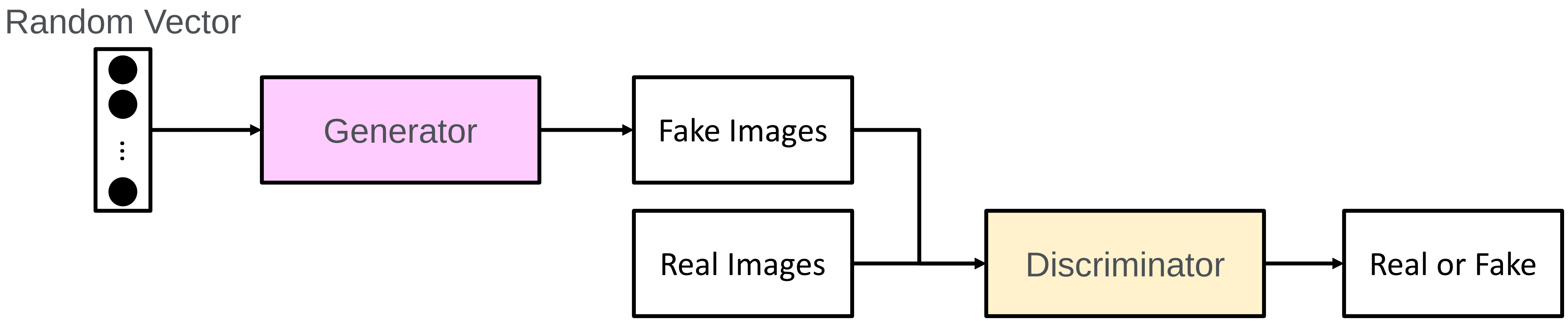
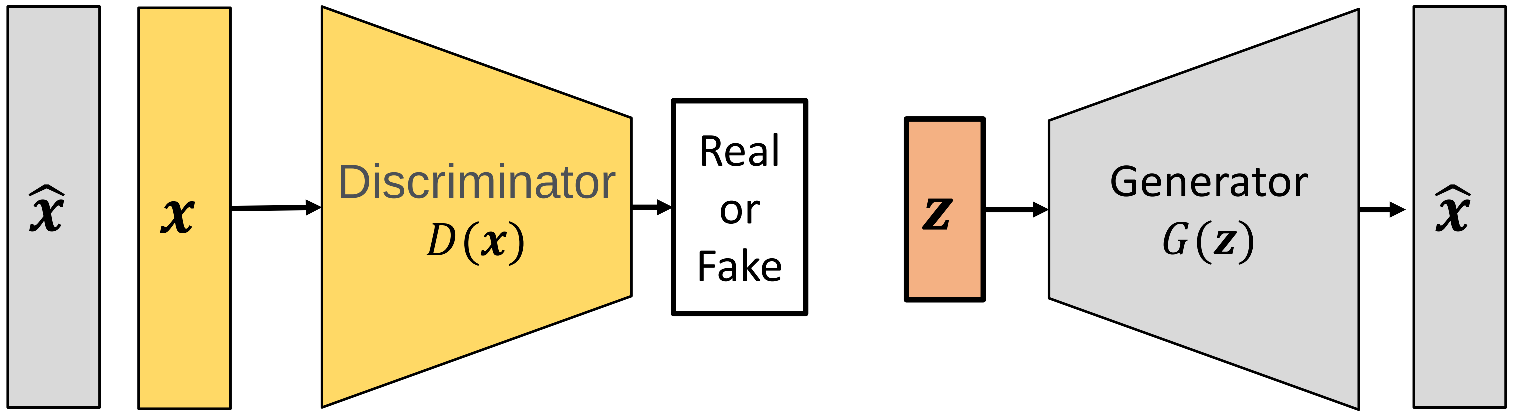
深度學習
Diffusion, AutoEncoder & GAN

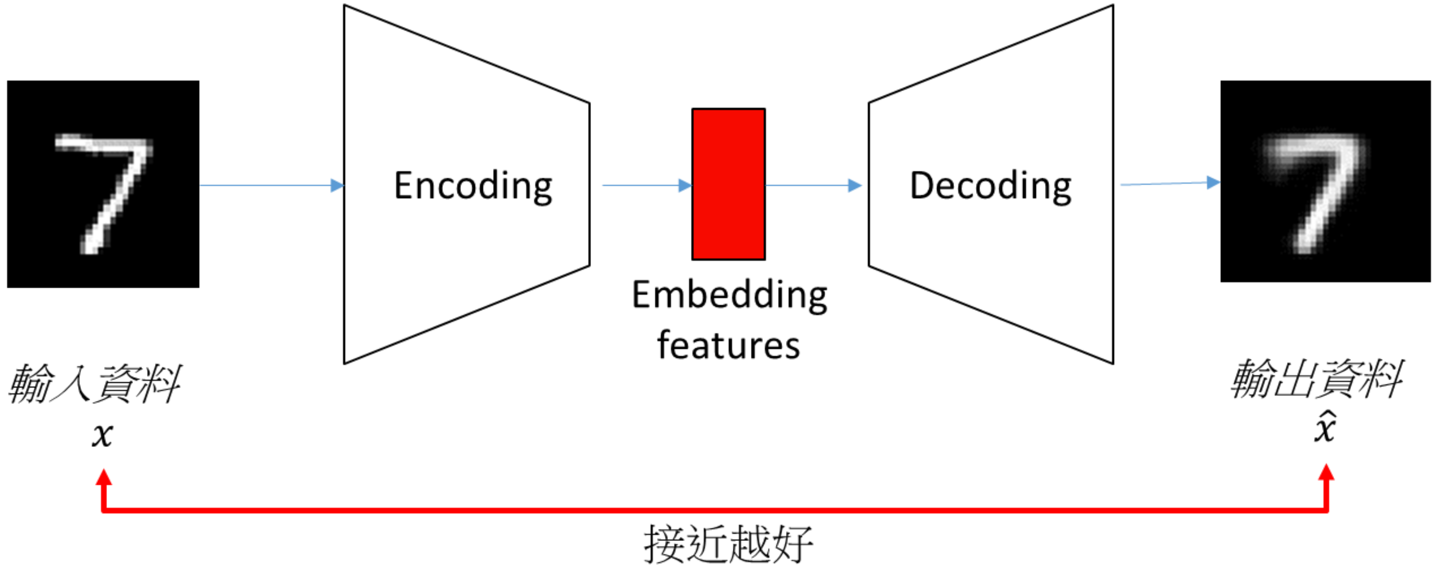
黃志勝 (Tommy Huang)
義隆電子 人工智慧研發部
國立陽明交通大學 AI學院 合聘助理教授
國立台北科技大學 電資學院合聘助理教授

GAN

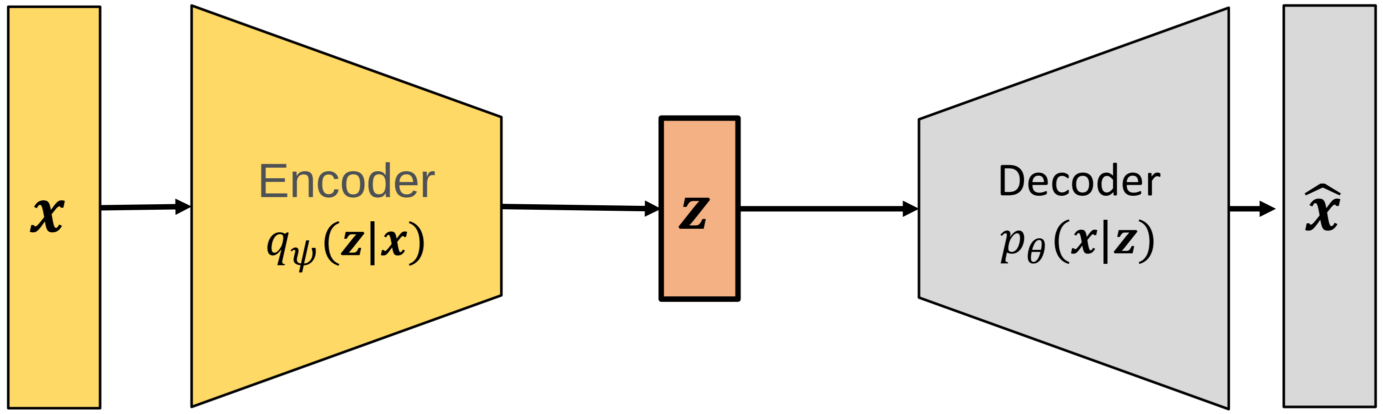

Random Vector
Generator
Fake Images
Real Images
Discriminator
Real or Fake


$\hat{\boldsymbol{x}}$
$\boldsymbol{x}$
Discriminator $D(\boldsymbol{x})$
Real or Fake
$\boldsymbol{z}$
Generator $G(\boldsymbol{z})$
$\hat{\boldsymbol{x}}$

VAE

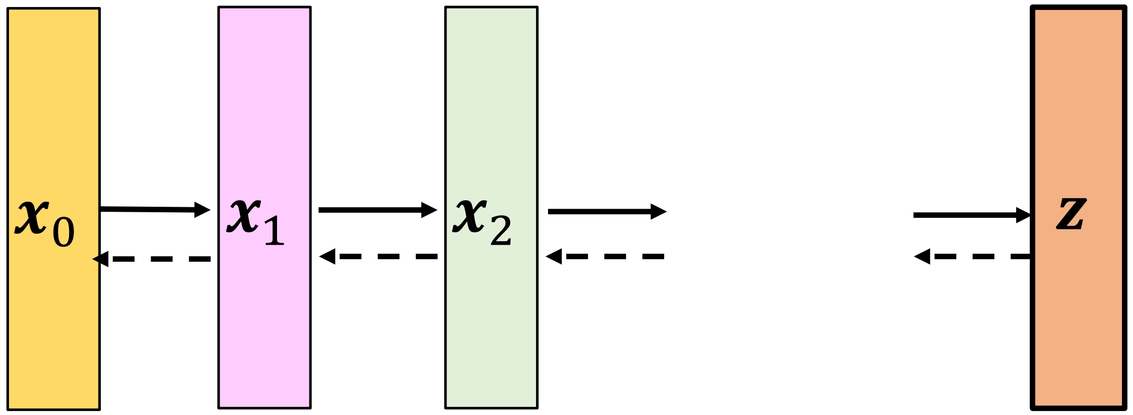


Encoding
Embedding features
Decoding
輸入資料
x
輸出資料
$\hat{x}$
接近越好



$\boldsymbol{x}$
Encoder
$q_{\psi}(\boldsymbol{z}|\boldsymbol{x})$
$\boldsymbol{z}$
Decoder
$p_{\theta}(\boldsymbol{x}|\boldsymbol{z})$
$\widehat{\boldsymbol{x}}$

Diffusion Model



Gradually add Gaussian noise and then reverse

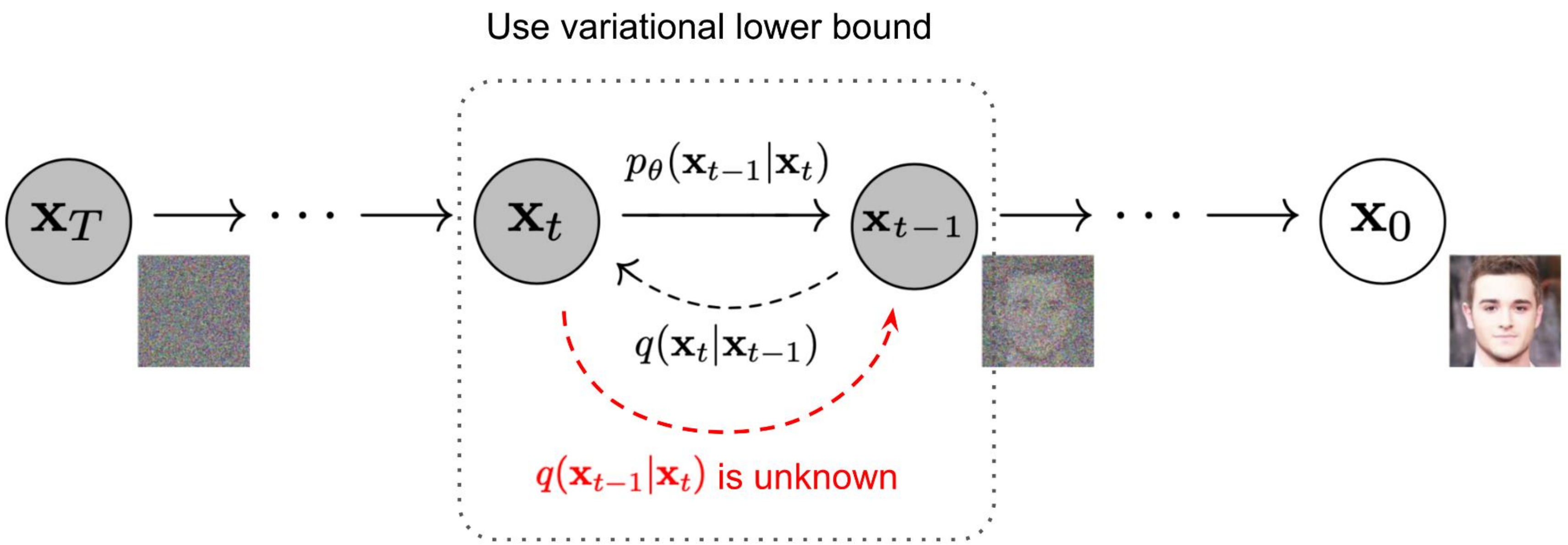
Markov chain of diffusion steps to slowly add random noise to data and then learn to reverse the diffusion process to construct desired data samples from the noise

Denoising diffusion probabilistic models (**DDPM**; Ho et al. 2020).

$$p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{x}_{t-2}, \ldots \boldsymbol{x}_{t-p}) = p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{x}_{t-2}, \ldots \boldsymbol{x}_{t-p}, \boldsymbol{x}_{t-p-1}, \ldots \boldsymbol{x}_0)$$

https://lilianweng.github.io/posts/2021-07-11-diffusion-models/

Diffusion Model


Use variational lower bound
$\mathbf{x}_T$
$\mathbf{x}_t$
$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$
$\mathbf{x}_{t-1}$
$q(\mathbf{x}_t|\mathbf{x}_{t-1})$
$\mathbf{x}_0$
$q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is unknown

Real data Distribution: $\boldsymbol{x}_0 \sim q(\boldsymbol{x})$

$\boldsymbol{x}_1 = \boldsymbol{x}_0 + \alpha_0 N(0,1) \sim q(\boldsymbol{x}) + \alpha_0 N(0,1)$

$\boldsymbol{x}_2 = \boldsymbol{x}_1 + \alpha_1 N(0,1) \sim q(\boldsymbol{x}) + (\alpha_0 + \alpha_1) N(0,1)$

…

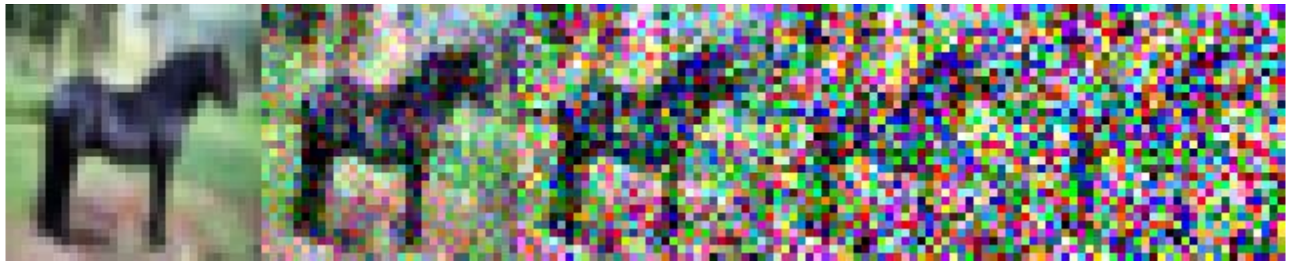
$$\boldsymbol{x}_T \sim q(\boldsymbol{x}) + \left(\sum_{t=0}^{T} \alpha_t\right) N(0,1) \approx N(0,1)$$

Forward diffusion process (模糊化程序)

$$q(x_1|x_0) = N(x_1; \sqrt{1-\beta_1}x_0, \beta_1\mathbf{I})$$
$$q(x_2|x_1) = N(x_2; \sqrt{1-\beta_2}x_1, \beta_2\mathbf{I})$$
$$\dots$$
$$q(x_T|x_{T-1}) = N(x_T; \sqrt{1-\beta_T}x_{T-1}, \beta_T\mathbf{I})$$

$$q(x_t|x_{t-1}) = N\left(x_{t-1}; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I}\right), t = 1,2,\dots,T$$
$$q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1})$$

β稱為 variance schedule，介於0~1(也可以學來，也可以是固定值(DDPM是固定值))



Reverse diffusion process (逆模糊化程序)

逆模糊化我們定義成：

$$p_\theta(x_{t-1}|x_t)$$

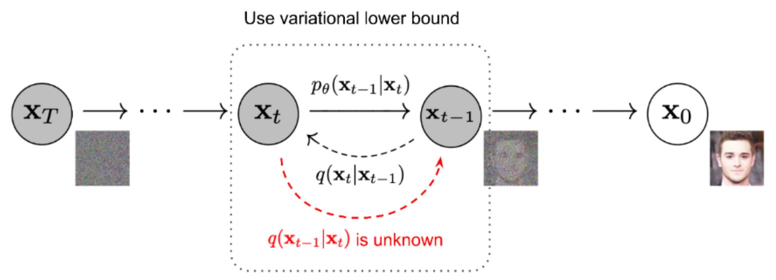
很難去估計$q(x_t|x_{t-1})$，但我們知道

$$q(x_t|x_{t-1}) = N\left(x_{t-1};\ \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I}\right)$$

模型p_θ(服從常態分佈)去估計條件機率來表示reverse diffusion process

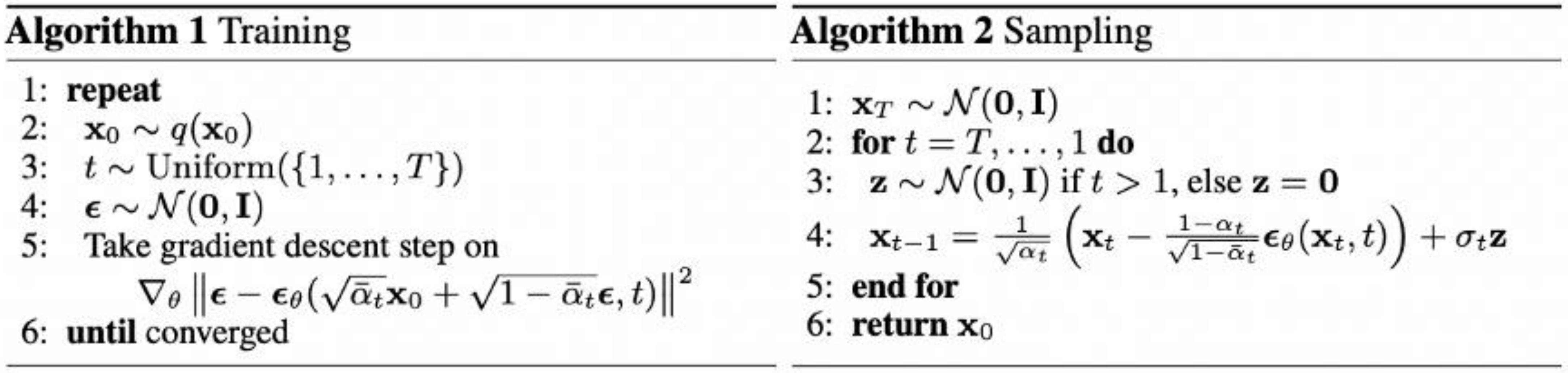
$$p_\theta(x_{t-1}|x_t) \sim N(x_{t-1}; \mu_\theta(x_t,t); \Sigma_\theta(x_t,t))$$

$$p_\theta(x_{0:T}) = p(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)$$


Use variational lower bound
$q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is unknown

$$\begin{tikzcd}
\mathbf{x}_T \arrow[r] & \cdots \arrow[r] & \mathbf{x}_t \arrow[r, "p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)"] & \mathbf{x}_{t-1} \arrow[l, bend left, dashed, "q(\mathbf{x}_t|\mathbf{x}_{t-1})"] \arrow[r] & \cdots \arrow[r] & \mathbf{x}_0
\end{tikzcd}$$

DDPM



Algorithm 1 Training

```
1: repeat
2:   $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3:   $t \sim \mathrm{Uniform}(\{1, \dots, T\})$
4:   $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5:   Take gradient descent step on
         $\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$
6: until converged
```

Algorithm 2 Sampling

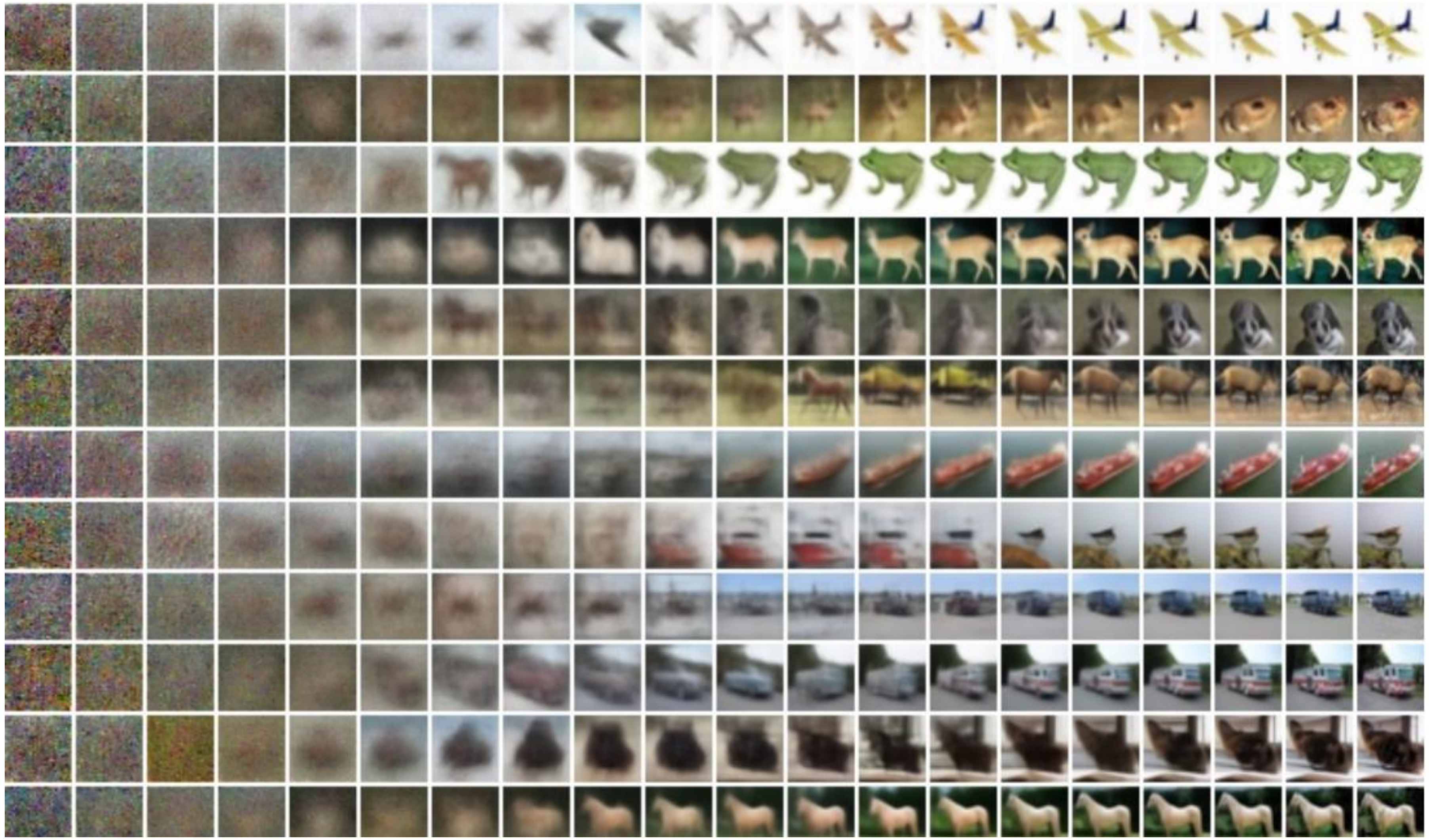
```
1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: for $t = T, \dots, 1$ do
3:   $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4:   $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left( \mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$
5: end for
6: return $\mathbf{x}_0$
```

$$\underset{\theta}{\operatorname{argmin}} \left[D_{KL}(q(x_T|x_0)||p_\theta(x_T)) - \log(p_\theta(x_0|x_1)) + \sum_{t=2}^{T} D_{KL}(q(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t)) \right]$$

$$\begin{aligned} L_t^{\text{simple}} &= \mathbb{E}_{t\sim[1,T],\mathbf{x}_0,\epsilon_t}\left[\|\boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\|^2 \right] \\ &= \mathbb{E}_{t\sim[1,T],\mathbf{x}_0,\epsilon_t}\left[\|\boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t, t)\|^2 \right] \end{aligned}$$

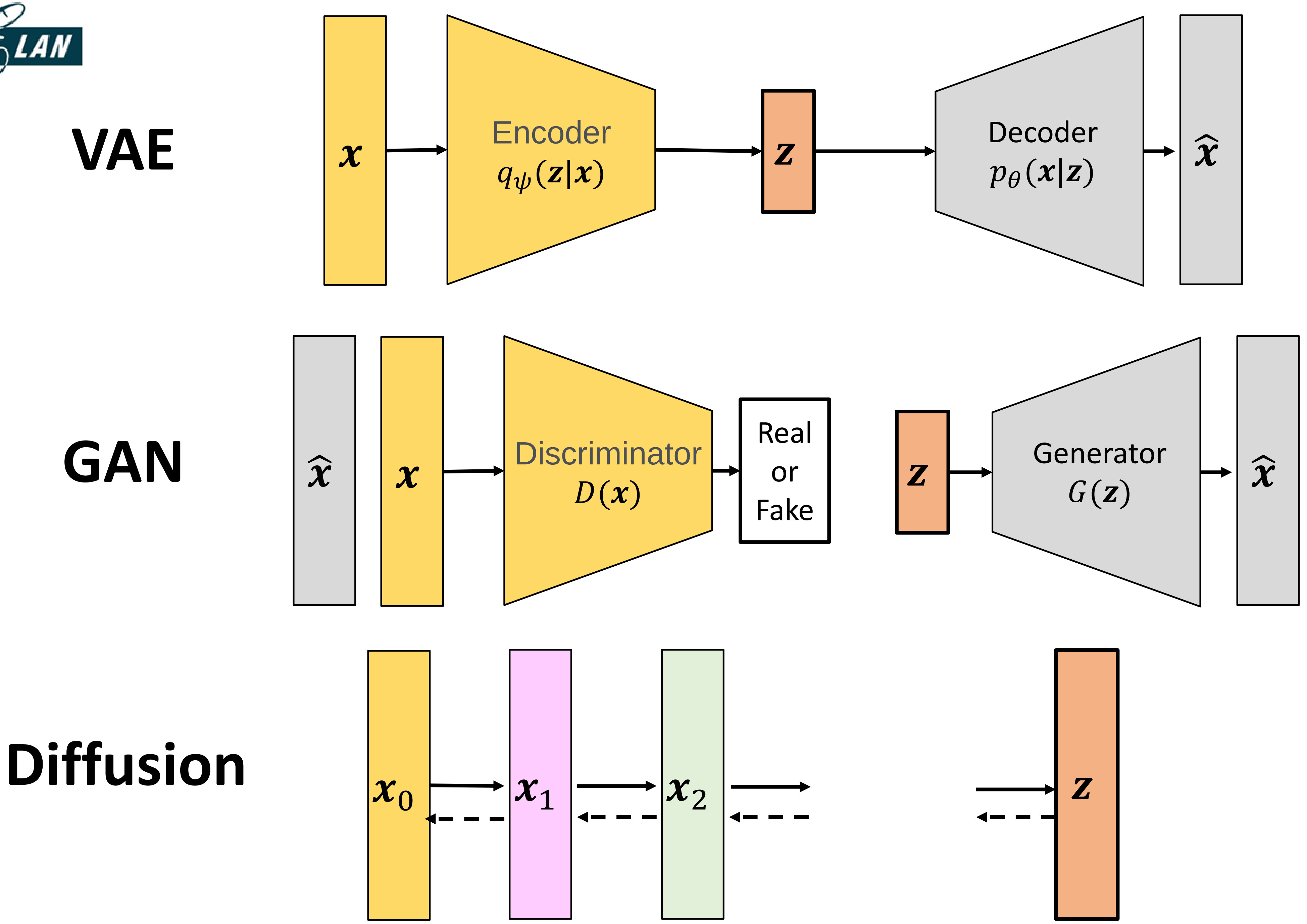
https://lilianweng.github.io/posts/2021-07-11-diffusion-models/

Reverse diffusion process



Implementation (別人寫的)

- https://colab.research.google.com/drive/1NFxjNI-UIR7Ku0KERmv7Yb_586vHQW43?usp=sharing#scrollTo=txWbmGFRcyQ2


VAE
$\boldsymbol{x}$
Encoder
$q_\psi(\boldsymbol{z}|\boldsymbol{x})$
$\boldsymbol{z}$
Decoder
$p_\theta(\boldsymbol{x}|\boldsymbol{z})$
$\hat{\boldsymbol{x}}$
GAN
$\hat{\boldsymbol{x}}$
$\boldsymbol{x}$
Discriminator
$D(\boldsymbol{x})$
Real or Fake
$\boldsymbol{z}$
Generator
$G(\boldsymbol{z})$
$\hat{\boldsymbol{x}}$
Diffusion
$\boldsymbol{x}_0$
$\boldsymbol{x}_1$
$\boldsymbol{x}_2$
$\boldsymbol{z}$