《神经网络与深度学习》

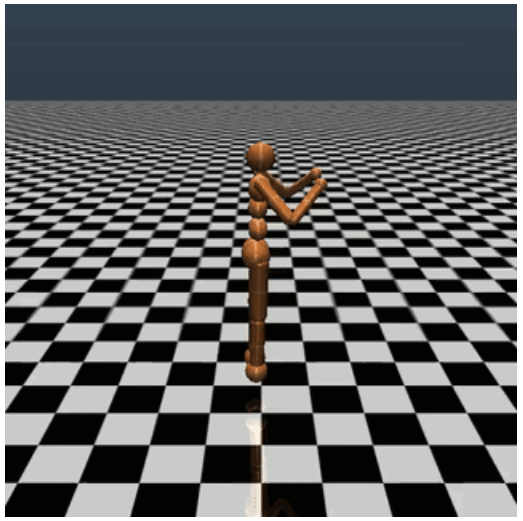
深度生成模型
Deep Generative Models

https://nndl.github.io/

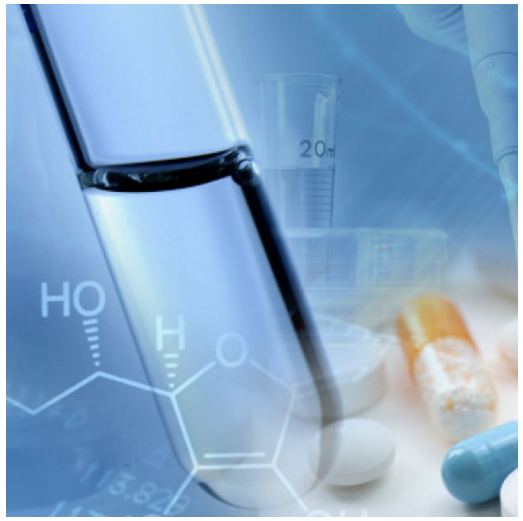
深度生成模型

AI绘画

AI配音



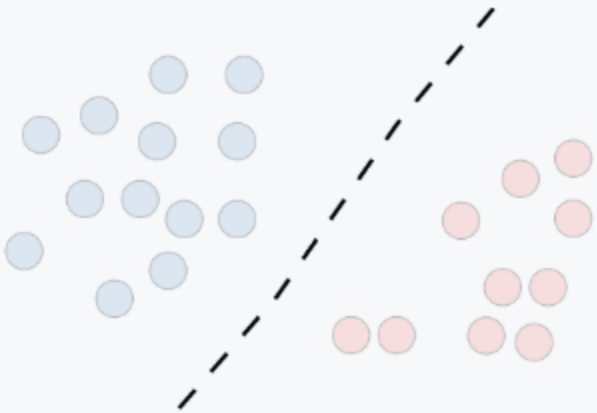
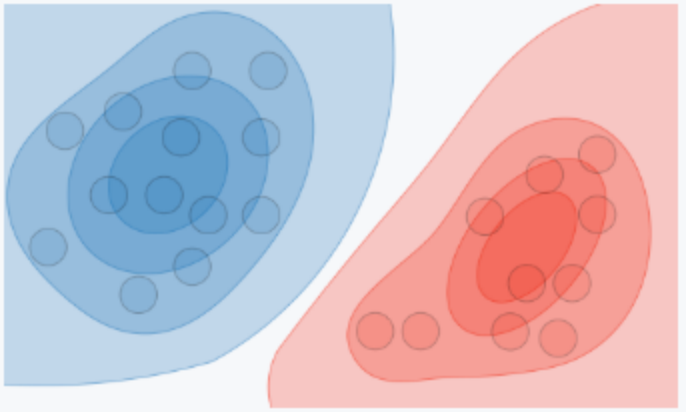
模仿学习



生物医药

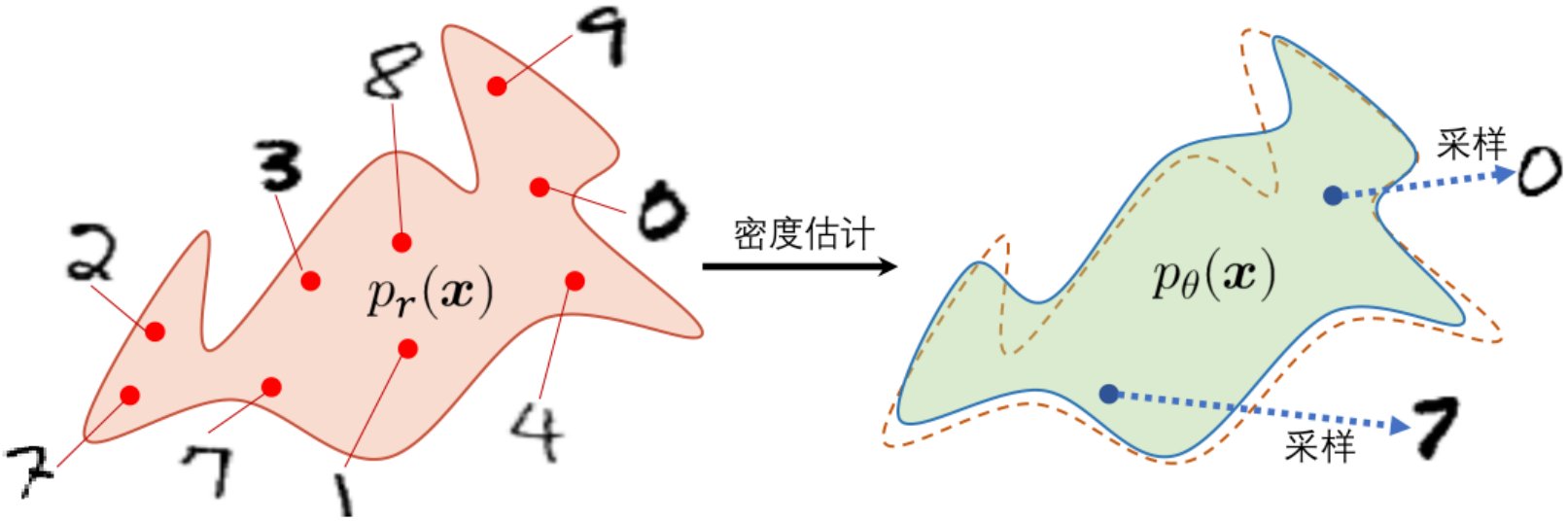
生成模型
Generative Models

机器学习的两种范式

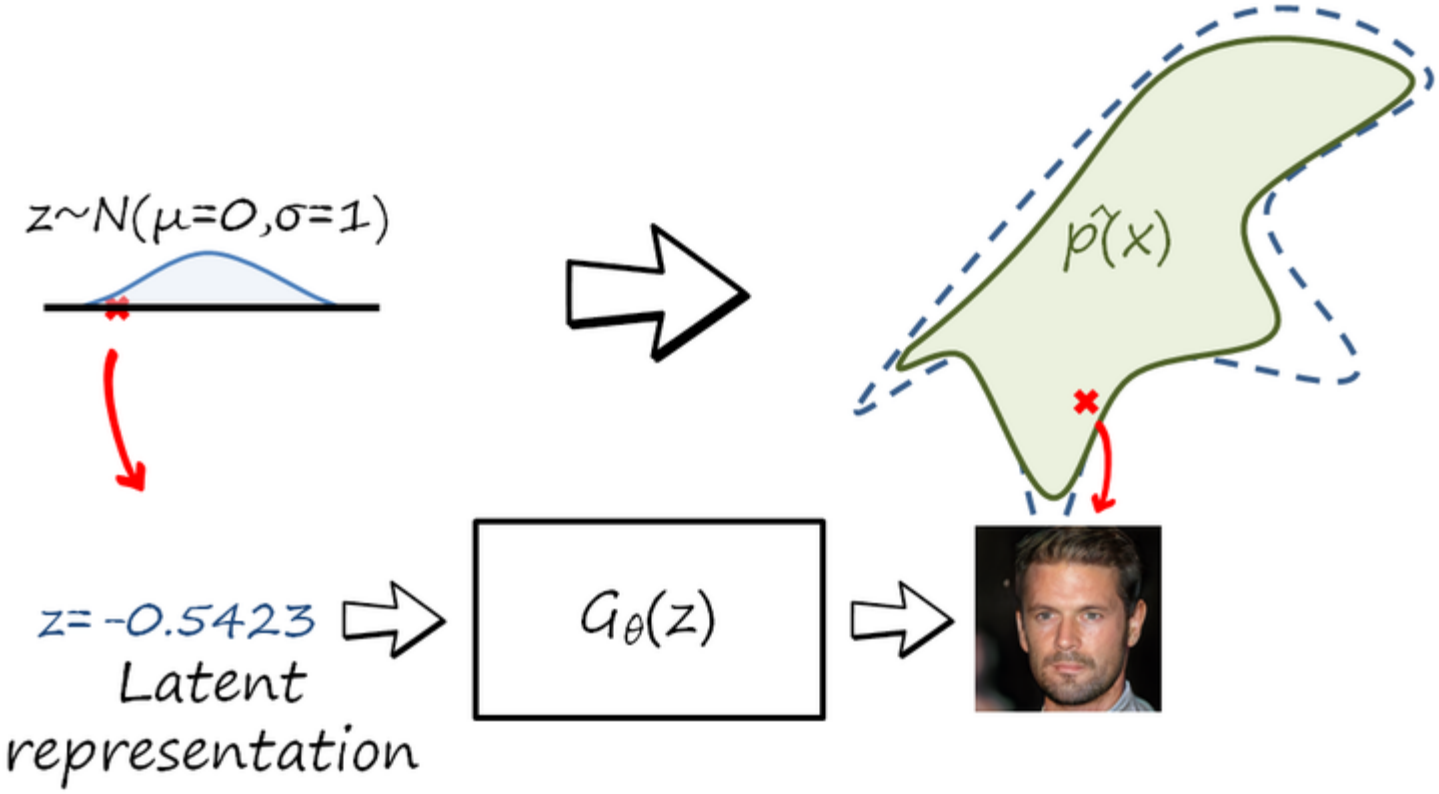
	Discriminative model	Generative model
Goal	Directly estimate $P(y\|x)$	Estimate $P(x\|y)$ to then deduce $P(y\|x)$
What's learned	Decision boundary	Probability distributions of the data
Illustration		
Examples	Regressions, SVMs	GDA, Naive Bayes

生成模型：一系列用于随机生成可观测数据的模型

- 生成模型包含两个步骤：
  - 密度估计
  - 采样



生成数据的另一种思路



显式密度模型和隐式密度模型

- 显式密度模型
  - 显式地构建出样本的密度函数$p(x|\theta)$，并通过最大似然估计来求解参数；
  - 变分自编码器、深度信念网络
- 隐式密度模型
  - 不显式地估计出数据分布的密度函数
  - 但能生成符合数据分布$p_{data}(x)$的样本
  - 无法用最大似然估计

深度生成模型

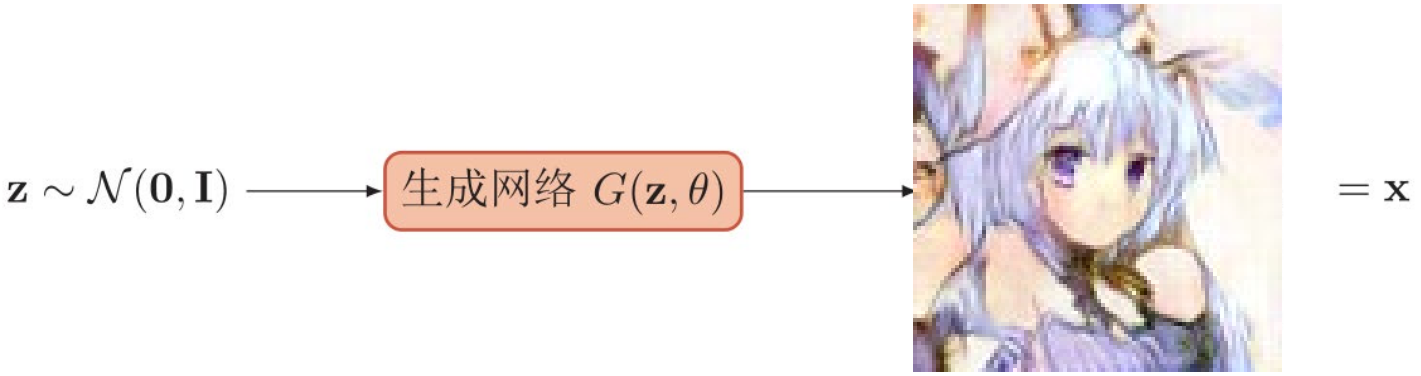
- 深度生成模型就是利用神经网络构建的生成模型。

  - 生成对抗网络（Generative Adversarial Network，GAN）
    - [Goodfellow et al., 2014]
  - 变分自编码器（Variational Autoencoder，VAE）
    - [Kingma and Welling, 2013, Rezende et al., 2014]
  - 基于流的方法（Flow-based Method）
  - 扩散模型（Diffusion Model）

生成对抗网络
Generative Adversarial Network（GAN）

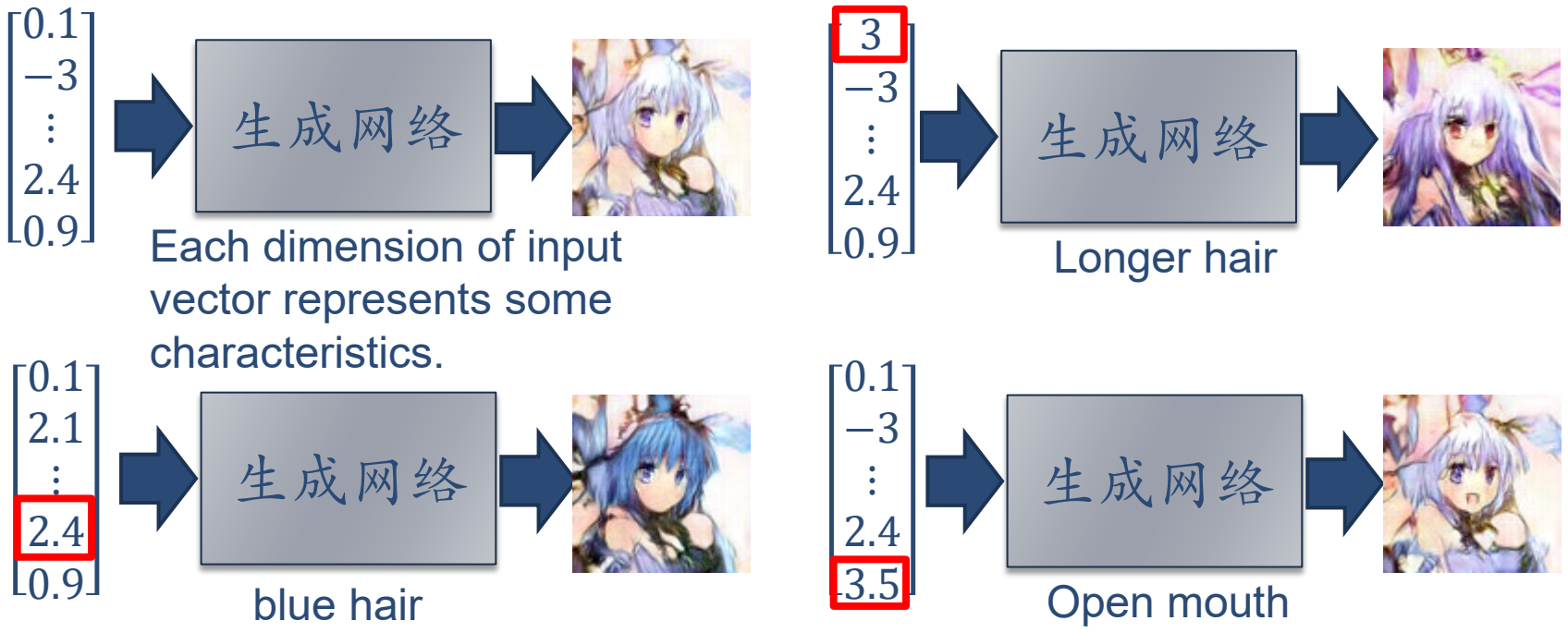
生成网络

- 生成网络从隐空间（latent space）中随机采样作为输入，其输出结果需要尽量模仿训练集中的真实样本。



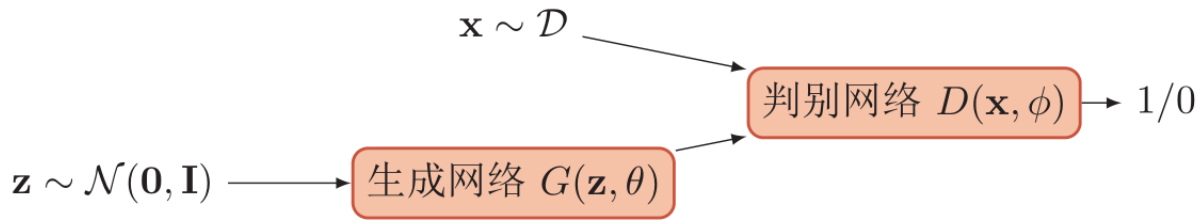
如何学习生成网络?

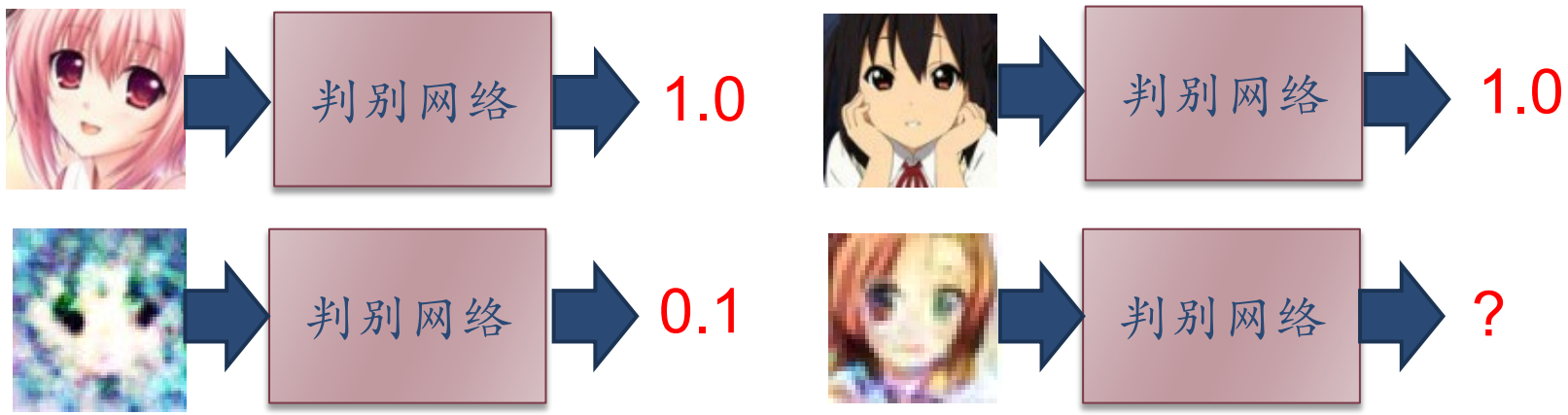
生成网络示例


0.1
−3
2.4
0.9
生成网络
Each dimension of input vector represents some characteristics.
3
−3
2.4
0.9
生成网络
Longer hair
0.1
2.1
2.4
0.9
生成网络
blue hair
0.1
−3
2.4
3.5
生成网络
Open mouth

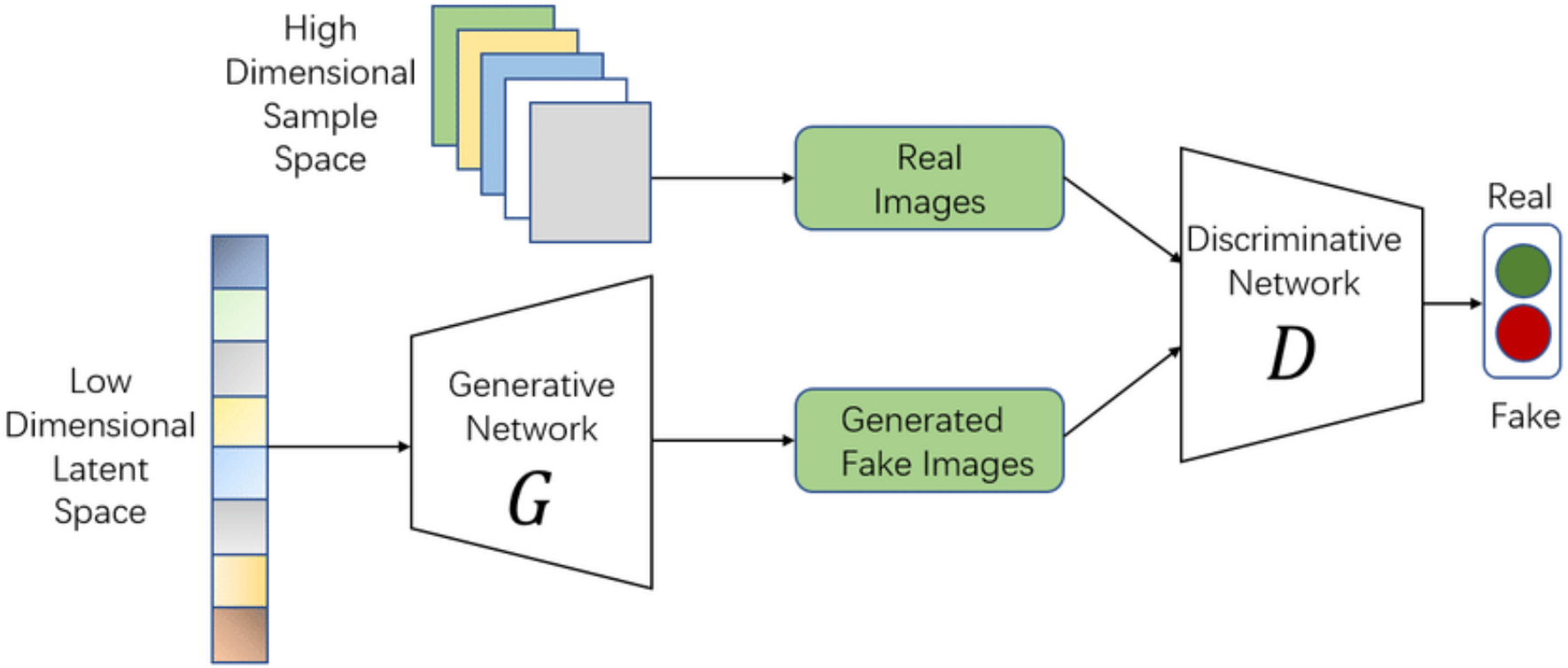
判别网络

- 判别网络的输入则为真实样本或生成网络的输出，其目的是将生成网络的输出从真实样本中尽可能分辨出来。





生成对抗网络

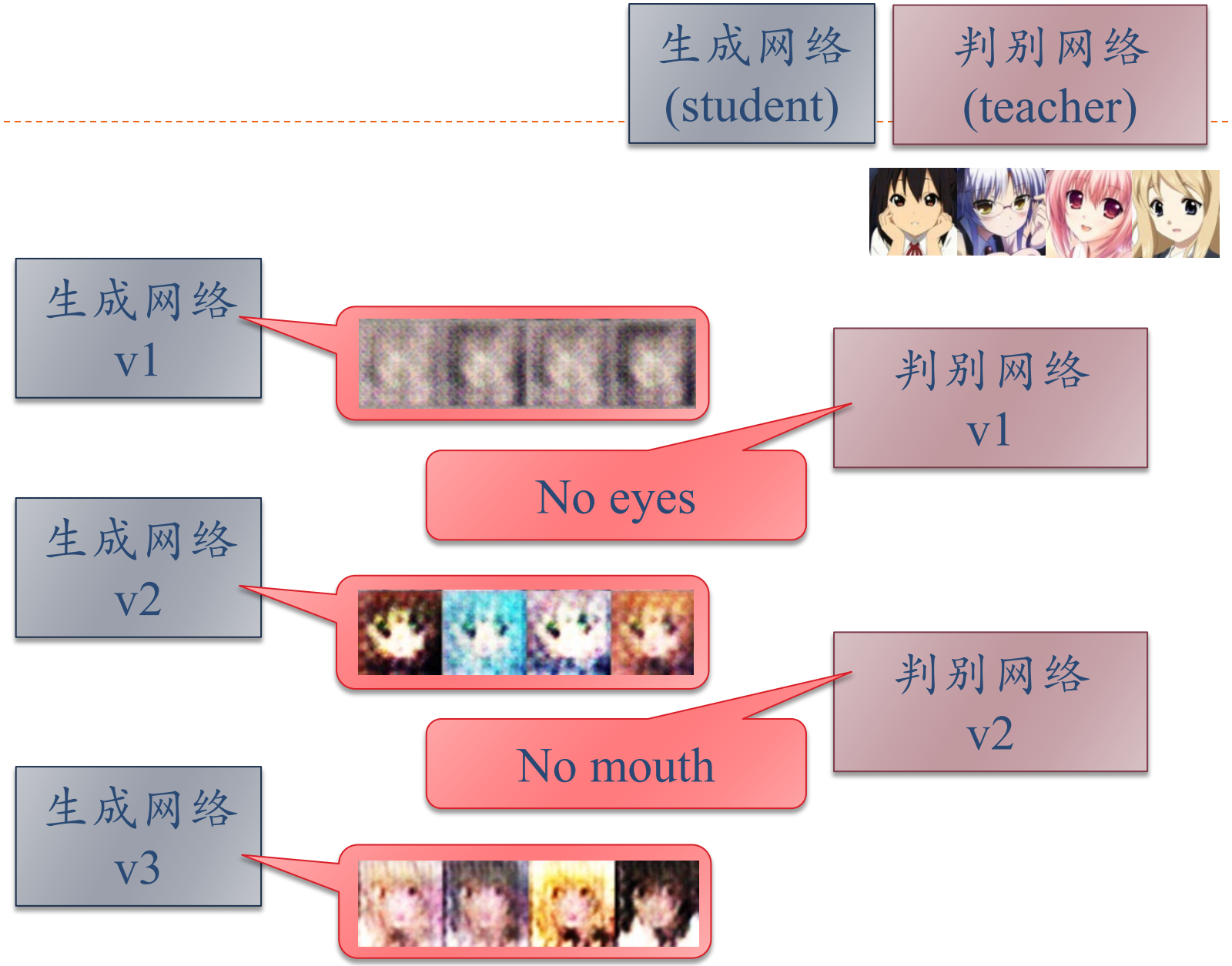

High
Dimensional
Sample
Space
Real
Images
Low
Dimensional
Latent
Space
Generative
Network
G
Generated
Fake Images
Discriminative
Network
D
Real
Fake

MiniMax Game

- 对抗训练
  - 生成网络要尽可能地欺骗判别网络。
  - 判别网络将生成网络生成的样本与真实样本中尽可能区分出来。

- 两个网络相互对抗、不断调整参数，最终目的是使判别网络无法判断生成网络的输出结果是否真实。

对抗过程


生成网络
(student)
判别网络
(teacher)
生成网络
v1
判别网络
v1
No eyes
生成网络
v2
判别网络
v2
No mouth
生成网络
v3

MiniMax Game

- 判别网络

$$\max_{\phi} \mathbb{E}_{\boldsymbol{x}\sim p_r(\boldsymbol{x})}\Big[\log D(\boldsymbol{x};\phi)\Big] + \mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z})}\Big[\log(1 - D\Big(G(\boldsymbol{z};\theta);\phi\Big)\Big]$$

- 生成网络

$$\max_{\theta}\left(\mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z})}\Big[\log D\Big(G(\boldsymbol{z};\theta);\phi\Big)\Big]\right)$$

- Minimax Game

$$\min_{\theta}\max_{\phi}\left(\mathbb{E}_{\boldsymbol{x}\sim p_r(\boldsymbol{x})}\Big[\log D(\boldsymbol{x};\phi)\Big] + \mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z})}\Big[\log(1 - D\Big(G(\boldsymbol{z};\theta);\phi\Big)\Big]\right)$$

训练过程

算法 13.1 生成对抗网络的训练过程

输入: 训练集 $\mathcal{D}$，对抗训练迭代次数 T，每次判别网络的训练迭代次数 K，小批量样本数量 M

随机初始化 θ, ϕ;

for $t \leftarrow 1$ **to** T **do**

  // 训练判别网络 $D(\boldsymbol{x}; \phi)$

  for $k \leftarrow 1$ **to** K **do**

    // 采集小批量训练样本

    从训练集 $\mathcal{D}$ 中采集 M 个样本 $\{\boldsymbol{x}^{(m)}\}, 1 \leq m \leq M$;

    从分布 $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$ 中采集 M 个样本 $\{\boldsymbol{z}^{(m)}\}, 1 \leq m \leq M$;

    使用随机梯度上升更新 ϕ，梯度为

$$\frac{\partial}{\partial \phi}\left[\frac{1}{M}\sum_{m=1}^{M}\left(\log D(\boldsymbol{x}^{(m)}; \phi) + \log\left(1 - D(G(\boldsymbol{z}^{(m)}; \theta); \phi)\right)\right)\right];$$

  end

  // 训练生成网络 $G(\boldsymbol{z}; \theta)$

  从分布 $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$ 中采集 M 个样本 $\{\boldsymbol{z}^{(m)}\}, 1 \leq m \leq M$;

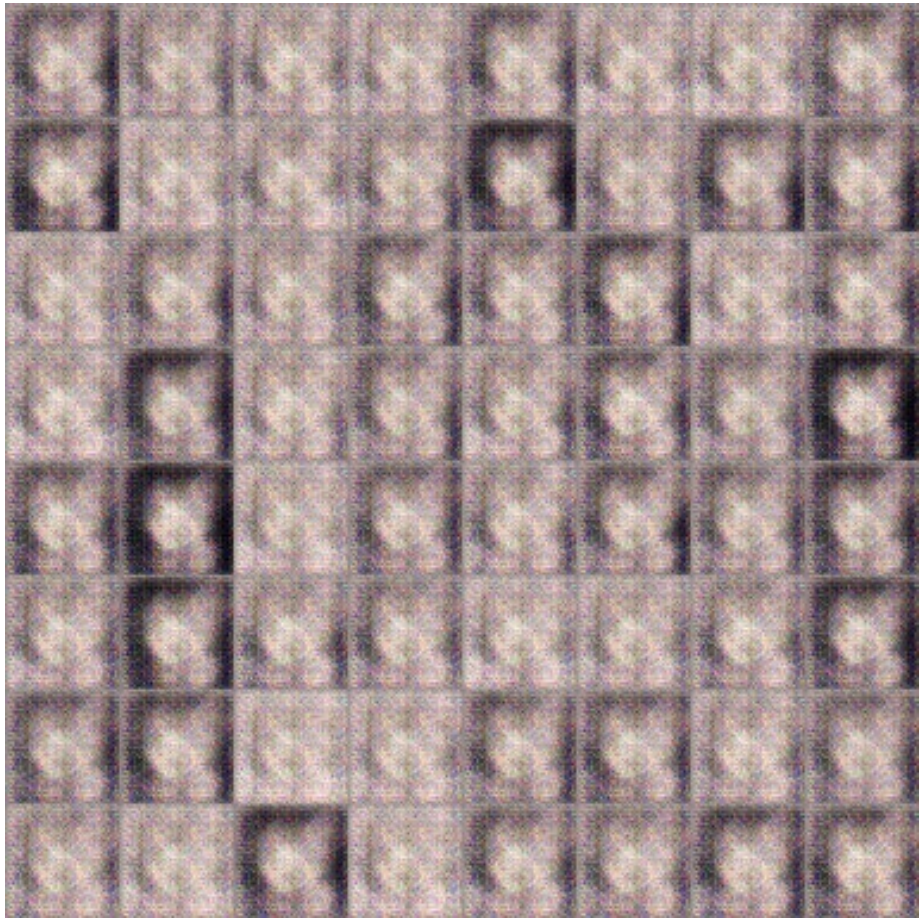
  使用随机梯度上升更新 θ，梯度为

$$\frac{\partial}{\partial \theta}\left[\frac{1}{M}\sum_{m=1}^{M} D(G(\boldsymbol{z}^{(m)}; \theta), \phi)\right];$$

end

输出: 生成网络 $G(\boldsymbol{z}; \theta)$

Anime Face Generation



100 updates

1000 updates

Anime Face Generation

2000 updates

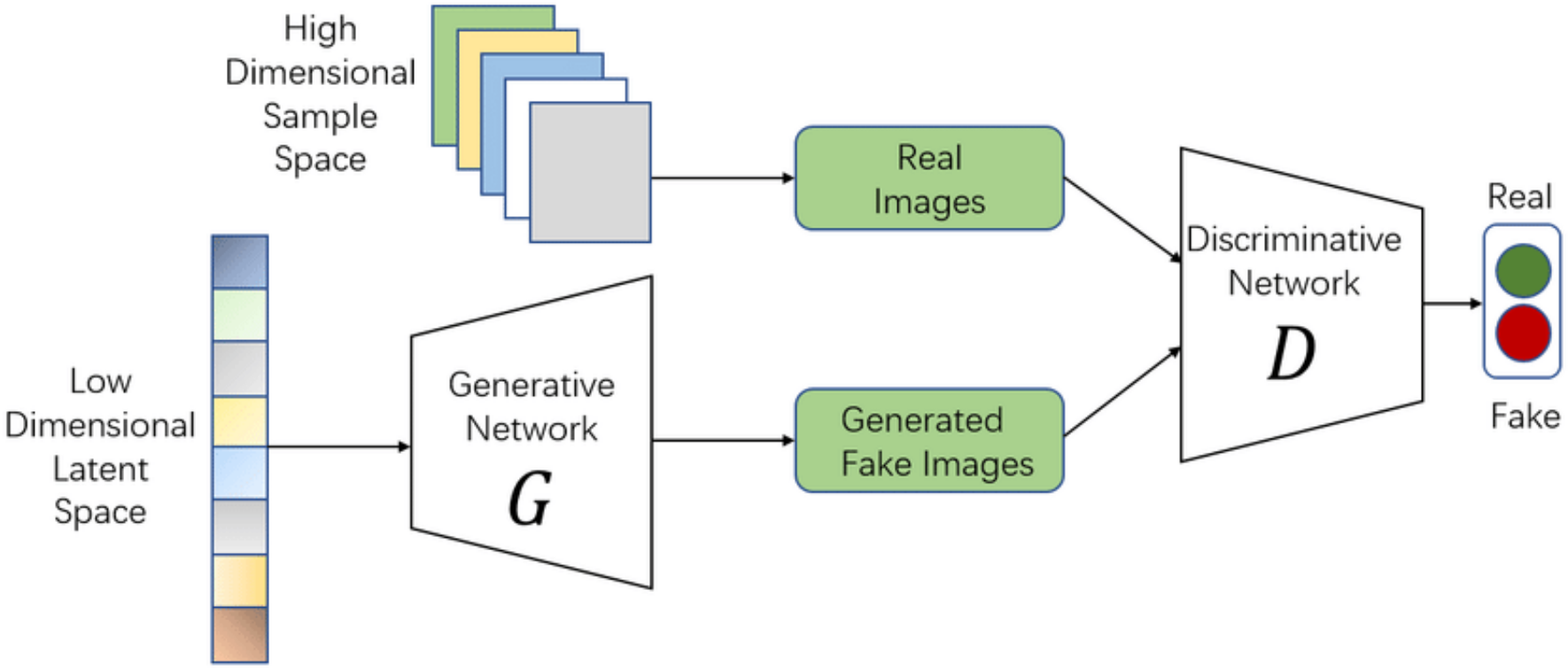
5000 updates

Anime Face Generation

10,000 updates

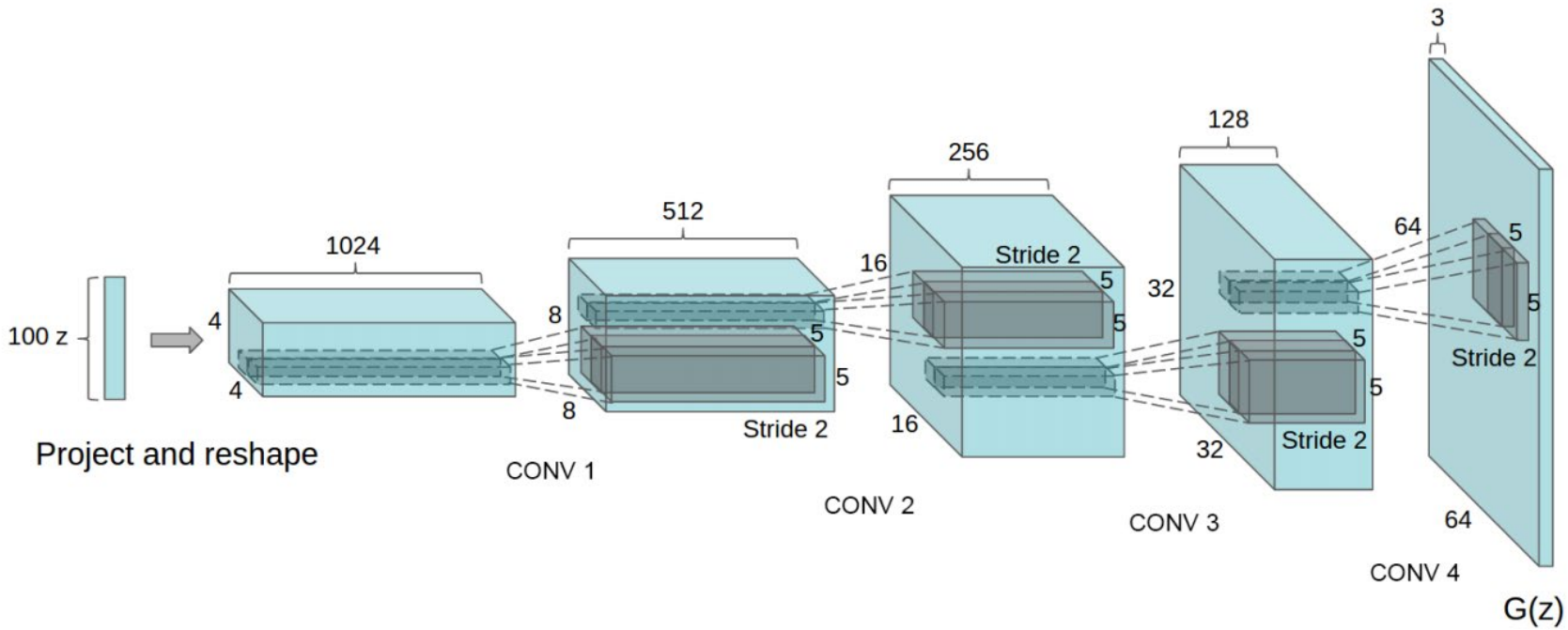
50,000 updates

生成对抗网络


High
Dimensional
Sample
Space
Real
Images
Discriminative
Network
D
Real
Low
Dimensional
Latent
Space
Generative
Network
G
Generated
Fake Images
Fake

一个具体的模型：DCGANs

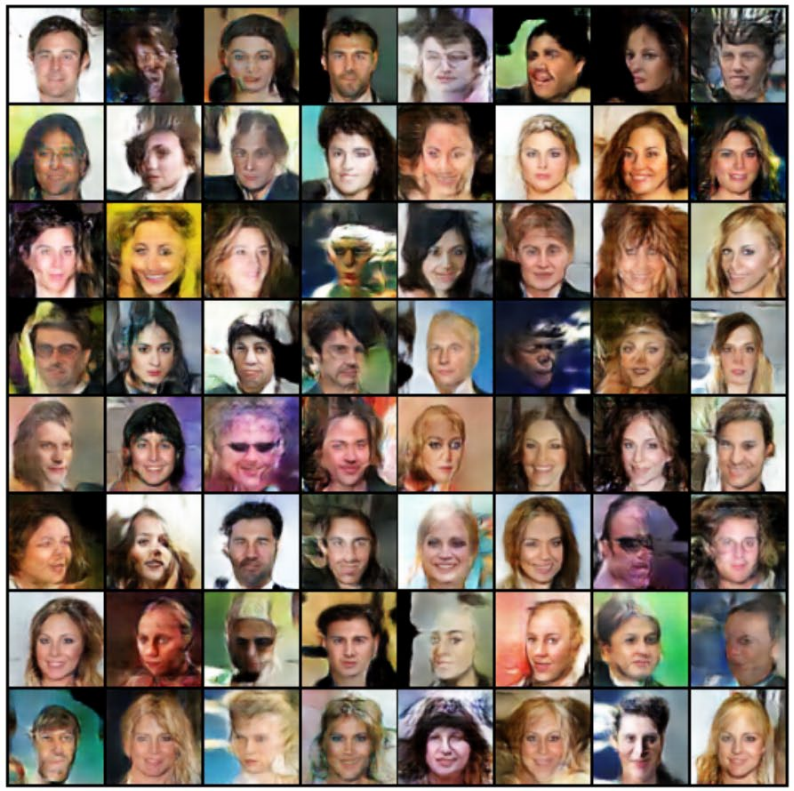
- DCGAN (Deep Convolutional Generative Adversarial Networks)
- 判别网络是一个传统的深度卷积网络，但使用了带步长的卷积来实现下采样操作，不用最大汇聚（pooling）操作。
- 生成网络使用一个特殊的深度卷积网络来实现使用微步卷积来生成64×63大小的图像。



DCGANs

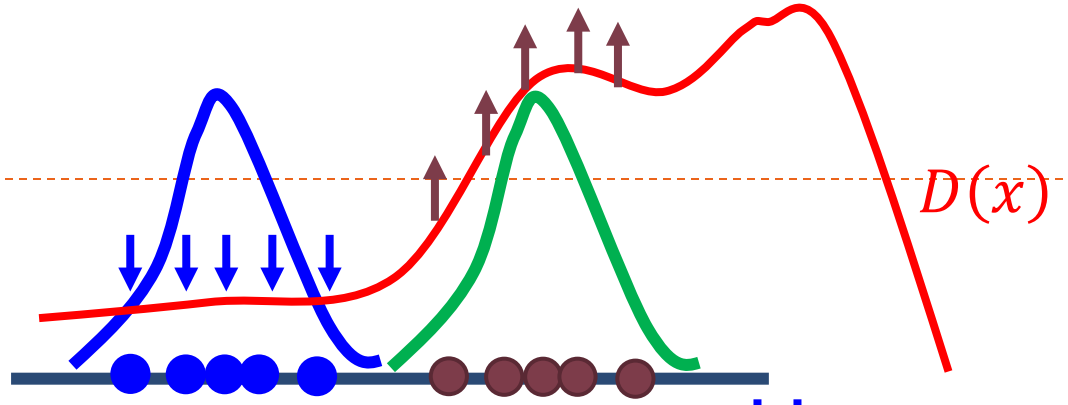
Real Images

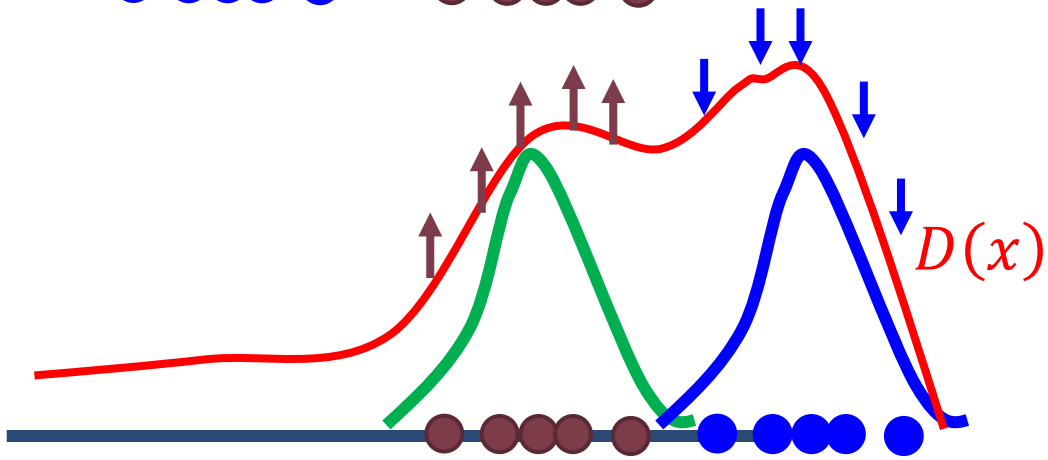
Fake Images



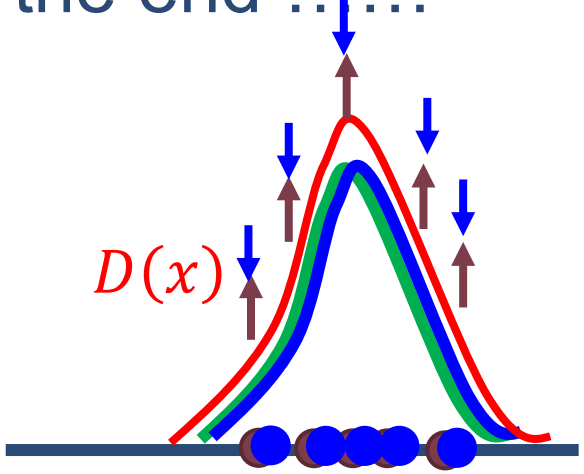
模型分析

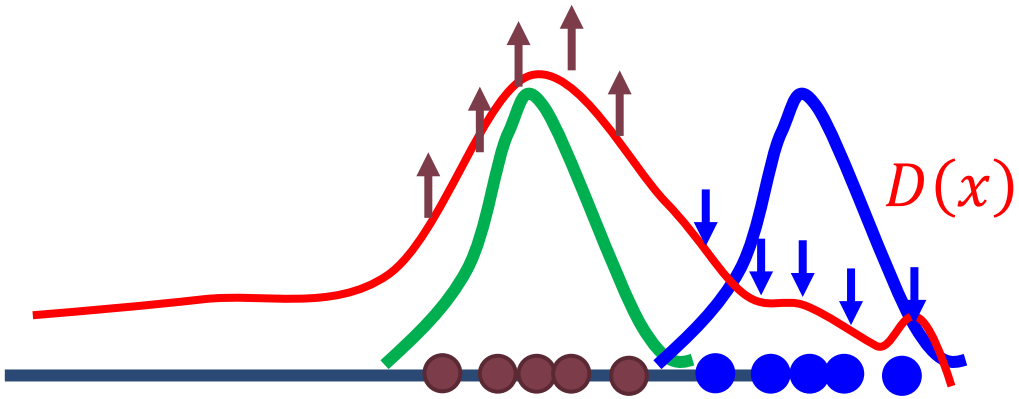
数据分布





In the end ……





模型分析

- 假设$p_r(x)$和$p_\theta(x)$已知，则最优的判别器为

$$D^\star(\boldsymbol{x}) = \frac{p_r(\boldsymbol{x})}{p_r(\boldsymbol{x}) + p_\theta(\boldsymbol{x})}$$

- 目标函数变为

$$\begin{aligned}
\mathcal{L}(G|D^\star) &= \mathbb{E}_{\boldsymbol{x}\sim p_r(\boldsymbol{x})}\Big[\log D^\star(\boldsymbol{x})\Big] + \mathbb{E}_{\boldsymbol{x}\sim p_\theta(\boldsymbol{x})}\Big[\log(1 - D^\star(\boldsymbol{x}))\Big] \\
&= \mathbb{E}_{\boldsymbol{x}\sim p_r(\boldsymbol{x})}\left[\log \frac{p_r(\boldsymbol{x})}{p_r(\boldsymbol{x}) + p_\theta(\boldsymbol{x})}\right] + \mathbb{E}_{\boldsymbol{x}\sim p_\theta(\boldsymbol{x})}\left[\log \frac{p_\theta(\boldsymbol{x})}{p_r(\boldsymbol{x}) + p_\theta(\boldsymbol{x})}\right] \\
&= D_{\mathrm{KL}}\Big(p_r \| p_a\Big) + D_{\mathrm{KL}}\Big(p_\theta \| p_a\Big) - 2\log 2 \\
&= 2D_{\mathrm{JS}}(p_r \| p_\theta) - 2\log 2,
\end{aligned}$$

$$D_{\mathrm{JS}}(p\|q) = \frac{1}{2}D_{\mathrm{KL}}(p\|m) + \frac{1}{2}D_{KL}(q\|m)$$

$$m = \frac{1}{2}(p+q)$$

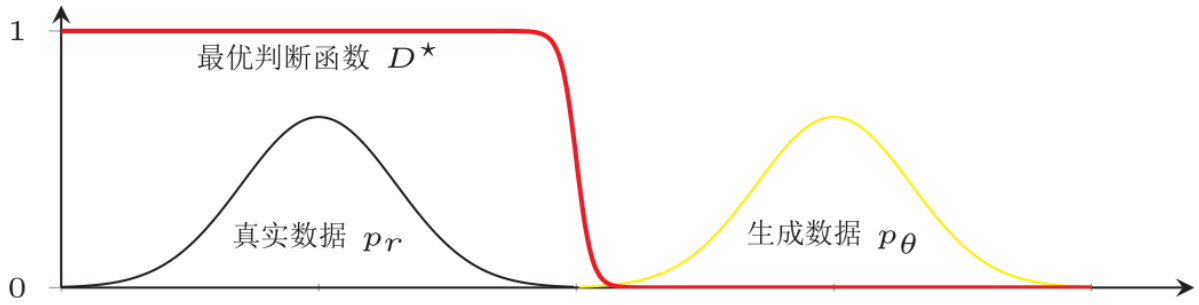
不稳定性：生成网络的梯度消失

$$\min_{\theta}\max_{\phi}\left(\mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})}\left[\log D(\mathbf{x},\phi)\right]+\mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}\left[\log(1-D\Big(G(\mathbf{z},\theta),\phi\Big)\right]\right)$$

在生成对抗网络中，当判断网络为最优时，生成网络的优化目标是最小化真实分布$p_r(x)$和模型分布$p_\theta(x)$之间的JS散度。
当两个分布相同时，JS散度为0，最优生成网络对应的损失为−2log2。

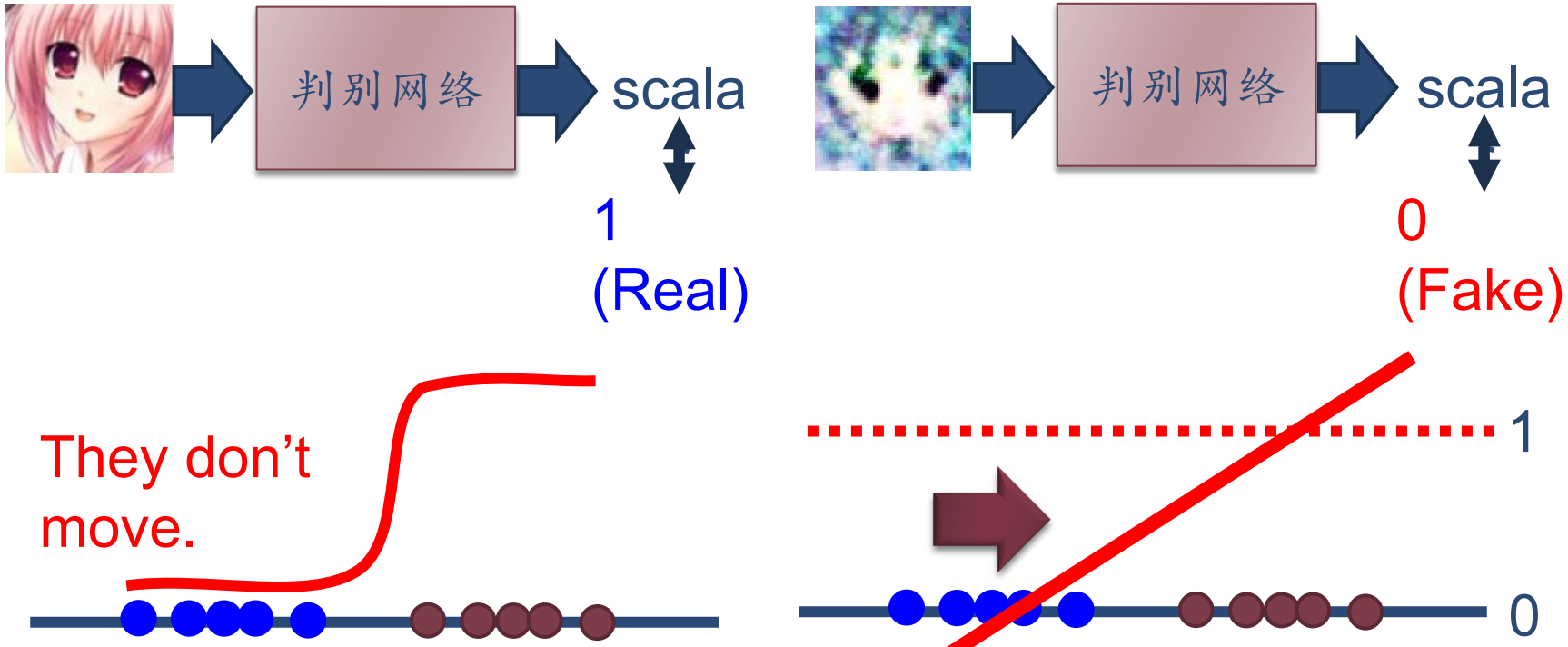
$$\frac{\partial\mathcal{L}(G|D^\star)}{\partial\theta}=0.$$

使用JS散度来训练生成对抗网络的一个问题是当两个分布没有重叠时，它们之间的JS散度恒等于常数log2。对生成网络来说，目标函数关于参数的梯度为0。



Least Square GAN (LSGAN)

- Replace sigmoid with linear (replace classification with regression)



Wasserstein GAN

Wasserstein距离

- Wasserstein距离用于衡量两个分布之间的距离。

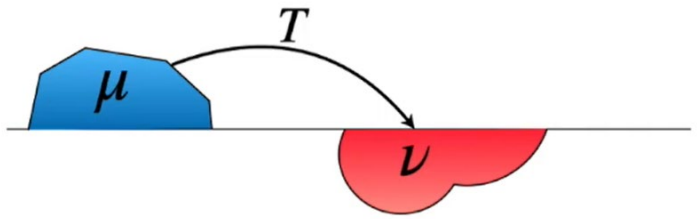
$$W_p(q_1, q_2) = \left(\inf_{\gamma(x,y) \in \Gamma(q_1, q_2)} \mathbb{E}_{(x,y)\sim\gamma(x,y)}[d(x,y)^p] \right)^{\frac{1}{p}}$$

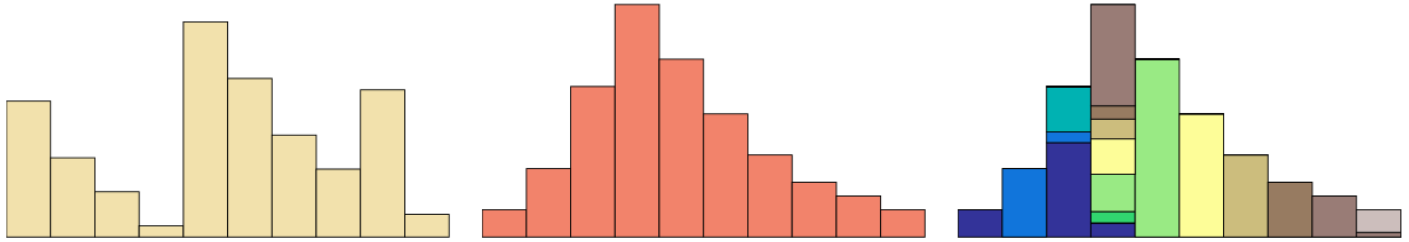
  - 其中$\Gamma(q_1, q_2)$是边际分布为q_1, q_2的所有可能的联合分布集合，d(x,y)为x和y的距离，比如ℓ_p 距离等。

- Wasserstein距离相比KL散度和JS散度的优势在于：即使两个分布没有重叠或者重叠非常少，Wasserstein距离仍然能反映两个分布的远近。

Wasserstein距离

- 如果将两个分布看作是两个土堆，联合分布γ(x,y)看作是从土堆q_1的位置x到土堆q_2的位置y的搬运土的数量。Wasserstein距离可以理解为搬运土堆的最小工作量，也称为推土机距离（Earth-Mover's Distance，EMD）。



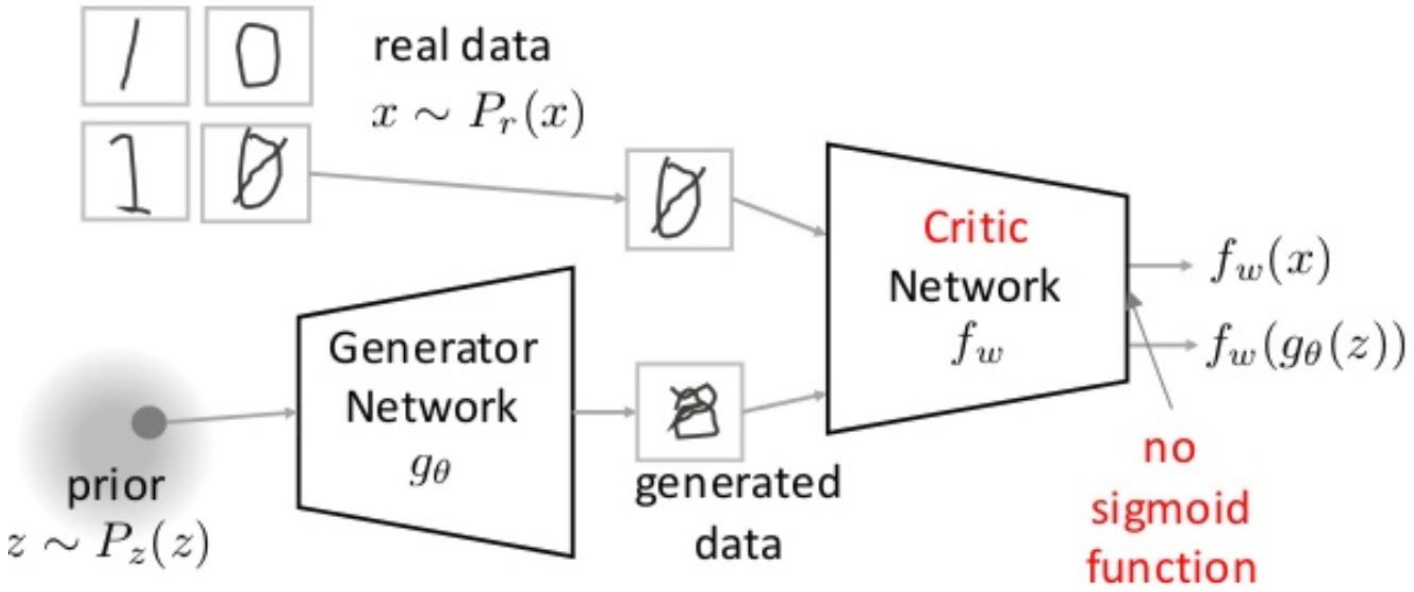


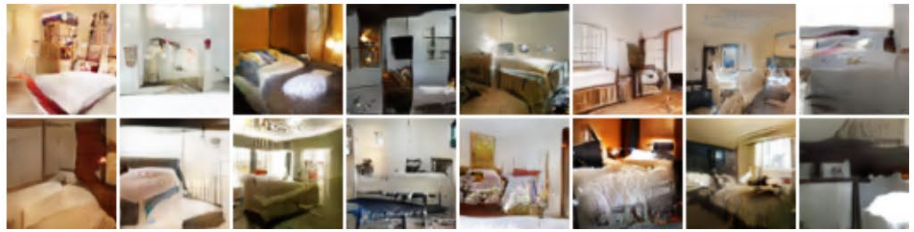
(a) $q_1(x)$　(b) $q_2(x)$　(c) q_1 到 q_2 最优的运输方案

Wasserstein GAN

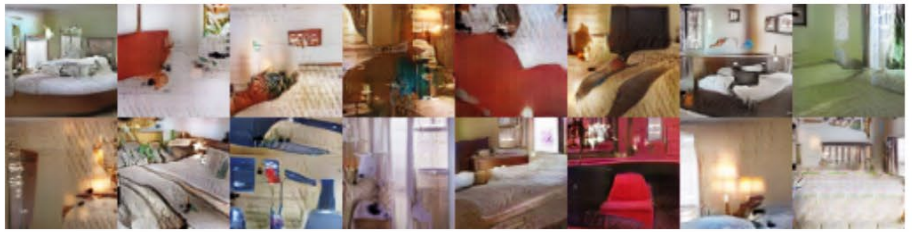
$$\min_{\theta} \max_{w \in [-k,k]^l} \mathbb{E}_{x \sim P_r}[f_w(x)] - \mathbb{E}_{z \sim P_z}[f_w(g_\theta(z))]$$

k-Lipschitz Constraint



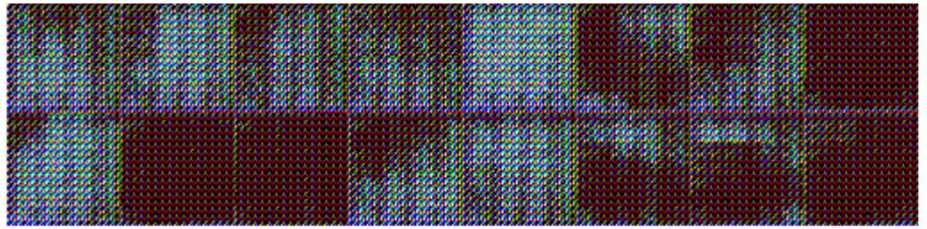


WGAN



DCGAN

batch normalization
constant number of filters at every layer

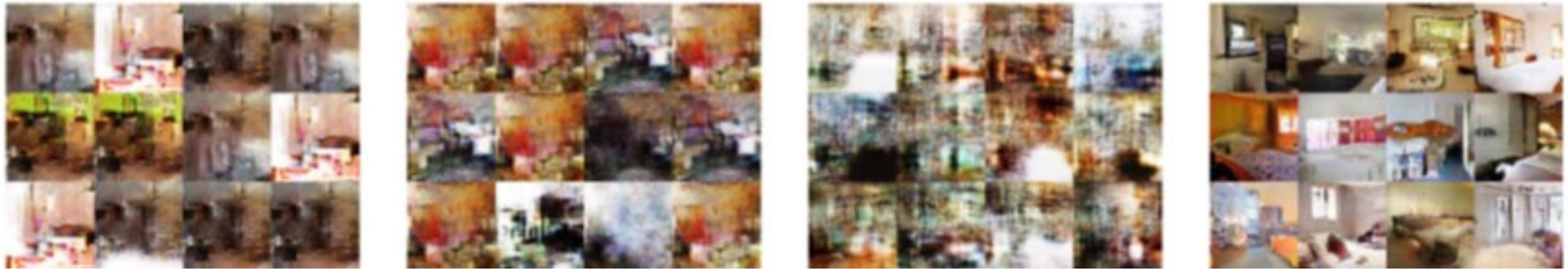


DCGAN | LSGAN | Original WGAN | Improved WGAN

G: CNN, D: CNN

G: CNN (no normalization), D: CNN (no normalization)

G: CNN (tanh), D: CNN(tanh)

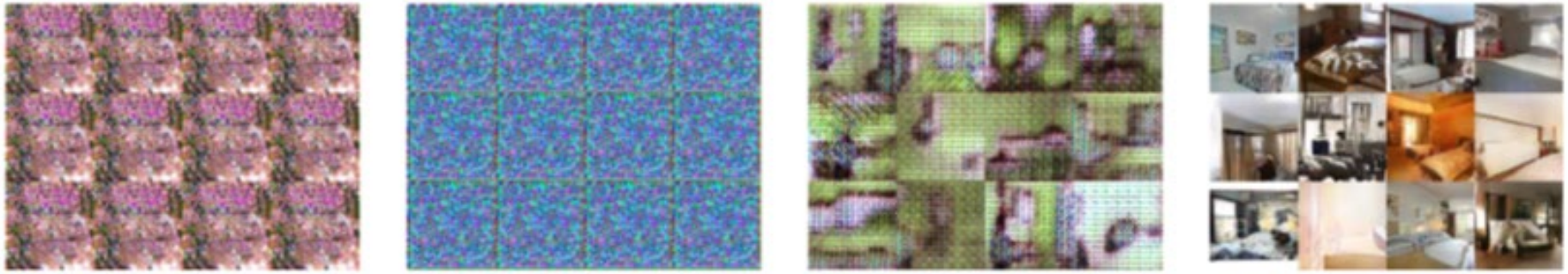


DCGAN | LSGAN | Original WGAN | Improved WGAN

G: MLP, D: CNN

G: CNN (bad structure), D: CNN

G: 101 layer, D: 101 layer



GAN的扩展

条件生成

- 根据条件针对性的生成数据

$$\begin{bmatrix} 0.3 \\ -0.1 \\ \vdots \\ -0.7 \end{bmatrix} \begin{bmatrix} 0.1 \\ -0.1 \\ \vdots \\ 0.7 \end{bmatrix} \begin{bmatrix} -0.3 \\ 0.1 \\ \vdots \\ 0.9 \end{bmatrix}$$

In a specific range

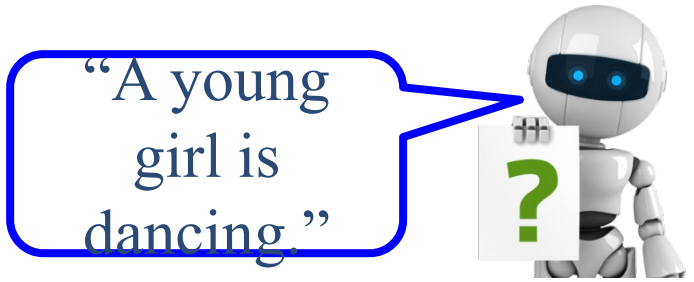
“Girl with red hair and red eyes”

“Girl with yellow ribbon”

条件生成

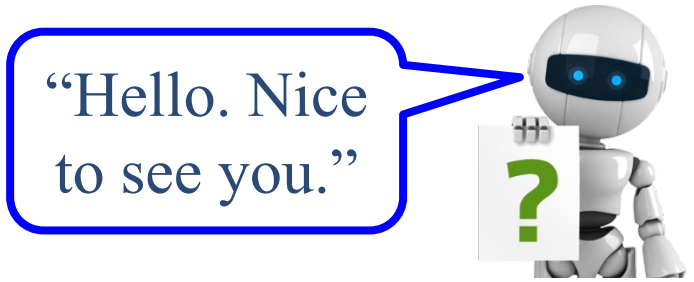
Caption Generation

Given condition:

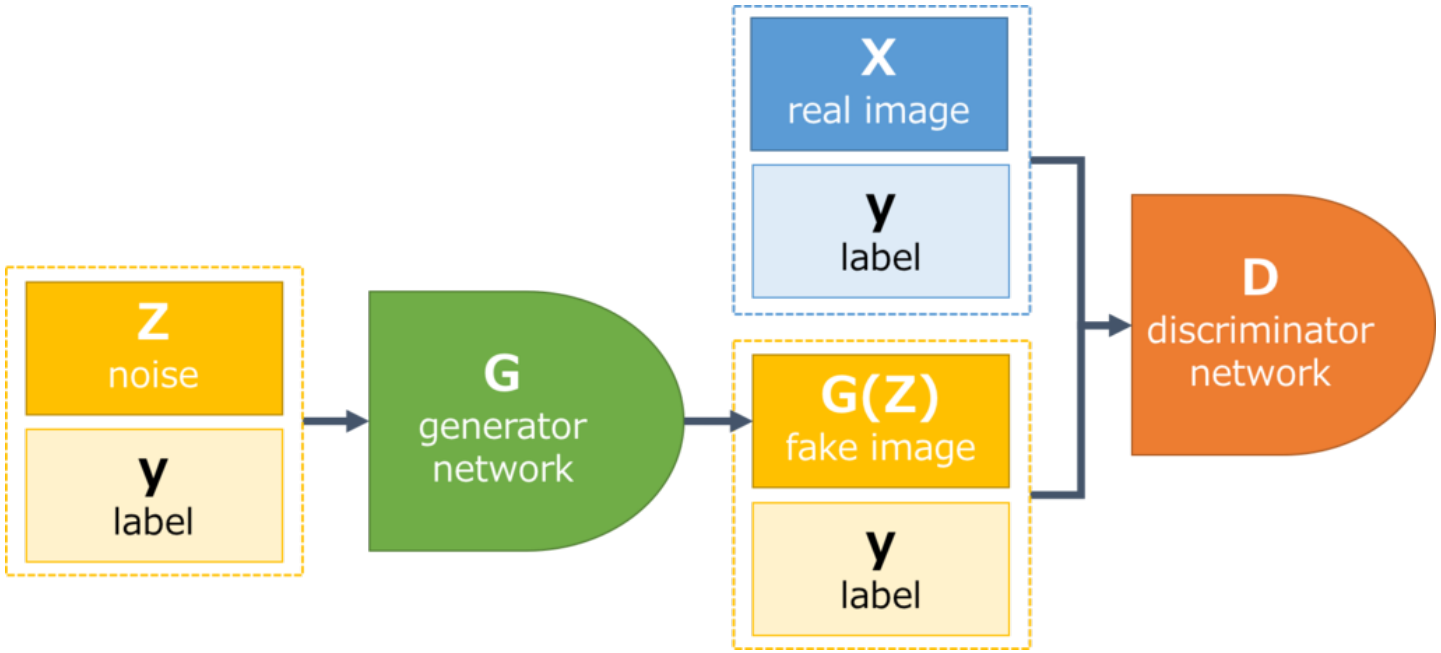


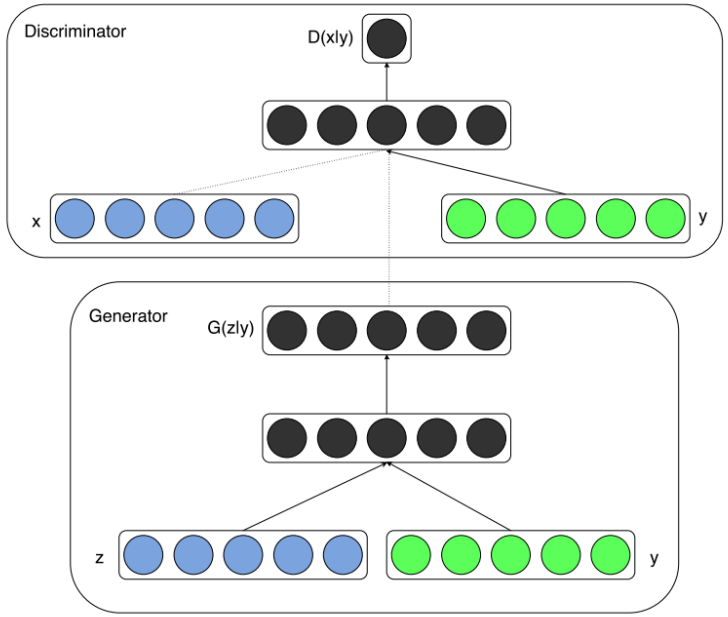
Chat-bot

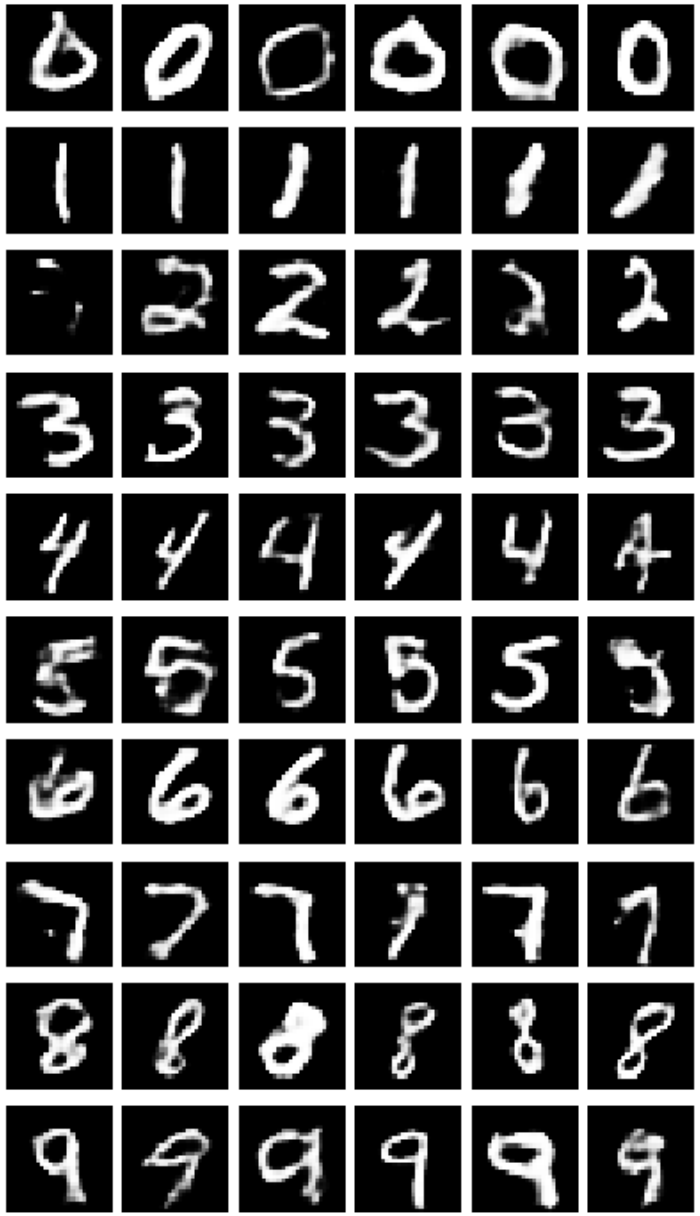
Given condition:

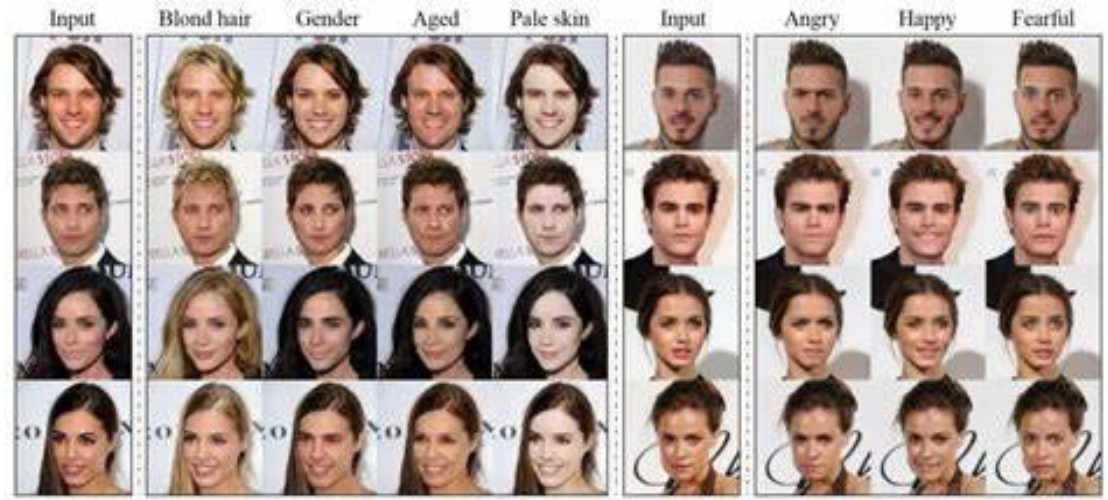


Conditional GAN

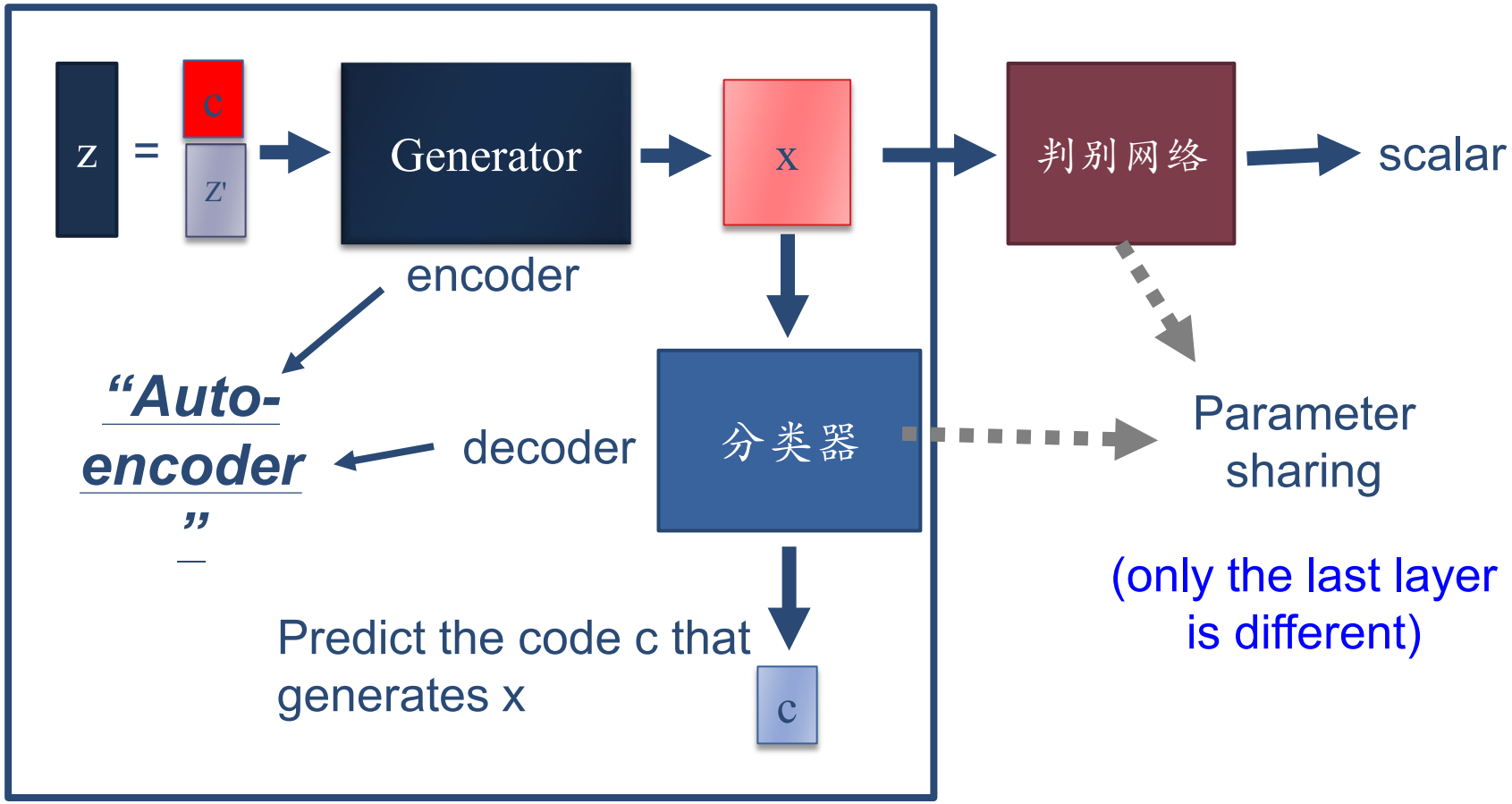


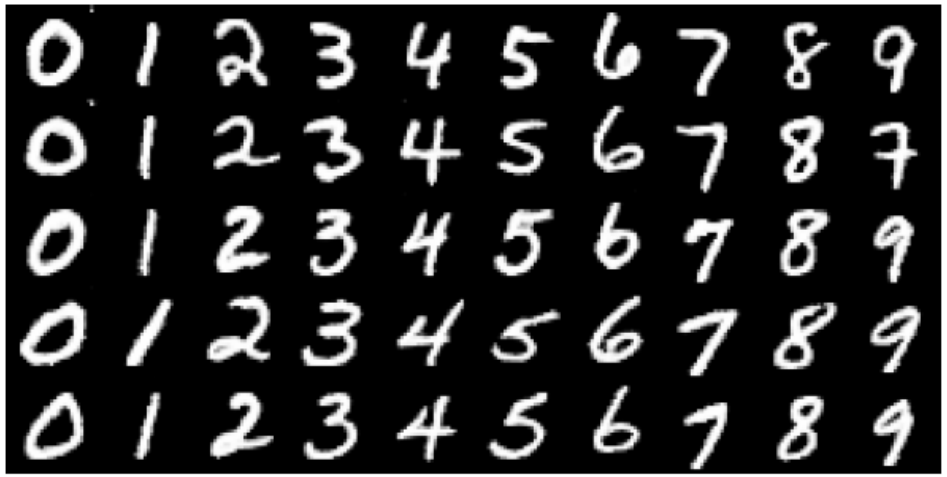




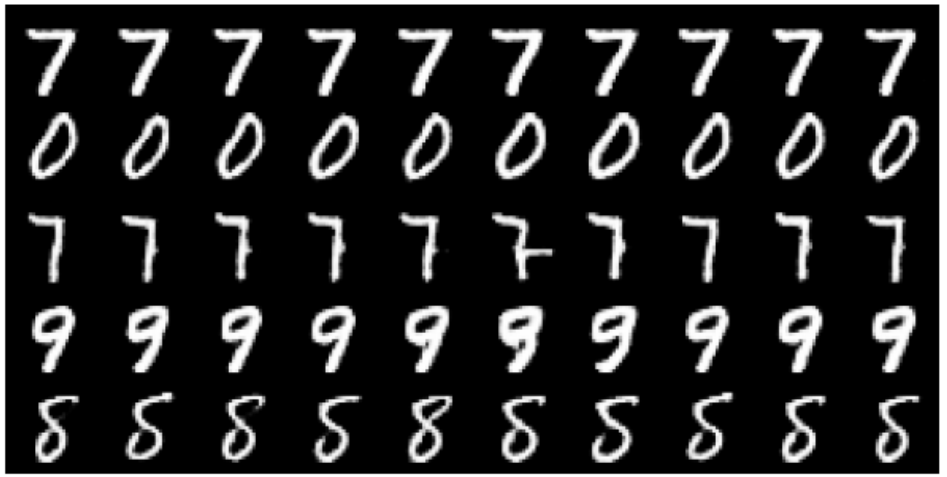

Input
Blond hair
Gender
Aged
Pale skin
Input
Angry
Happy
Fearful

InfoGAN

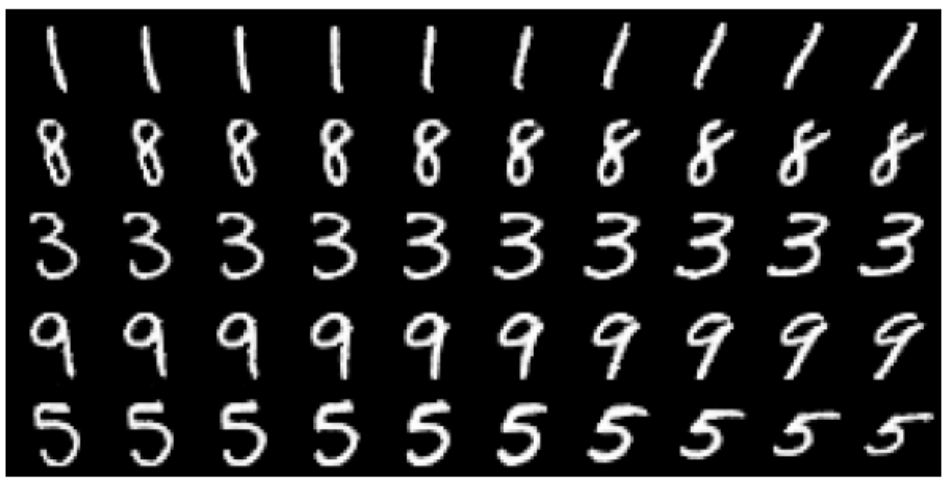

z
=
c
z'
Generator
x
判别网络
scalar
encoder
"Auto-encoder"
decoder
分类器
Parameter sharing
(only the last layer is different)
Predict the code c that generates x
c



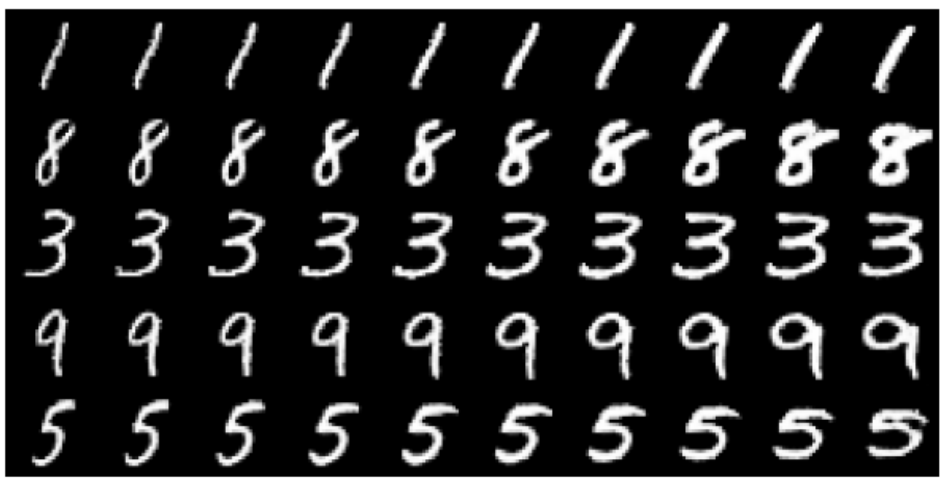
(a) Varying c_1 on InfoGAN (Digit type)



(b) Varying c_1 on regular GAN (No clear meaning)



(c) Varying c_2 from -2 to 2 on InfoGAN (Rotation)



(d) Varying c_3 from -2 to 2 on InfoGAN (Width)

https://arxiv.org/abs/1606.03657

(a) Rotation (b) Width

(c) Lighting (d) Wide or Narrow

https://arxiv.org/abs/1606.03657