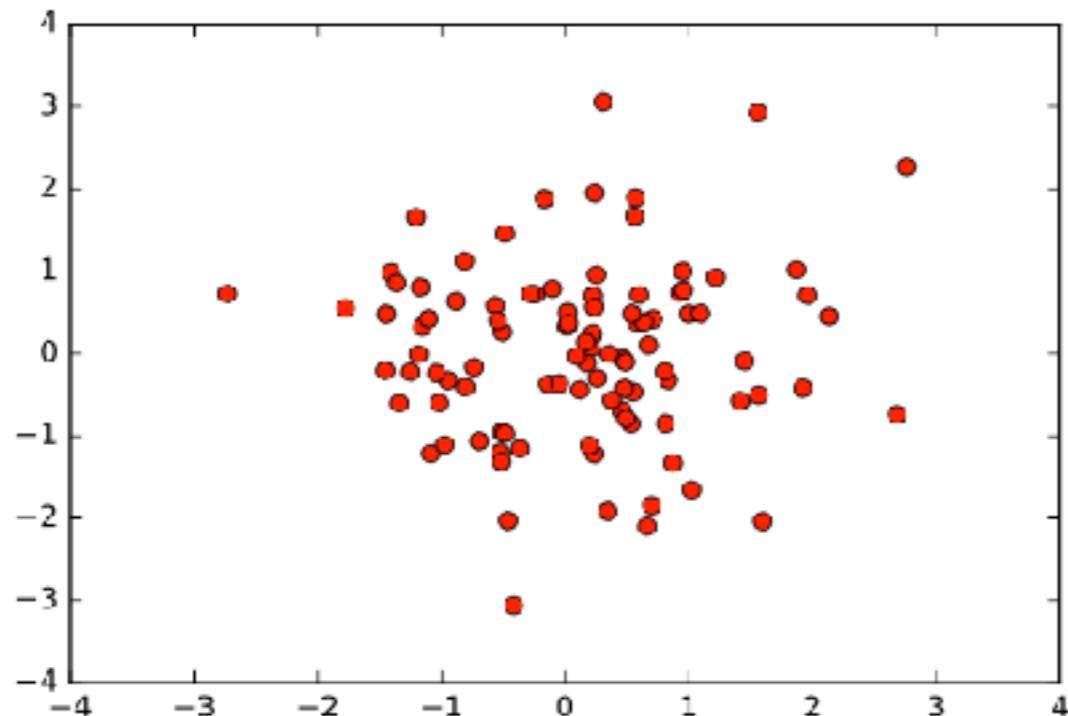


Figure 1: 高斯随机变量 z

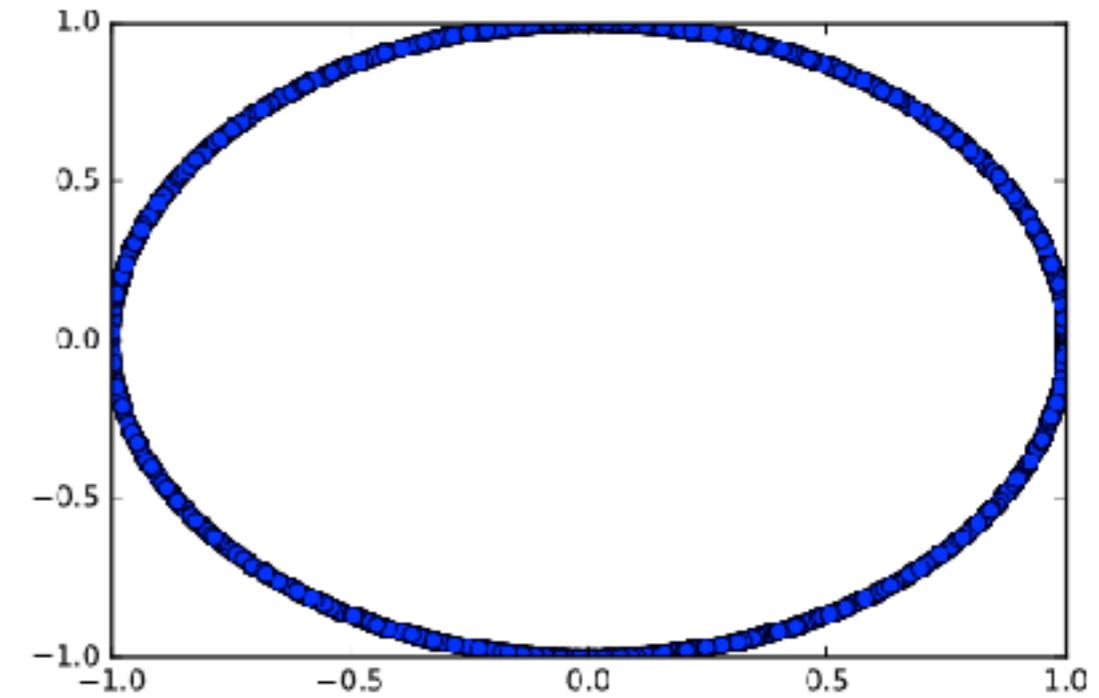


Figure 2: 将 z 映射到 x

其中 $z \sim \mathcal{N}(\mu, \sigma^2)$ 且 $|z| > \epsilon$, $x = z/\|z\| = g(z)$

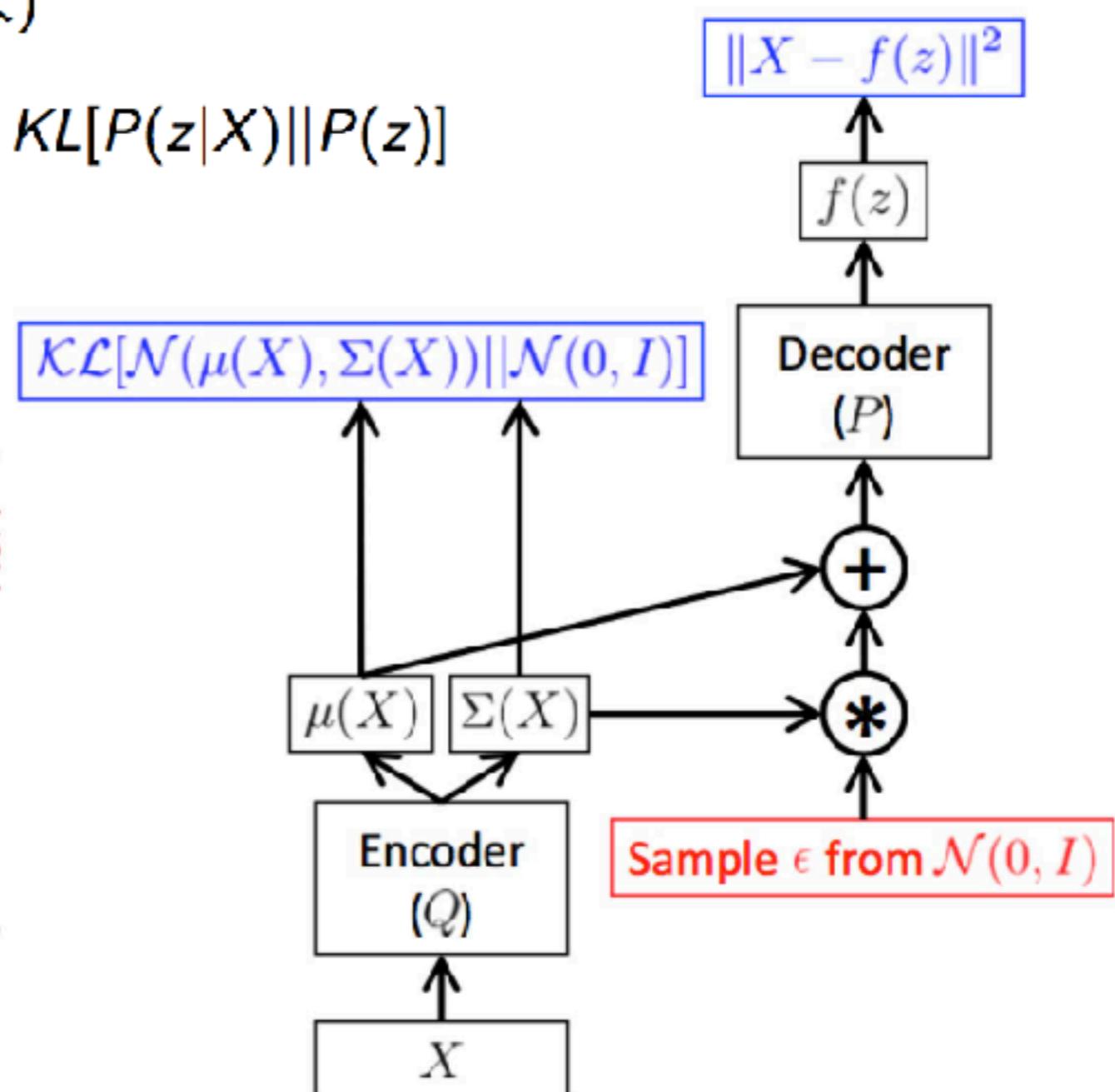
我们希望可以训练一个神经网络将一个分布近似地映射到另一个分布上, 扮演 $g(z)$ 的角色。当然, 当 $z \rightarrow 0$ 时, 这样的映射也会出现问题。

模型形式：

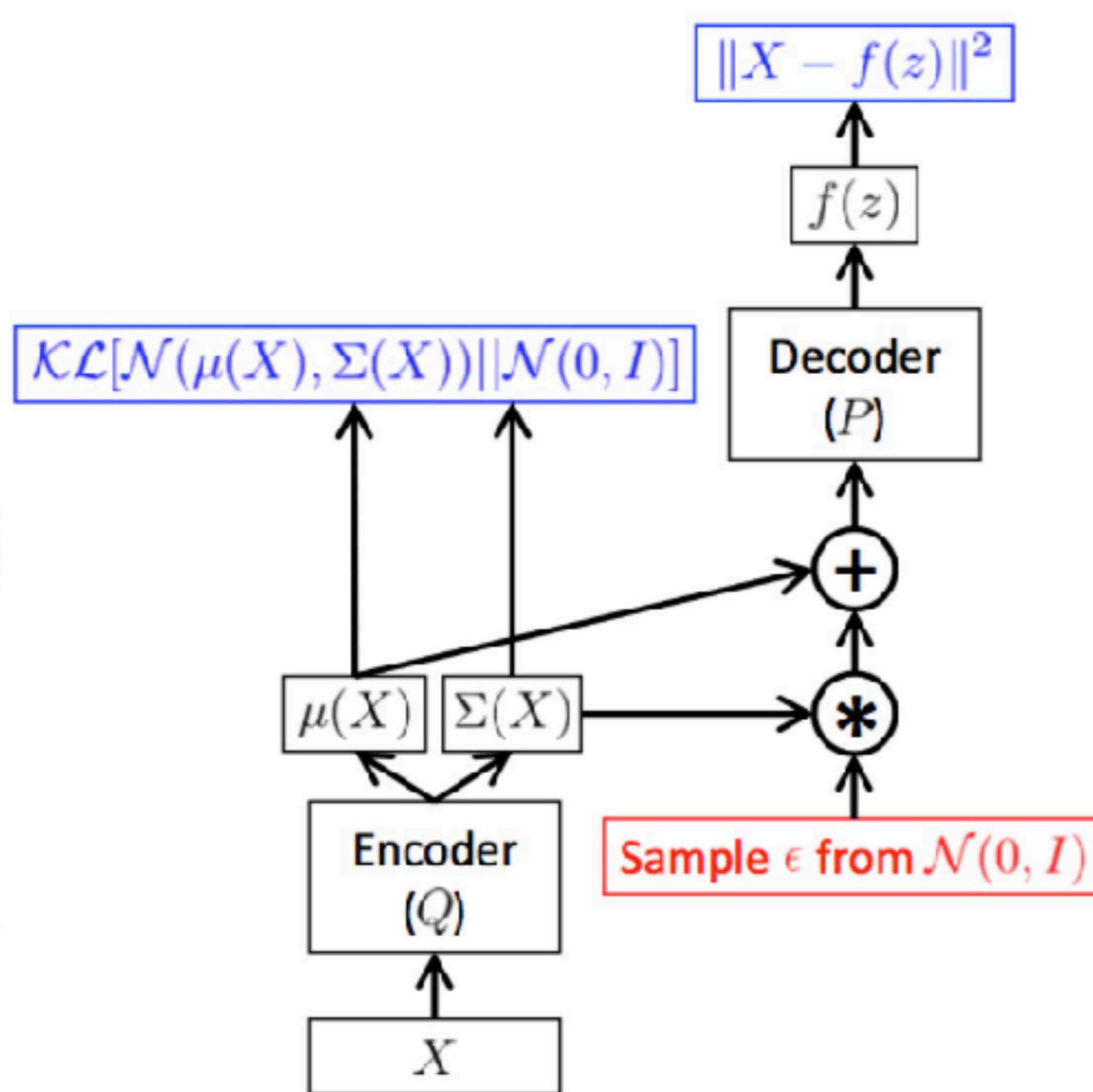
训练目标(假设网络的拟合性足够强大)

$$\max \log P(X) = \mathbb{E}_z[\log P(X|z)] - KL[P(z|X)||P(z)]$$

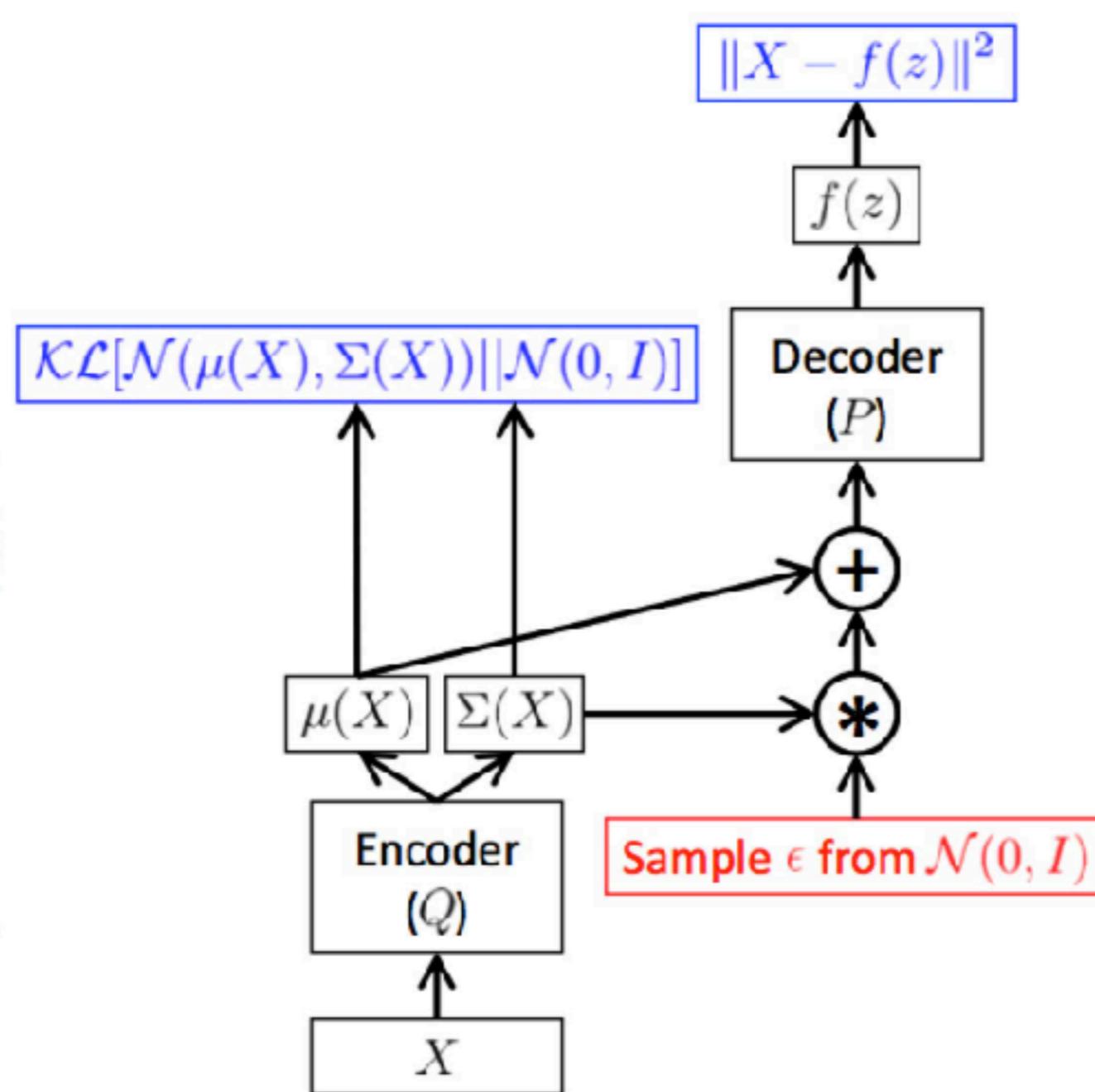
不严格相等，逼近



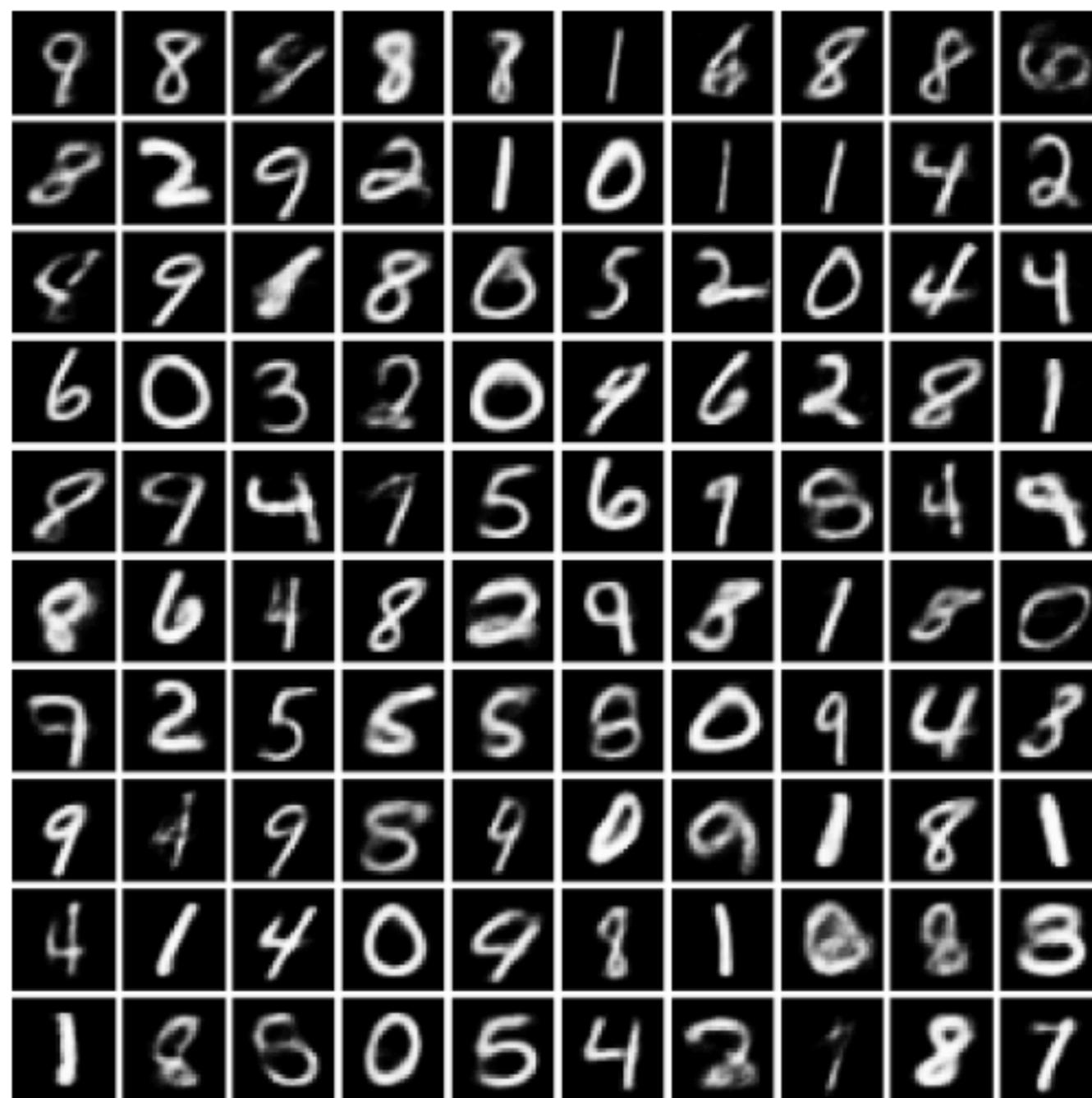
参数估计：后向传播 & 梯度下降，局部最优



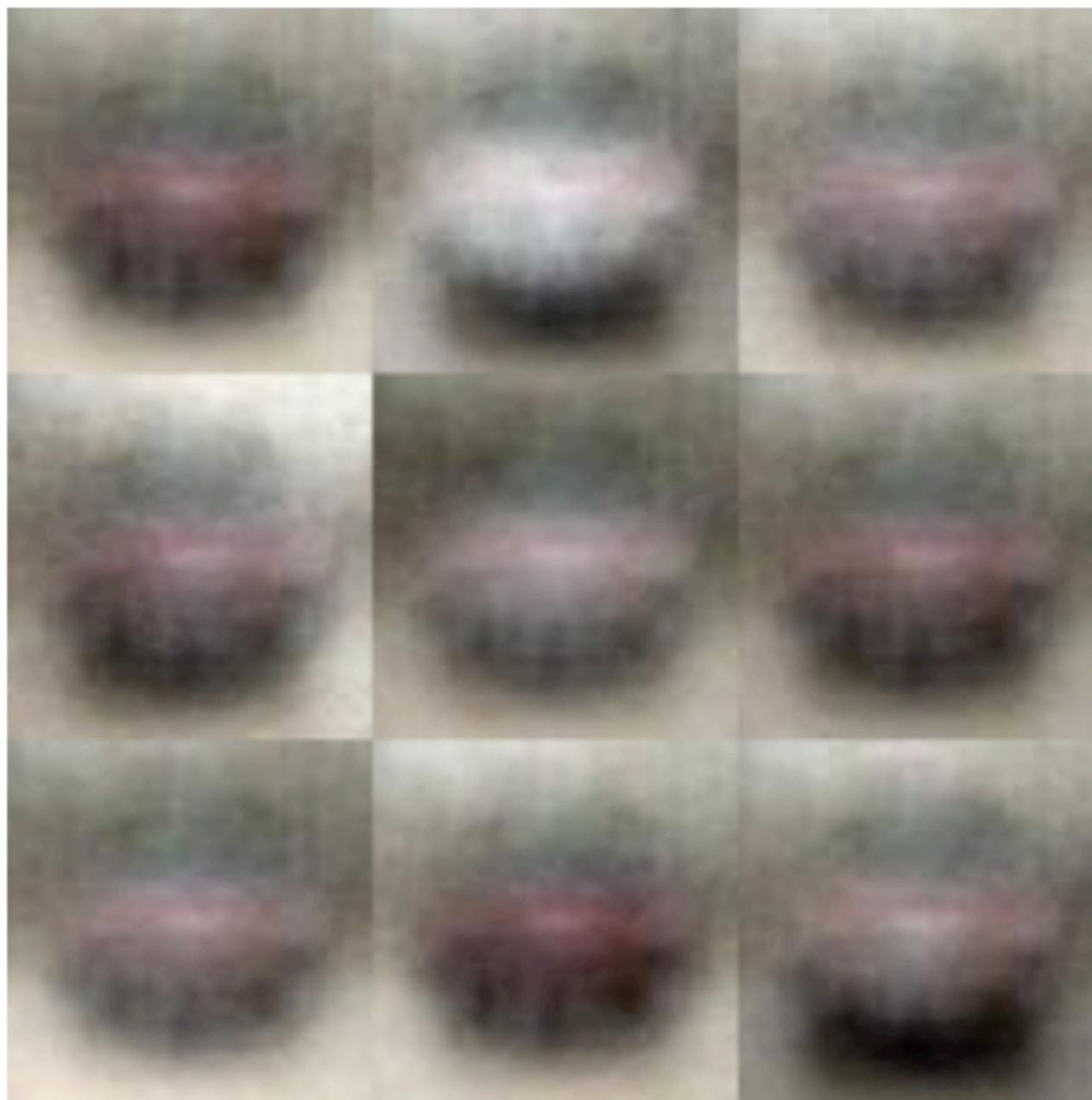
思考：为什么需要编码(ENCODER)层？



手写数字(MINIST数据集)生成样本：



汽车数据集生成样本：



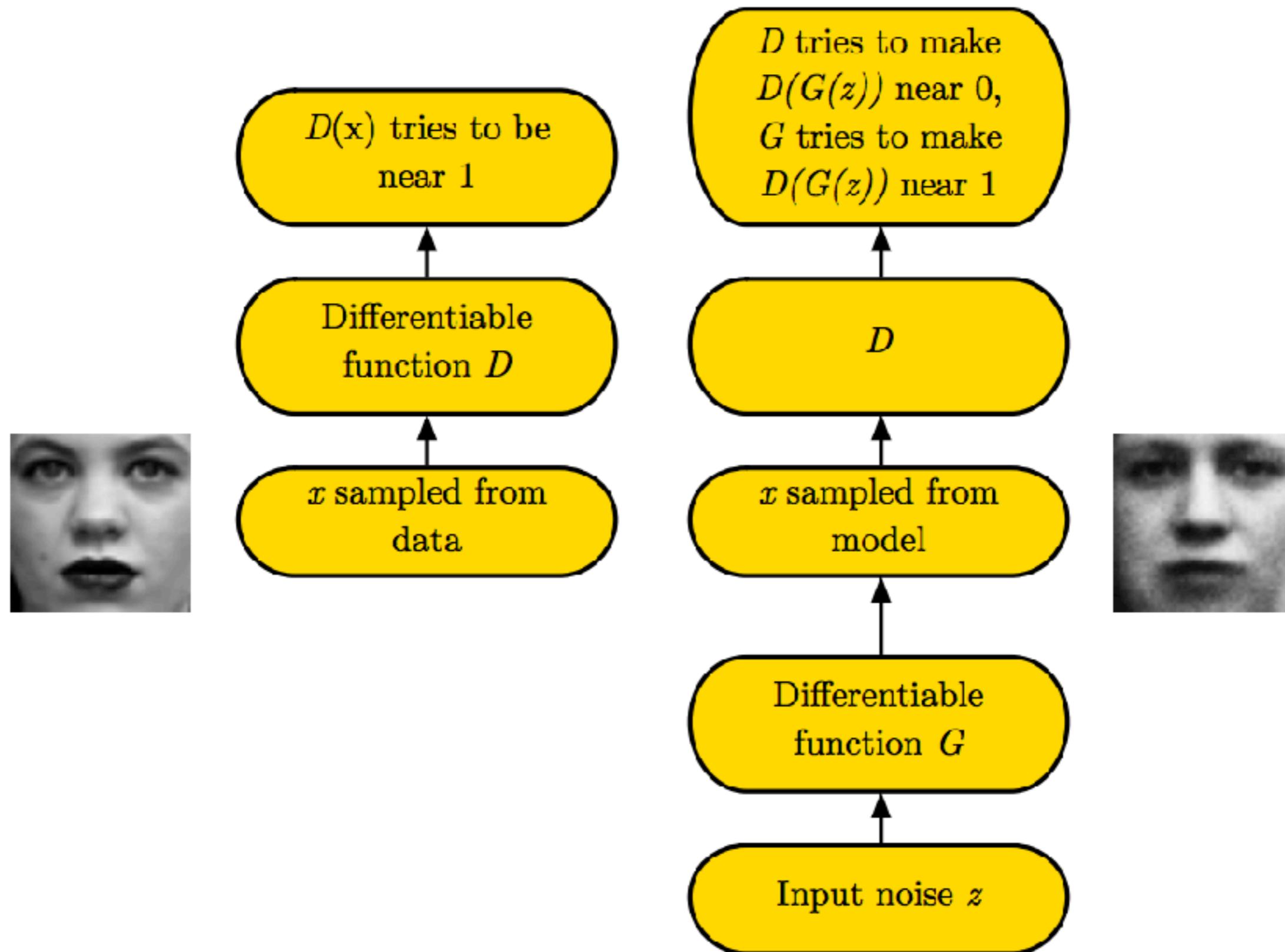
GENERATIVE ADVERSARIAL NETS

对抗神经网络

模型思想：

- ▶ 两个组件：生成网络G，判别网络D
- ▶ 生成网络G将随机变量的分布映射到图像分布，伪造图像
- ▶ 判别网络D判别真实图像与G生成的伪造图像
- ▶ 相互对抗：G试图欺瞒D，D试图识破G

对抗神经网络



模型形式：

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim P_{data}} \log(D(x)) + \mathbb{E}_{z \sim \mathcal{N}} \log(1 - D(G(z)))$$

► G 的代价函数

$$Loss_G = \mathbb{E}_{z \sim \mathcal{N}} - \log(D(G(z)))$$

G 生成的图片越是逼真， $D(G(z))$ 就越大， $Loss_G$ 就越小，其中 z 在高斯分布中随机取样。

► D 的代价函数

$$Loss_D = -[\mathbb{E}_{x \sim P_{data}} \log(D(x)) + \mathbb{E}_{z \sim \mathcal{N}} \log(1 - D(G(z)))]$$

左式表示 D 需要能够鉴真($D(x) \rightarrow 1$)，右式表示 D 同时也需要防伪($D(G(z)) \rightarrow 0$)。 x, z 来自于随机取样。

参数估计：后向传播 & 交叉更新(G和D)

for number of training iterations **do**

for k steps **do**

- Sample minibatch of m noise samples $\{\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(m)}\}$ from noise prior $p_g(\mathbf{z})$.
- Sample minibatch of m examples $\{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\mathbf{x})$.
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^m \left[\log D(\mathbf{x}^{(i)}) + \log (1 - D(G(\mathbf{z}^{(i)}))) \right].$$

end for

- Sample minibatch of m noise samples $\{\mathbf{z}^{(1)}, \dots, \mathbf{z}^{(m)}\}$ from noise prior $p_g(\mathbf{z})$.
- Update the generator by descending its stochastic gradient:

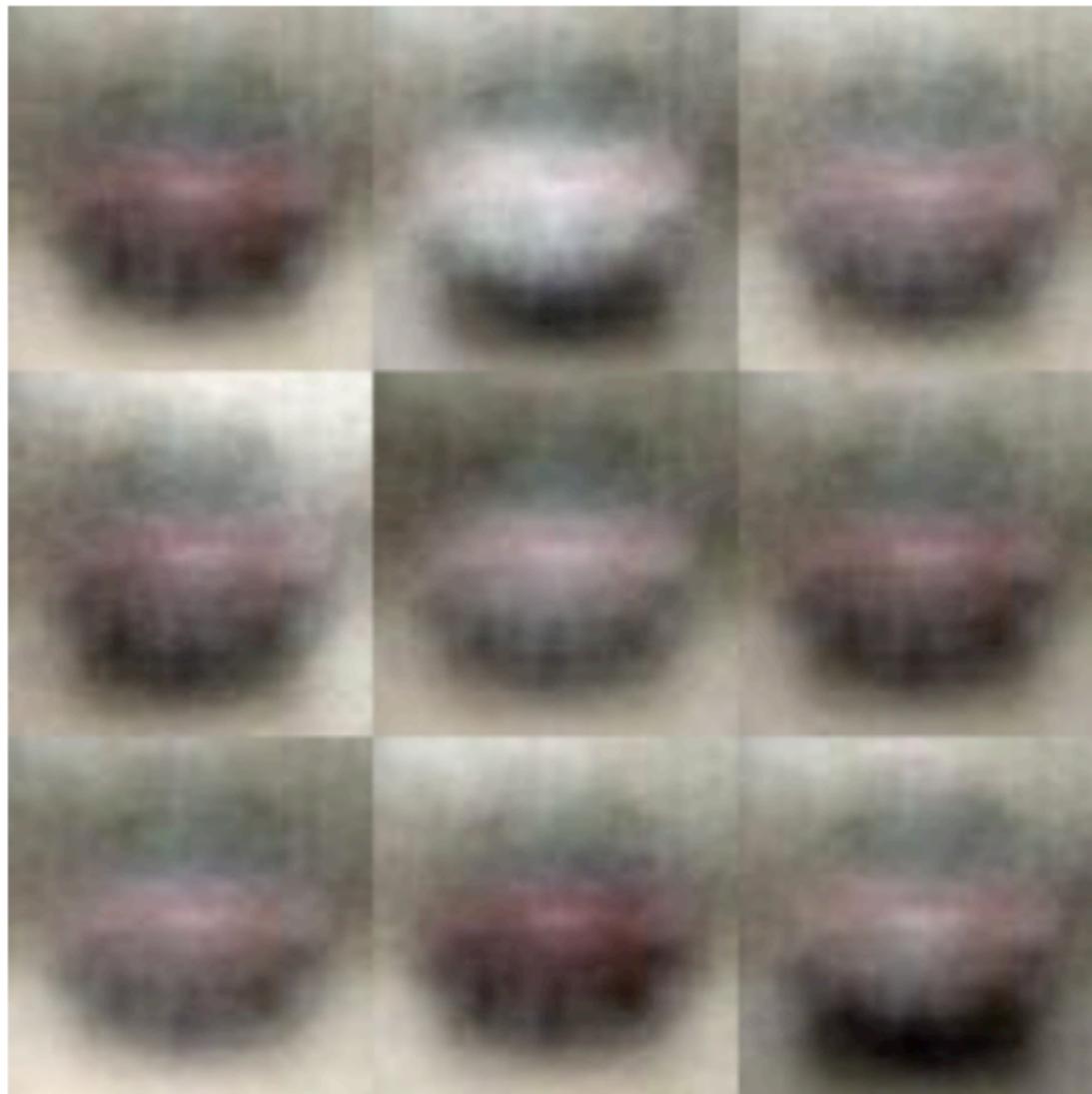
$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^m \log (1 - D(G(\mathbf{z}^{(i)}))).$$

思考：

- ▶ 交叉更新D和G(目标并不一致), 参数是否会来回震荡?

图片生成案例

案例

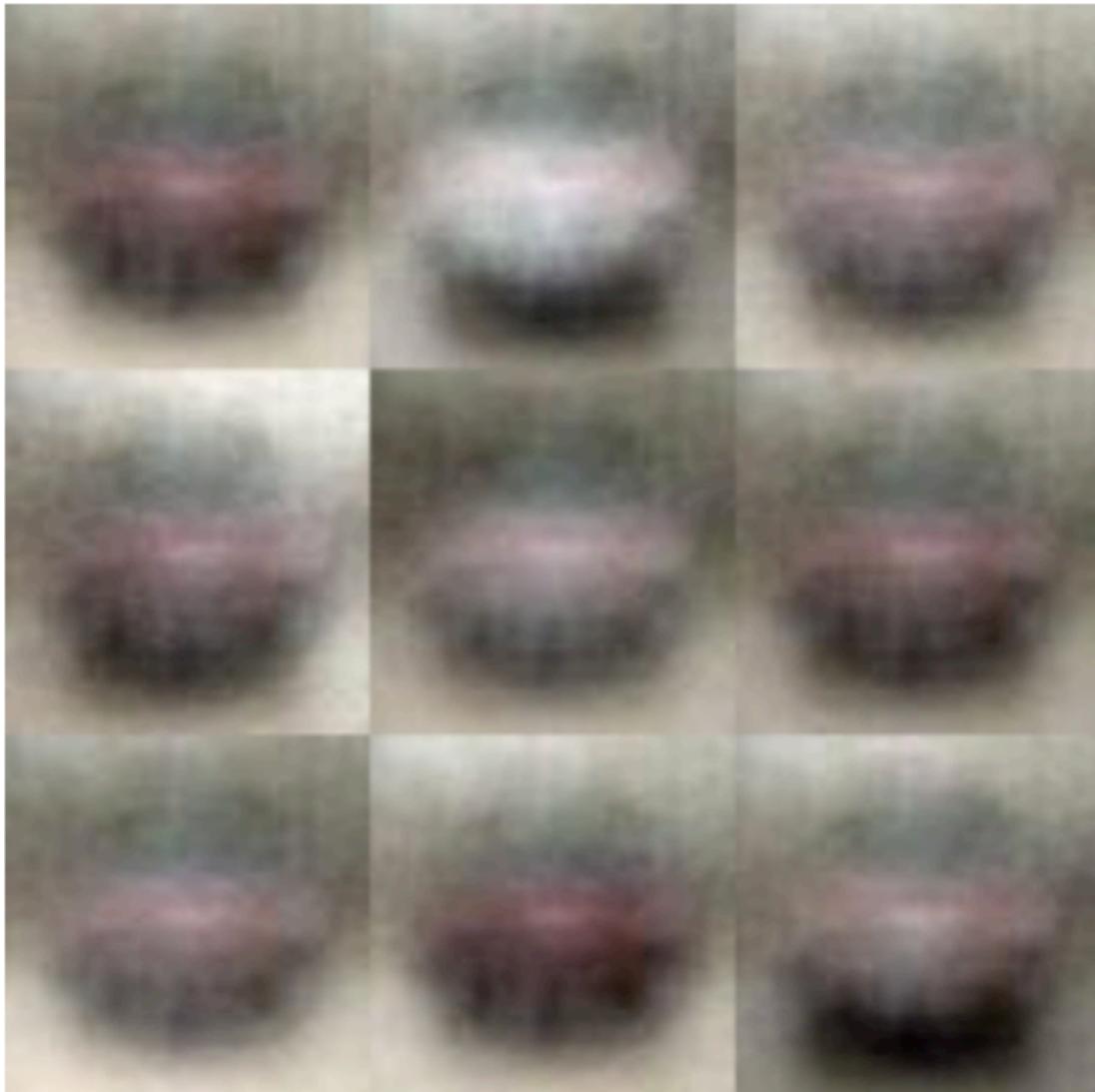


VAE



GAN

案例



VAE



GAN

思考：VAE和GAN为何会有如此大的差异？

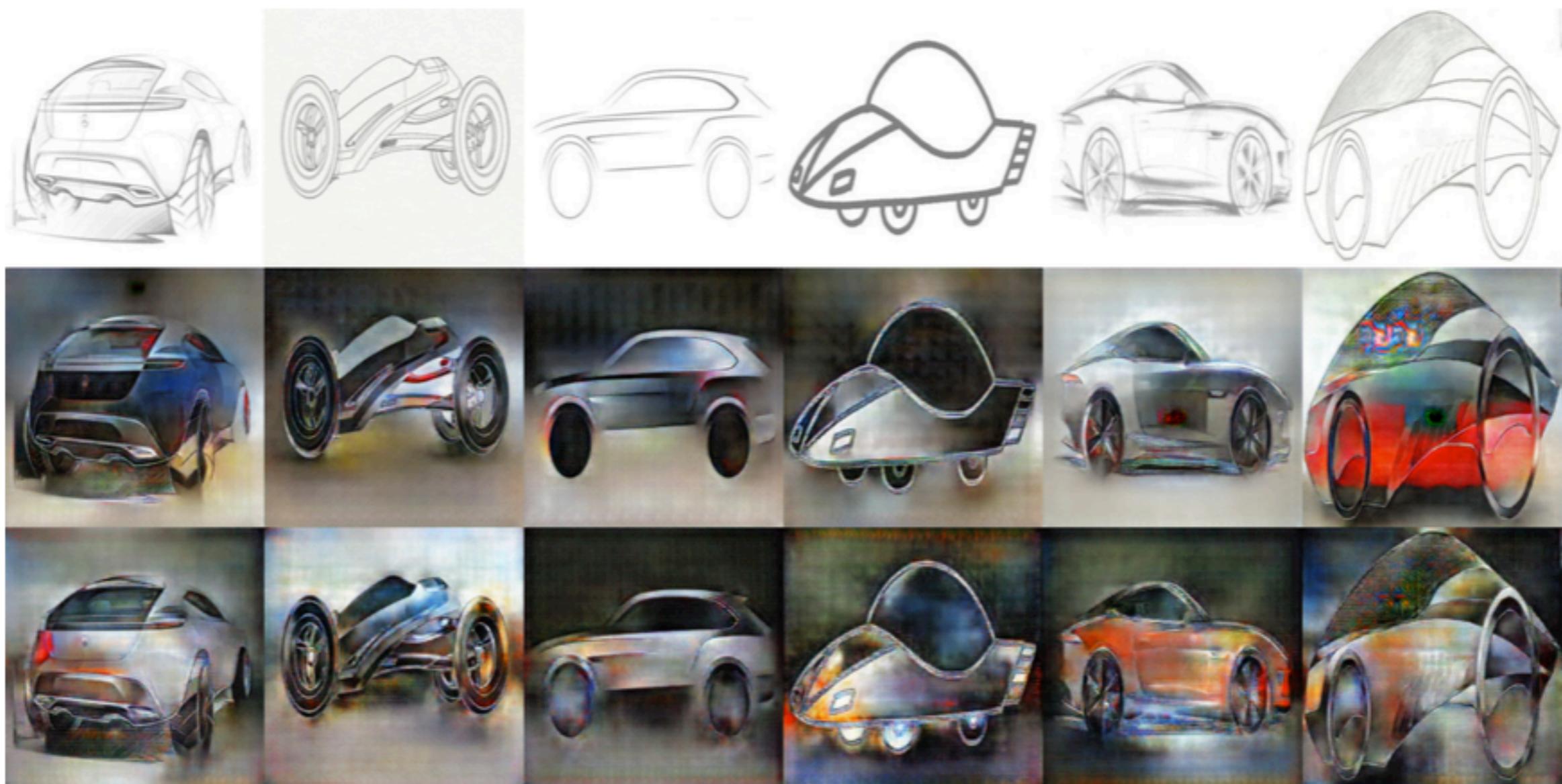


Figure 6: CGAN生成鞋子示例[4]

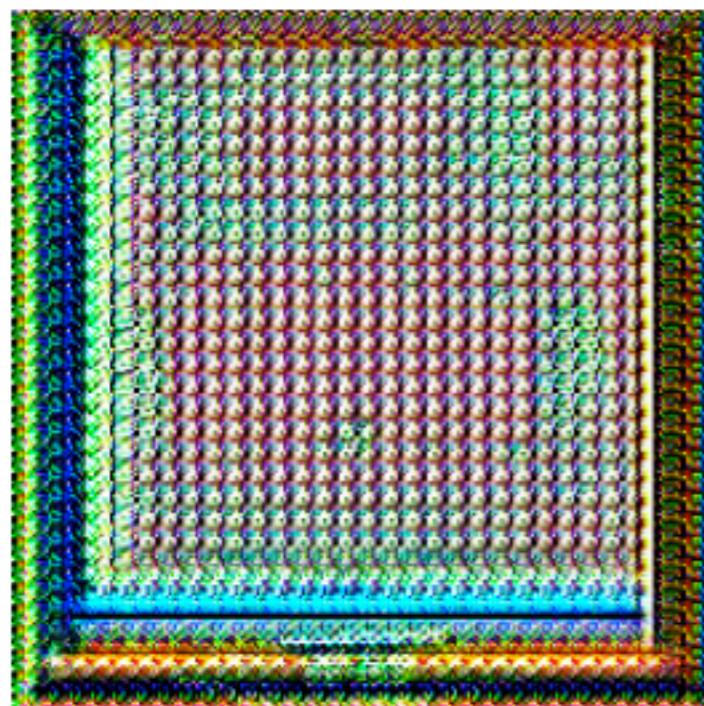
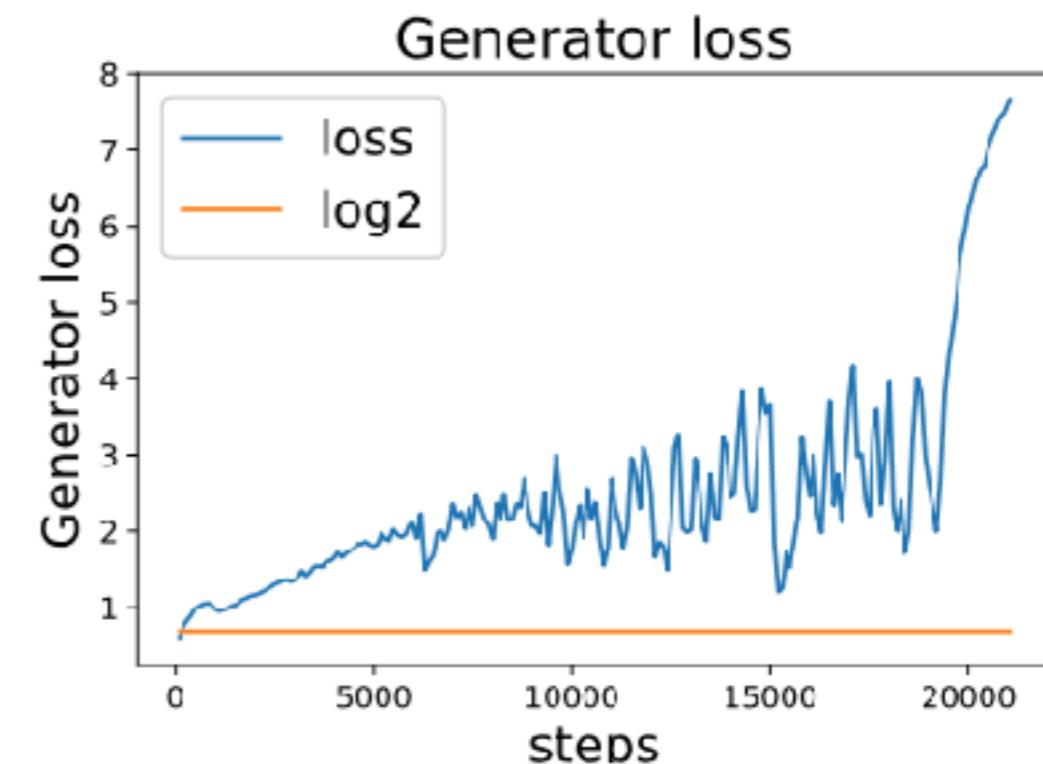
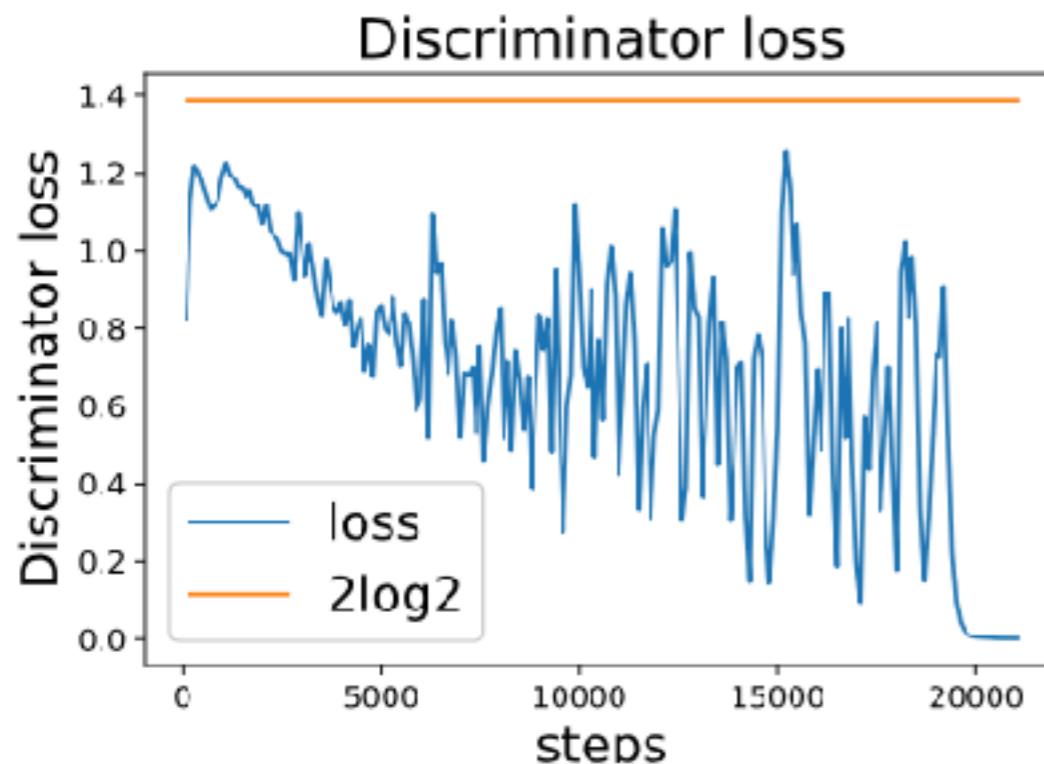
$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim P_{data}} \log(D(x|y)) + \mathbb{E}_{z \sim N} \log(1 - D(G(z|y)))$$

在代价函数中加入了 y ,而由于 y 的加入,训练将变得容易很多。
生成模型针对 y 提供的信息进行生成。

手绘生成图片实验效果

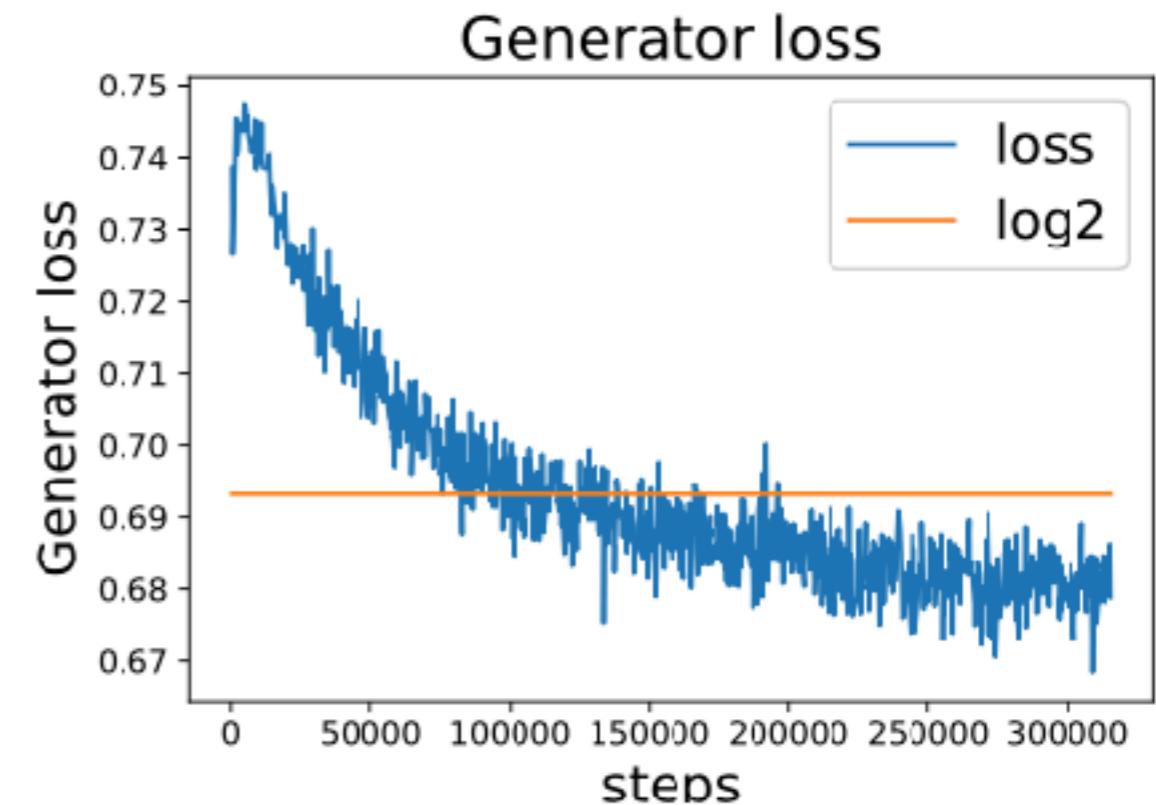
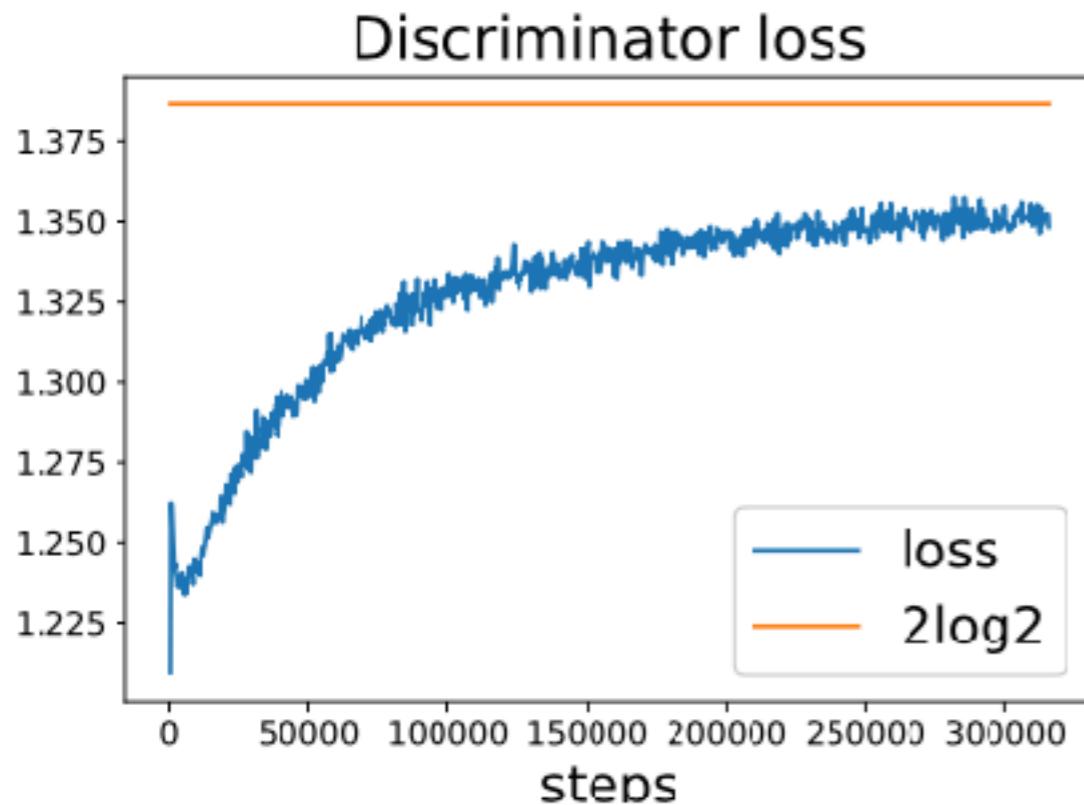


手绘生成图片训练过程



GAN的训练非常不稳定。如果 D 一旦过强， G 生成的所有结果都会被识破。平衡被打破，训练很容易就失败了。

手绘生成图片训练过程：Wasserstain loss



WASSERSTAIN LOSS：改进目标函数和优化算法

- ▶ 提出Wasserstrin Distance衡量真实数据分布与模拟分布的差异，给出更平滑的剃度
- ▶ 相互对抗：G试图欺瞒D，D试图识破G

参考文献

- ▶ [Generative Adversarial Networks](#)
- ▶ 其他文献参见另一文件。