

第7章：概率图模型和深度生成模型

中国科学技术大学
电子工程与信息科学系

主讲教师：李厚强 (lihq@ustc.edu.cn)
周文罡 (zhwg@ustc.edu.cn)
李 礼 (li11@ustc.edu.cn)
胡 洋 (eeeyhu@ustc.edu.cn)



生成式模型 (Generative Models)

- 生成式模型概论
- 自回归生成模型
- 变分自编码器
- 扩散模型
- 生成对抗网络
- 条件生成式模型

扩散模型 (Diffusion Models)

Imagen

by Google



Sprouts in the shape of text 'Imagen' coming out of a fairytale book.

Stable diffusion



DALL·E

by OpenAI



An astronaut riding a horse in photorealistic style.

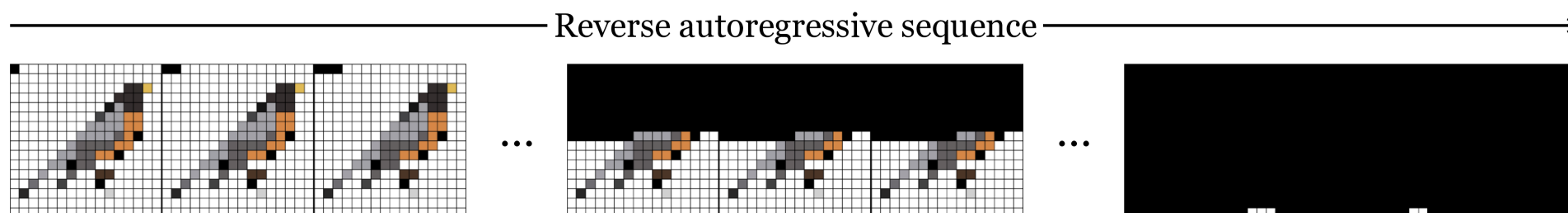
Midjourney



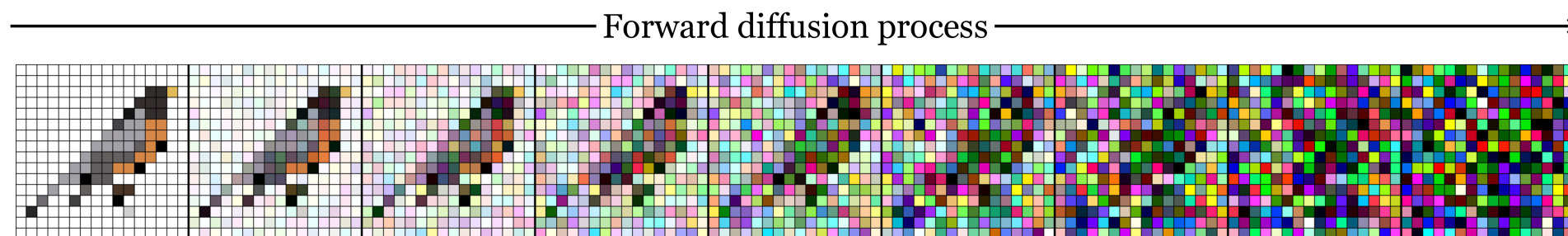
a cute fluffy bunny grumpily working on her trip itinerary.

扩散模型

□ 自回归生成的逆过程

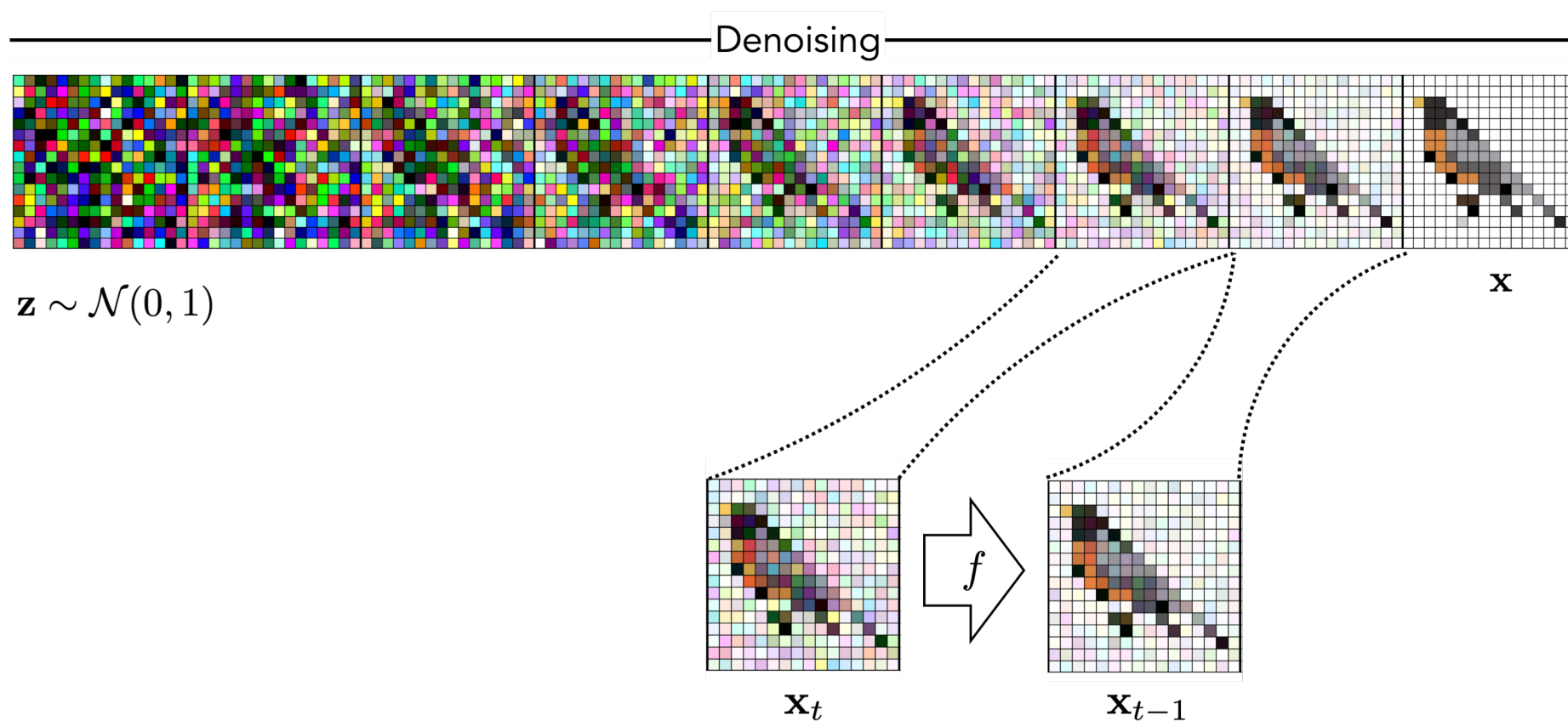


□ 前向扩散过程

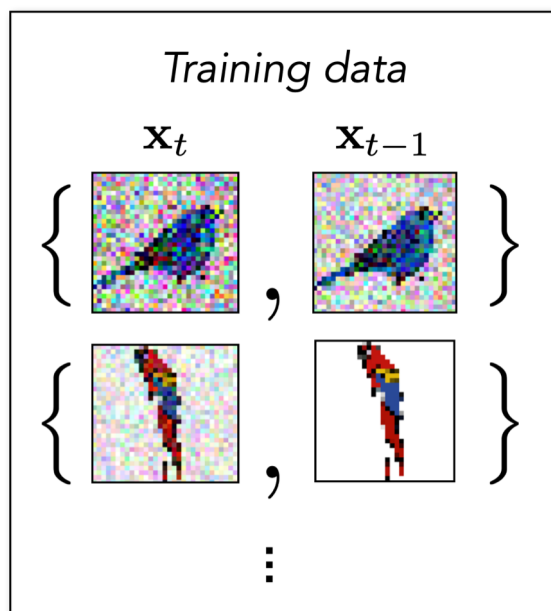
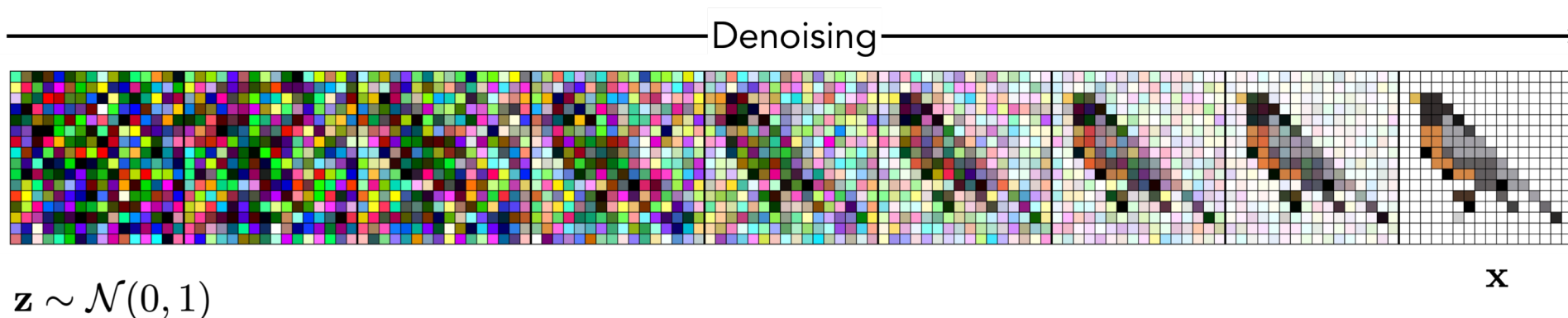


扩散模型

- 逆向去噪过程：通过逐步去噪实现生成



扩散模型

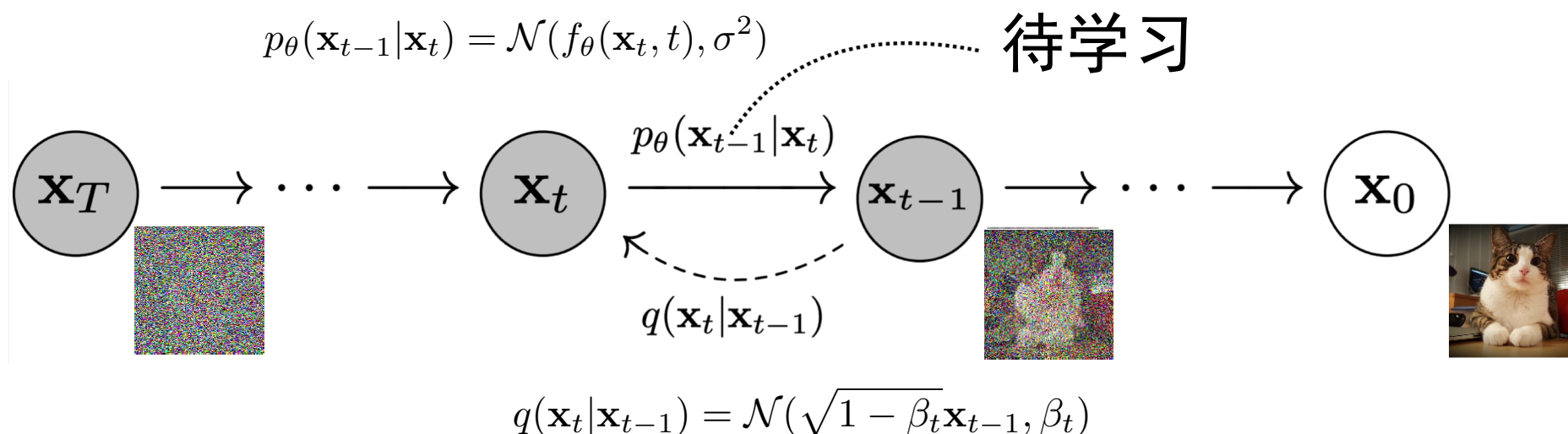


$$\arg \min_{f \in \mathcal{F}} \sum_{i=1}^N \mathcal{L}(f(\mathbf{x}_t), \mathbf{x}_{t-1})$$

将生成建模转化为监督预测问题

扩散模型

□ 马尔可夫链建模

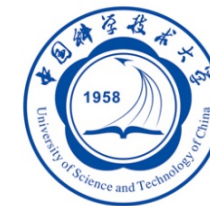


前向过程:

$$\epsilon_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \quad \mathbf{x}_t = \sqrt{(1 - \beta_t)}\mathbf{x}_{t-1} + \sqrt{\beta_t}\epsilon_t$$

逆向过程:

$$\mu = f_{\theta}(\mathbf{x}_t, t) \quad \mathbf{x}_{t-1} \sim \mathcal{N}(\mu, \sigma^2)$$



扩散模型

□ 训练过程

Algorithm 1.2: Training a diffusion model.

- 1 **Input:** training data $\{\mathbf{x}^{(i)}\}_{i=1}^N$
 - 2 **Output:** trained model f_θ
 - 3 **Generate training sequences via diffusion:**
 - 4 **for** $i = 1, \dots, N$ **do**
 - 5 **for** $t = 1, \dots, T$ **do**
 - 6 $\epsilon_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
 - 7 $\mathbf{x}_t^{(i)} \leftarrow \sqrt{(1 - \beta_t)} \mathbf{x}_{t-1}^{(i)} + \sqrt{\beta_t} \epsilon_t$
 - 8
 - 9 **Train denoiser f_θ to reverse these sequences:**
 - 10 $\theta^* = \arg \min_\theta \sum_{i=1}^N \sum_{t=1}^T \mathcal{L}(f_\theta(\mathbf{x}_t^{(i)}, t), \mathbf{x}_{t-1}^{(i)})$
 - 11 **Return:** f_{θ^*}
-

实际

$$\mathbf{x}_t \leftarrow \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon$$
$$\alpha_t = 1 - \beta_t, \quad \bar{\alpha}_t = \prod_{i=1}^t \alpha_i,$$

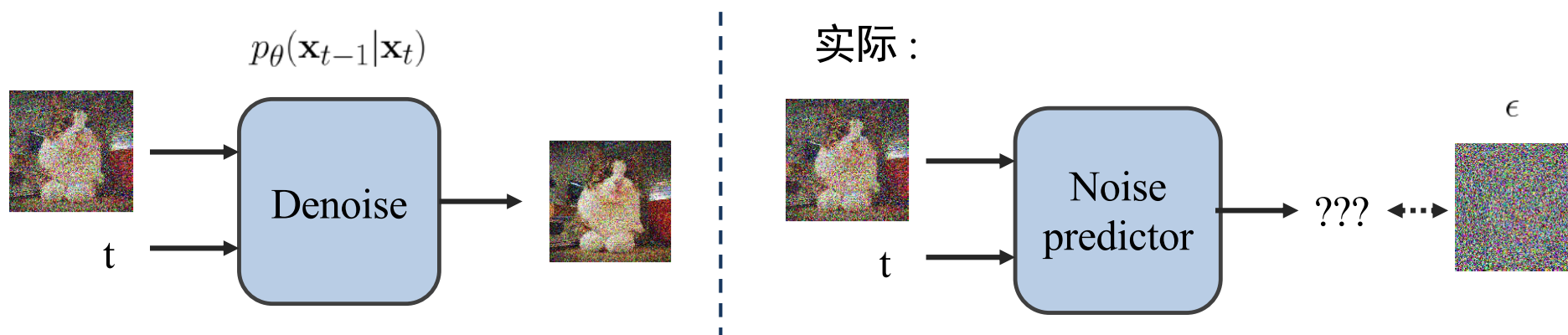
扩散模型

□ 去噪扩散概率模型 (DDPM)

$$L_{\text{simple}} = \mathbb{E}_{\mathbf{x}_0 \sim q(\mathbf{x}_0), \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), t \sim \mathcal{U}(1, T)} [\|\epsilon - \underbrace{\epsilon_{\theta}(\sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, t)}_{\mathbf{x}_t}\|^2]$$

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

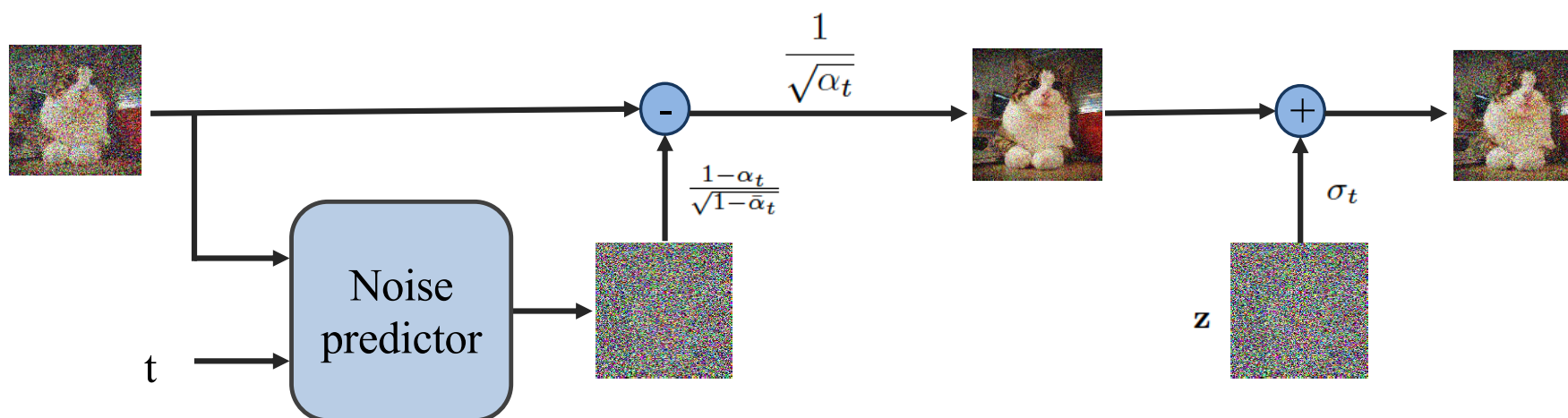


扩散模型

□ 去噪扩散概率模型 (DDPM)

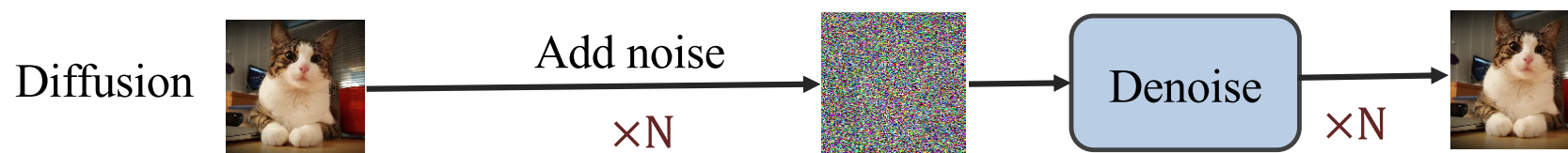
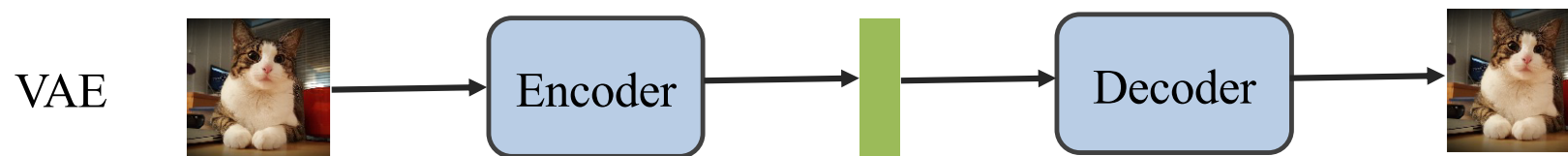
Algorithm 2 Sampling

```
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  $\mathbf{x}_0$ 
```



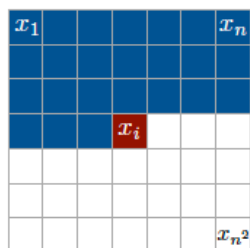
扩散模型

□ 扩散模型 vs. VAE

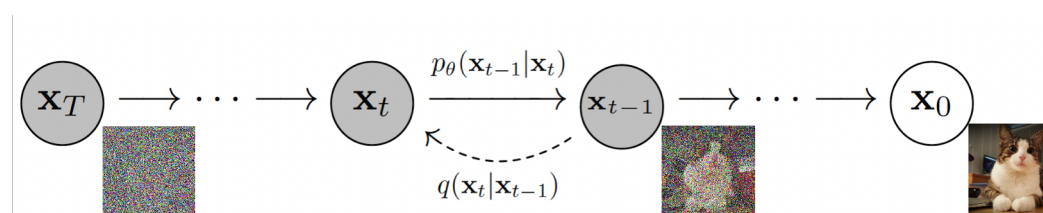


□ 扩散模型 vs. 自回归生成模型

PixelRNN/PixelCNN



Diffusion



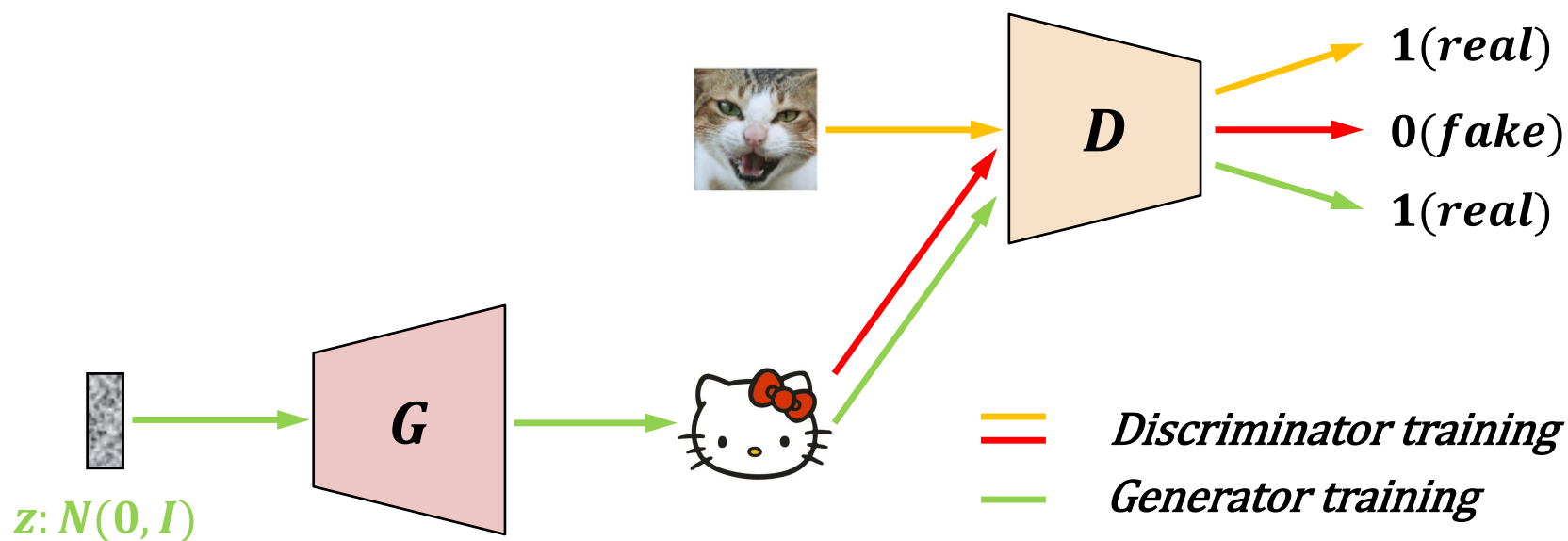


生成式模型 (Generative Models)

- 生成式模型概论
- 自回归生成模型
- 变分自编码器
- 扩散模型
- 生成对抗网络
- 条件生成式模型

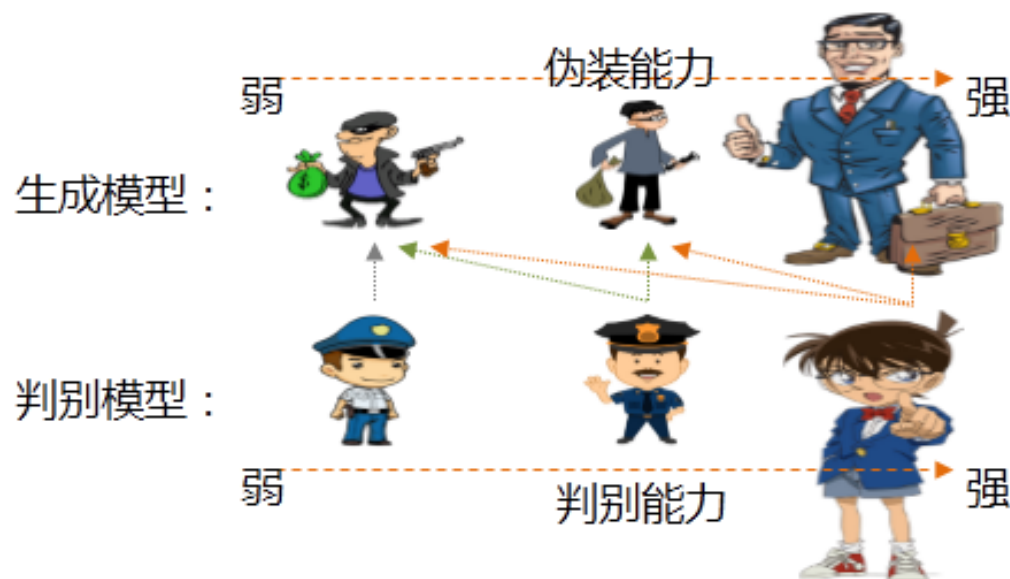
生成对抗网络 (Generative Adversarial Networks)

- 生成对抗网络包含两个子网络：**生成网络(G)**和**判别网络(D)**
- 生成网络G的输入为均值为0，协方差矩阵为单位矩阵的高斯噪声，输出为生成的“假”数据；判别网络D可以看作一个二分类器，用于分辨输入的数据是“真”数据还是“假”数据。



生成对抗网络

- GAN受博弈论启发，将生成问题视作**生成器G**和**判别器D**这两个网络的对抗和博弈：前者试图产生更真实的数据，后者试图更准确地分辨真实数据与生成数据。



- 由此，两个网络在对抗中进步，在进步后继续对抗，由生成式网络得的数据也就越来越完美，越来越逼近真实数据，从而可以生成想要得到的数据。



生成对抗网络

- 设 z 为随机噪声， x 为真实数据，判别器 D 可以看作一个二分类器，那么采用交叉熵损失，GAN的优化目标可以写作：

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

- 其中第一项的 $D(x)$ 表示判别器对真实数据的判断，第二项 $D(G(z))$ 则表示对生成数据的判断。
 - 判别器 D 的目标是最大化这个公式，也就是甄别出哪些数据是来自真实数据分布的。
 - 生成器 G 的目标是最小化这个公式，也就是让自己生成的数据被判别器判断为来自真实数据分布。
- 通过这样一个极大极小博弈，循环交替地分别优化 G 和 D 来训练所需要的生成式网络与判别式网络，直到到达Nash均衡点。

生成对抗网络

□ Minmax目标函数

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

□ 交替优化

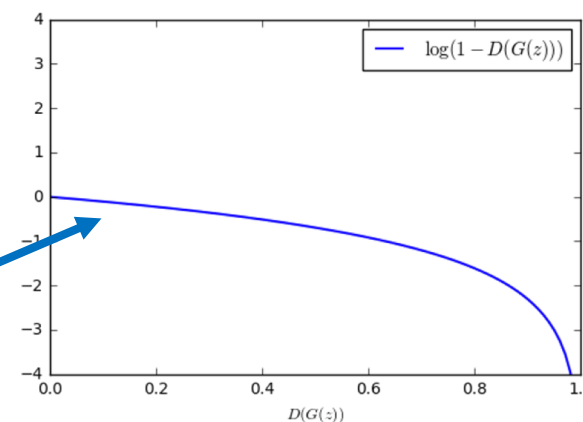
■ 优化判别器（梯度上升）

$$\max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

■ 优化生成器（梯度下降）

$$\min_{\theta_g} \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z)))$$

梯度消失



生成对抗网络

□ Minmax目标函数

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

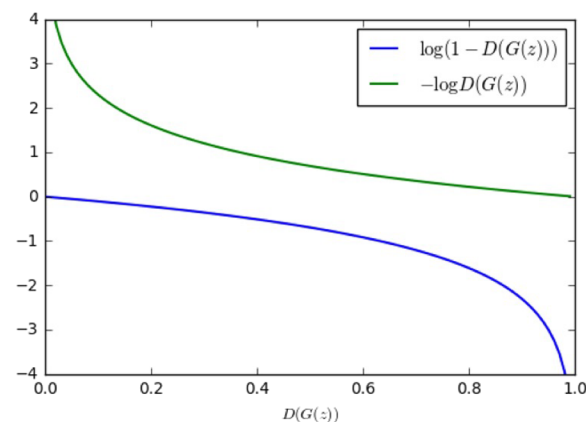
□ 交替优化

■ 优化判别器（梯度上升）

$$\max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

■ 优化生成器（目标函数替换，梯度上升）

$$\max_{\theta_g} \mathbb{E}_{z \sim p(z)} \log(D_{\theta_d}(G_{\theta_g}(z)))$$





生成对抗网络的优缺点

□ GAN的优点

- 模型的训练不需要对隐变量做推断
- 生成高质量图像
- 判别器作为可自主学习的度量函数，定义十分灵活，使得GAN很容易与其他任务结合

□ GAN的缺点

- 训练不稳定
- 模式丢失（mode collapse）问题
- 缺少对 $p(x)$, $p(z|x)$ 的建模和计算



生成式模型 (Generative Models)

- 生成式模型概论
- 自回归生成模型
- 变分自编码器
- 扩散模型
- 生成对抗网络
- 条件生成式模型

条件生成式模型

□ 条件自回归模型

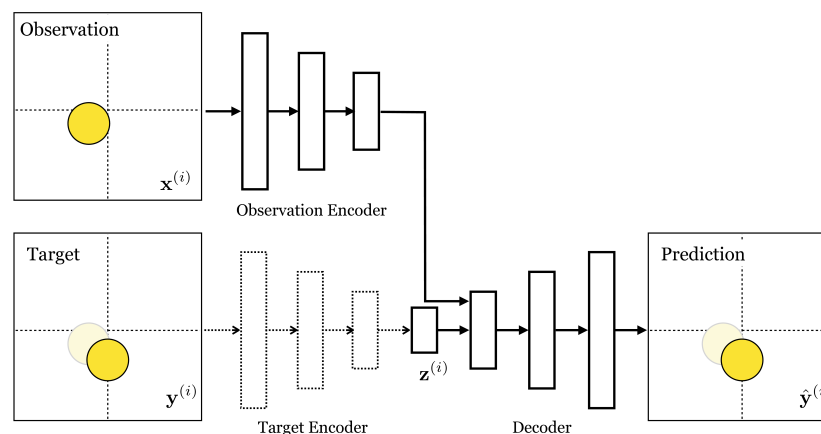
$$p_{\theta}(\mathbf{y} \mid \mathbf{x}) = \prod_{i=1}^n p_{\theta}(y_i \mid y_1, \dots, y_{i-1}, x_1, \dots, x_m)$$

□ 条件变分自编码器 (cVAE)

VAE likelihood model $p_{\theta}(\mathbf{x}) = \int_{\mathbf{z}} p_{\theta}(\mathbf{x} \mid \mathbf{z}) p_{\mathbf{z}}(\mathbf{z}) d\mathbf{z}$

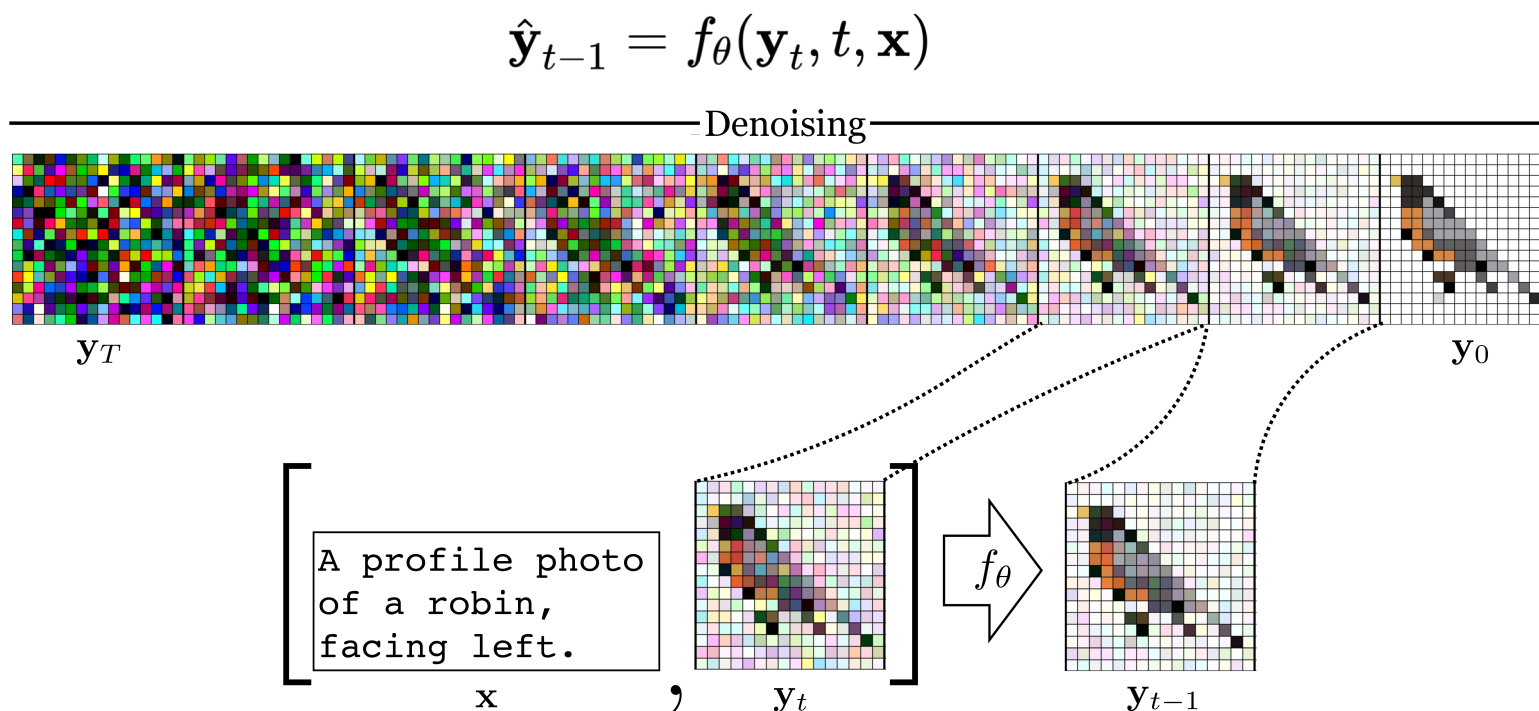
$$p_{\theta}(\mathbf{y} \mid \mathbf{x}) = \int_{\mathbf{z}} p_{\theta}(\mathbf{y} \mid \mathbf{z}, \mathbf{x}) p_{\mathbf{z}}(\mathbf{z} \mid \mathbf{x}) d\mathbf{z}$$

cVAE likelihood model $p_{\theta}(\mathbf{y} \mid \mathbf{x}) = \int_{\mathbf{z}} p_{\theta}(\mathbf{y} \mid \mathbf{z}, \mathbf{x}) p_{\mathbf{z}}(\mathbf{z}) d\mathbf{z}$



条件生成式模型

□ 条件扩散模型

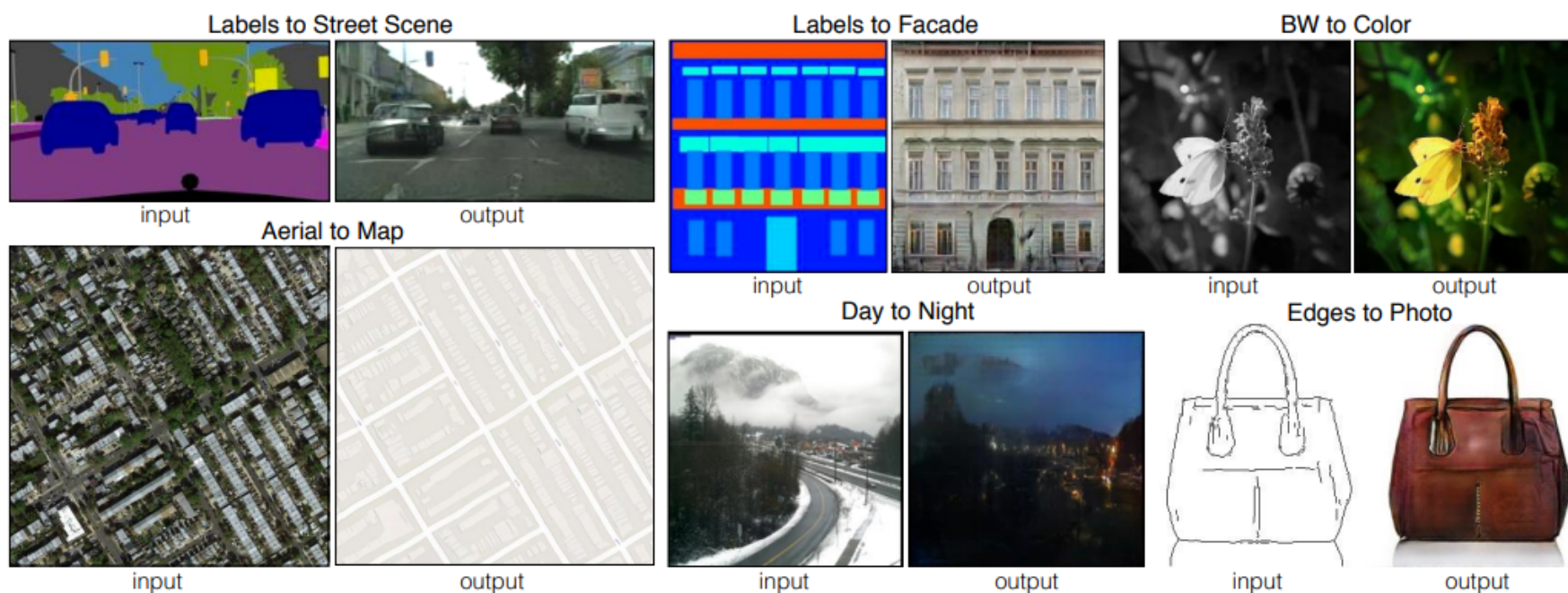


□ 条件生成对抗网络 (cGAN)

$$\arg \min_{\theta} \max_{\phi} \mathbb{E}_{\mathbf{z}, \mathbf{x}, \mathbf{y}} [\log d_{\phi}(\mathbf{x}, g_{\theta}(\mathbf{x}, \mathbf{z})) + \log(1 - d_{\phi}(\mathbf{x}, \mathbf{y}))]$$

应用举例

□ 图像翻译 (Image-to-image Translation)



$$\arg \min_G \max_D \mathbb{E}_{\mathbf{z}, \mathbf{x}, \mathbf{y}} \left[\log d_\phi(\mathbf{x}, g_\theta(\mathbf{x}, \mathbf{z})) + \log(1 - d_\phi(\mathbf{x}, \mathbf{y})) + \|\mathbf{g}_\theta(\mathbf{x}) - \mathbf{y}\|_1 \right]$$

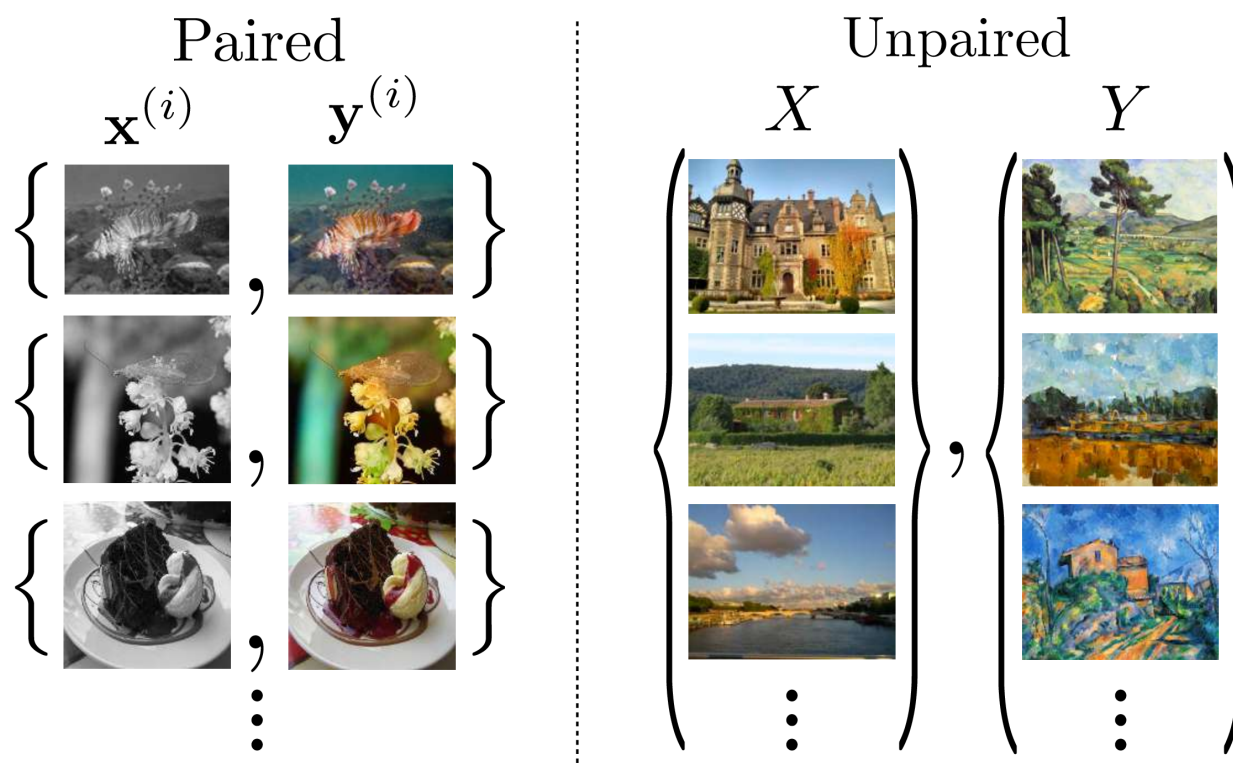
PatchGAN

$$d_\phi(\mathbf{x}, \mathbf{y}) = \frac{1}{NM} \sum_{i=0}^N \sum_{j=0}^M d_\phi^{\text{patch}}(\mathbf{x}[i:i+k, j:j+k], \mathbf{y}[i:i+k, j:j+k])$$

- Isola, Phillip, et al. "Image-to-image translation with conditional adversarial networks". In CVPR, 2017.

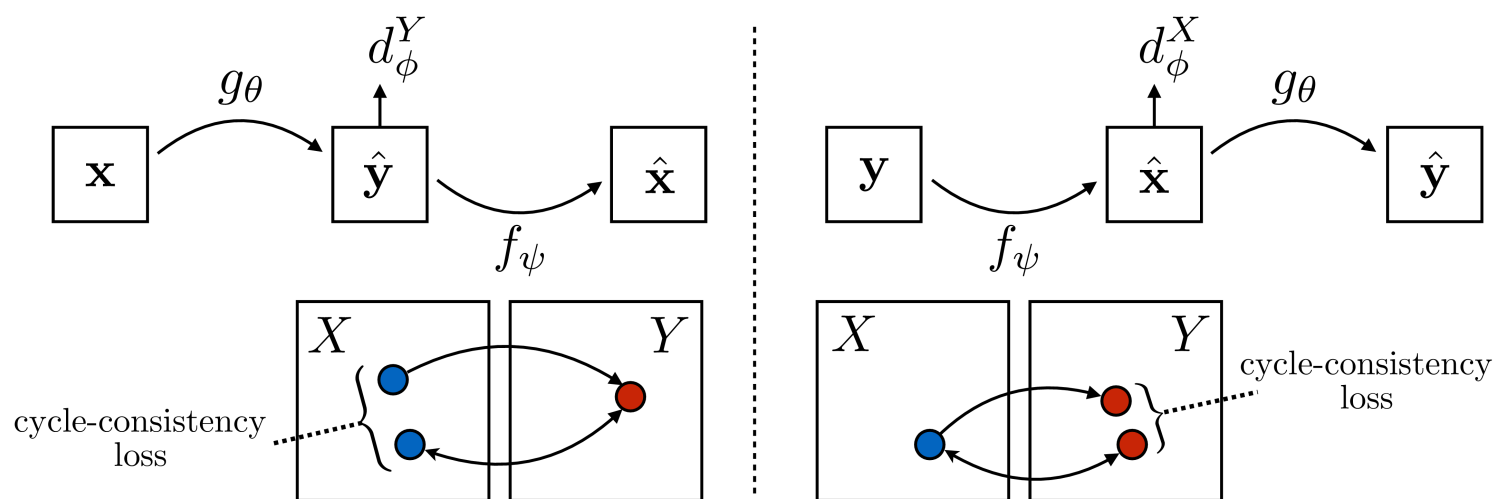
应用举例

□ 非成对图像翻译



应用举例

□ 非成对图像翻译



$$\arg \min_{\theta, \psi} \max_{\phi} \mathbb{E}_{\mathbf{x}, \mathbf{y}} \left[\log d_{\phi}^Y(g_{\theta}(\mathbf{x})) + \log(1 - d_{\phi}^Y(\mathbf{y})) + \|\mathbf{x} - f_{\psi}(g_{\theta}(\mathbf{x}))\|_1 + \right. \\ \left. \log d_{\phi}^X(f_{\psi}(\mathbf{y})) + \log(1 - d_{\phi}^X(\mathbf{x})) + \|\mathbf{y} - g_{\theta}(f_{\psi}(\mathbf{y}))\|_1 \right]$$

- J.Y. Zhu, et al. "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks". In ICCV, 2017.