

HUST

ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

ONE LOVE. ONE FUTURE.



**ĐẠI HỌC
BÁCH KHOA HÀ NỘI**
HANOI UNIVERSITY
OF SCIENCE AND TECHNOLOGY

Diffusion Models

ONE LOVE. ONE FUTURE.

Nội dung chính

- Idea
- Denoising Diffusion Probabilistic Models
 - Forward/Backward Process
 - Diffusion Model Architect
- Inference Process
 - Noise-Conditioned Score Network
 - Denoising Diffusion Implicit Models
- Ứng dụng trong bài toán thực tế



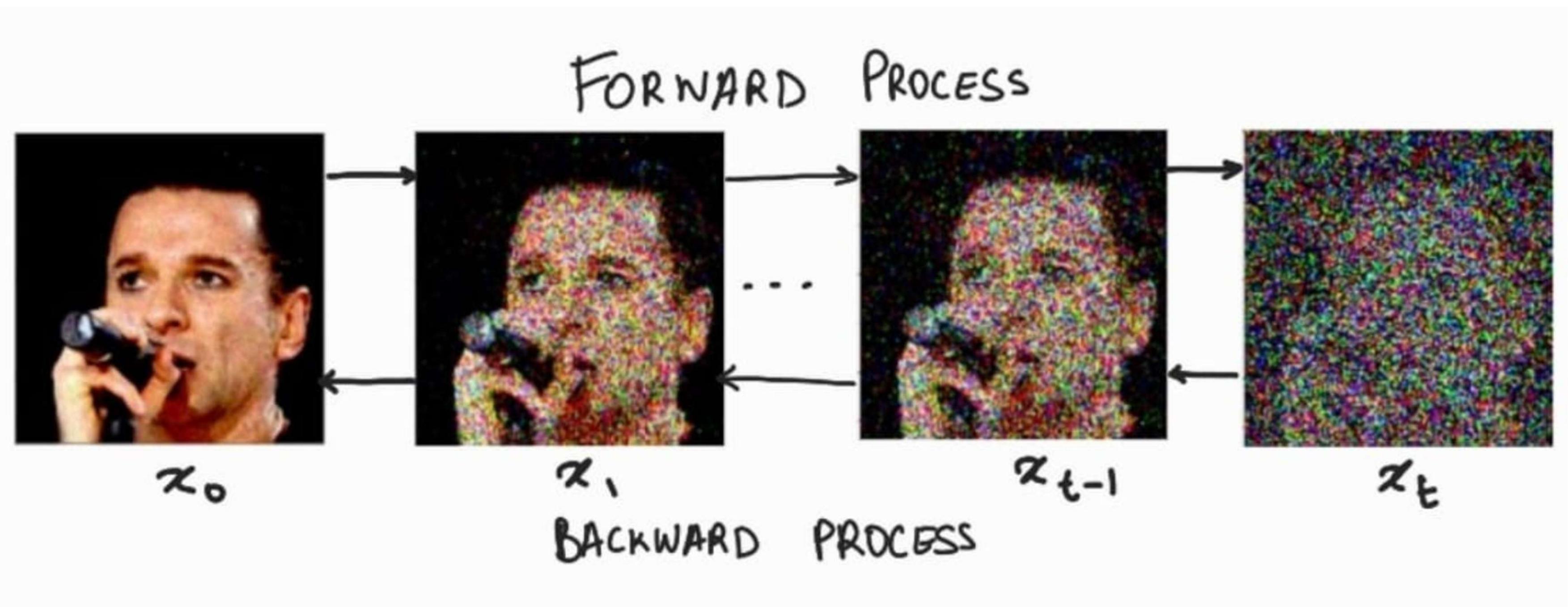
- Diffusion Model lần đầu được đề xuất năm 2015 dưới dạng ý tưởng sử dụng một quá trình khuếch tán (diffusion process) để dần dần phá hủy cấu trúc trong dữ liệu, và sau đó xây dựng một mô hình sinh (generative model) học cách đảo ngược quá trình này
- Đến 2020, có hai hướng tiếp cận độc lập nhau cải tiến ý tưởng trên và trở thành khuôn mẫu của các mô hình khuếch tán hiện đại:
 - Noise-Conditioned Score Network (2019)
 - Denoising Diffusion Probabilistic Models (2020)

Denoising Diffusion Probabilistic Models

- Diffusion Model về cơ bản là một neural network học cách khử nhiễu dần dần, bắt đầu từ một dữ liệu nhiễu hoàn toàn
- Mô hình gồm 2 quá trình:
 - Quá trình khuếch tán thuận (forward diffusion) - Cố định: dần dần thêm nhiễu Gaussian
 - Quá trình khuếch tán ngược (reverse diffusion) - Học được: mạng được huấn luyện để khử nhiễu dần dần



Denoising Diffusion Probabilistic Models



Forward Diffusion Process

- Forward Process là một chuỗi các phép biến đổi xác suất cố định với mục tiêu biến đổi dữ liệu thật x_0 thành nhiễu hoàn toàn x_T
- Chúng ta định nghĩa xác suất chuyển tiếp từ bước $t-1$ đến t là:

$$q(x_t | x_{t-1}) = N(x_t; \sqrt{1 - \beta_t}x_{t-1}, \beta_t)$$
$$x_t = \sqrt{1 - \beta_t}x_{t-1} + \sqrt{\beta_t}\varepsilon, \varepsilon \sim N(0, 1)$$

- Ở mỗi bước, dữ liệu gốc bị giảm đi một chút, và nhiễu tăng lên một chút. Trong đó β_t được gọi là Variance Schedule, quyết định tốc độ phá hủy dữ liệu



Reparameterization Trick

- Reparameterization Trick kết hợp với tính chất của phân phối Gaussian (Tổng của 2 phân phối chuẩn là một phân phối chuẩn), ta có thể gộp tất cả các bước trung gian lại thành một bước duy nhất

$$\begin{aligned}x_t &= \sqrt{1 - \beta_t} x_{t-1} + \sqrt{\beta_t} \varepsilon_{t-1} \\&= \sqrt{1 - \beta_t} \sqrt{1 - \beta_{t-1}} x_{t-2} + \sqrt{(1 - \beta_t)\beta_{t-1}} \varepsilon_{t-2} + \sqrt{\beta_t} \varepsilon_{t-1} \\&= \sqrt{(1 - \beta_t)(1 - \beta_{t-1})} x_{t-2} + \sqrt{1 - (1 - \beta_t)(1 - \beta_{t-1})} \varepsilon \\&\Rightarrow x_t = \sqrt{\alpha_t} x_0 + \sqrt{1 - \alpha_t} \varepsilon\end{aligned}$$

Reverse Diffusion Process

- Nếu biết được phân phối ngược $q(x_{t-1} | x_t)$, ta hoàn toàn có thể đảo ngược quá trình và có được dữ liệu thật từ nhiều
- Ta sử dụng một mô hình để học/xấp xỉ xác suất cần tìm:

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

- Tác giả gốc cho rằng việc bắt mạng học cùng lúc mean và variance khiến việc huấn luyện trở nên khó hơn và cũng khó hội tụ hơn nên quyết định tự tay gán variance cho một hằng số toán học



Reverse Diffusion Process

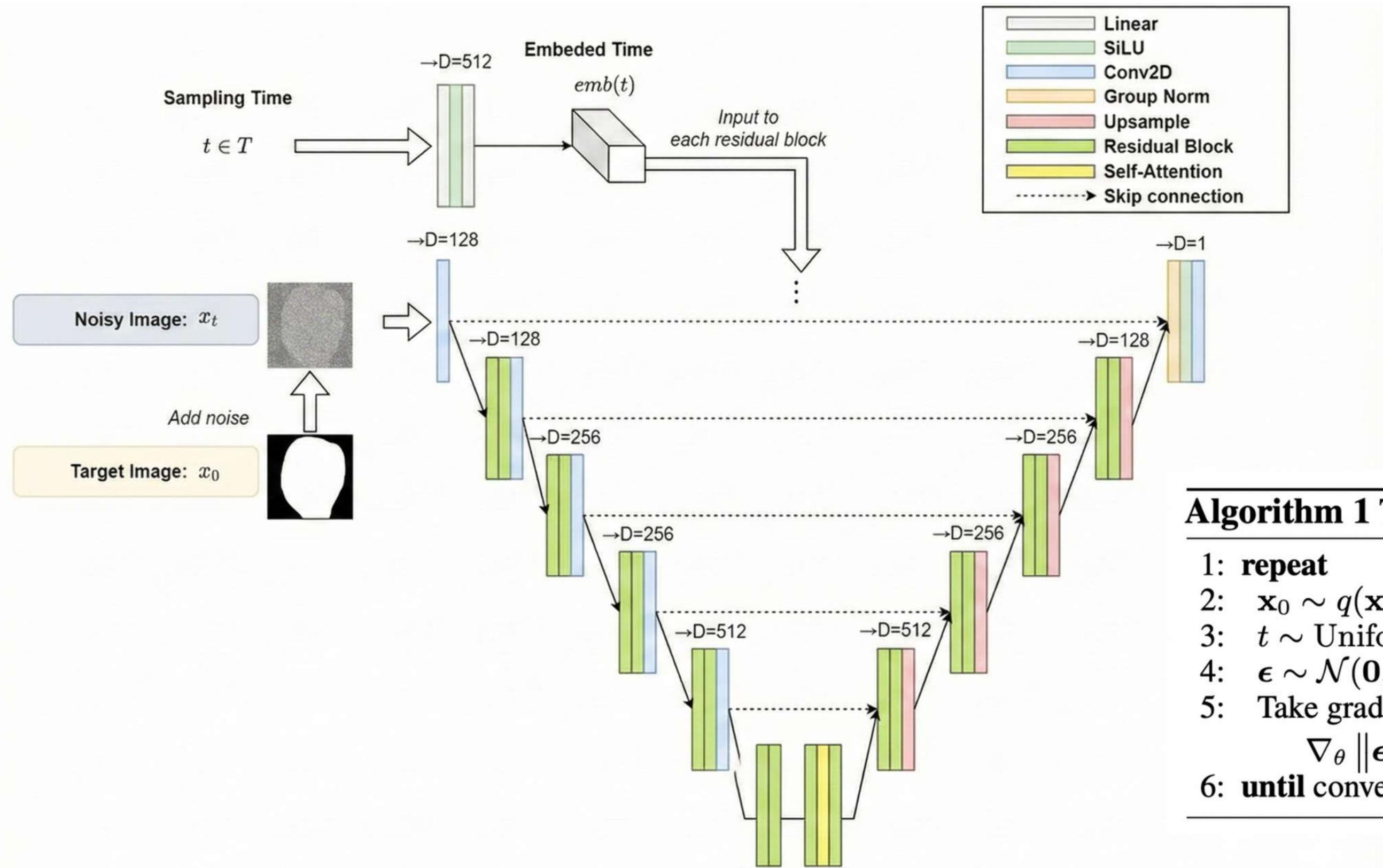
- Thay vì học trực tiếp mean, mô hình sẽ học cách dự đoán ra nhiễu ở step t và tính mean theo công thức:

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{1 - \beta_t}} (x_t - \frac{\beta_t}{\sqrt{1 - \alpha_t}} \varepsilon_\theta(x_t, t))$$

- Khi đó hàm mất mát của mô hình chỉ đơn giản là MSE Loss giữa nhiễu thật và nhiễu dự đoán



Diffusion Model Architecture

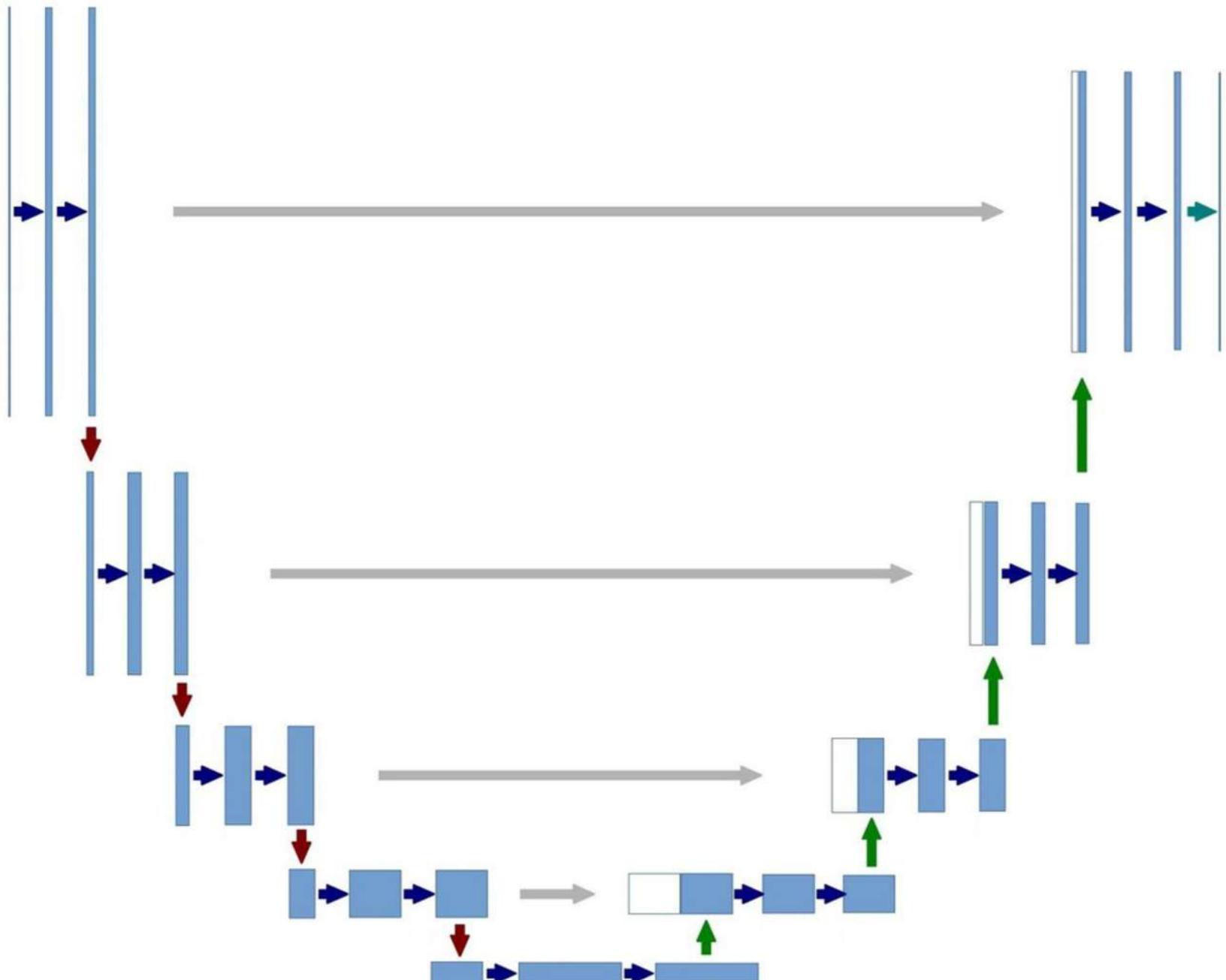


Algorithm 1 Training

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

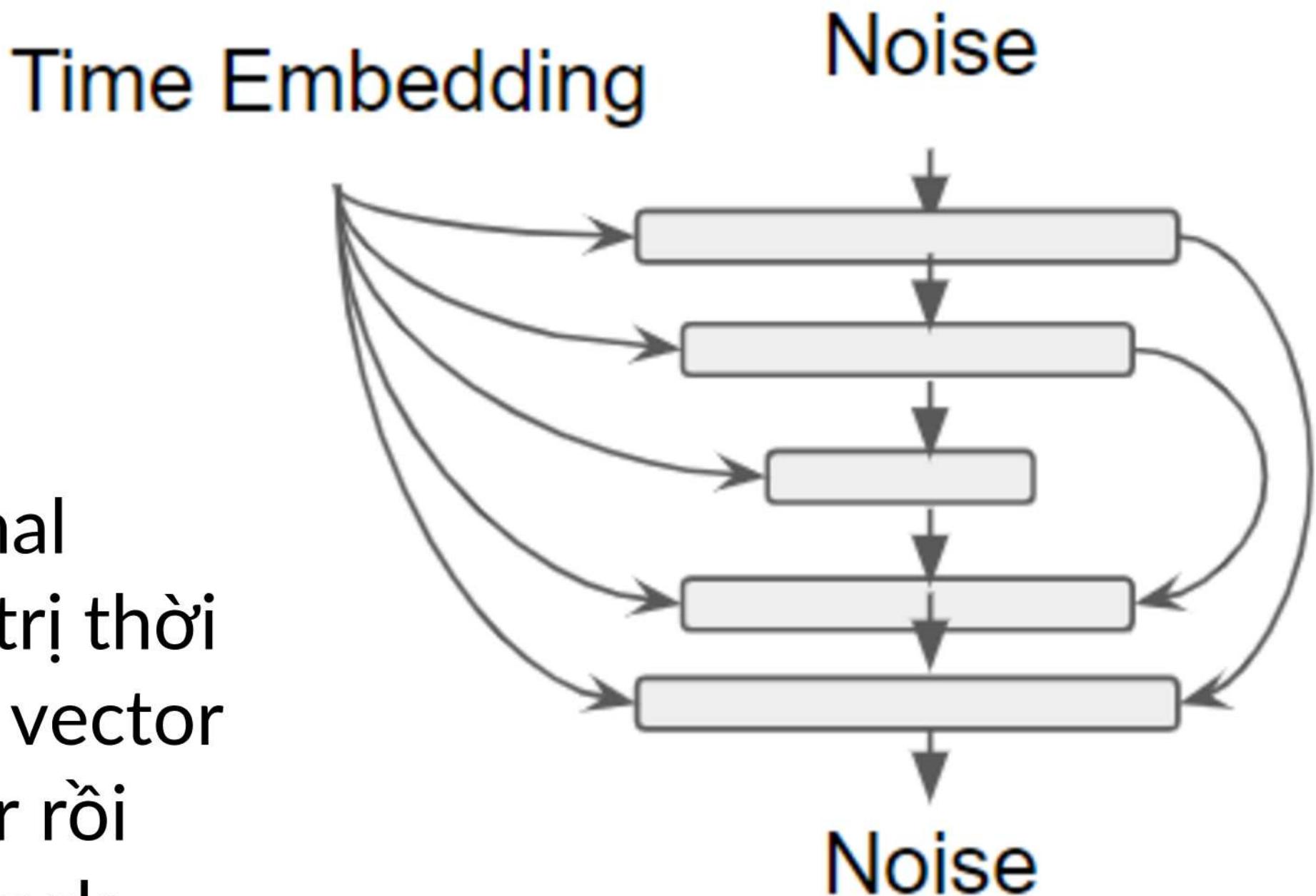
U-Net Backbone

- DDPM sử dụng cấu trúc U-Net kết hợp với các ResNet block
- Lý do sử dụng U-Net + ResNet:
 - Sự thành công của mô hình Pix2Pix (GAN) khi lấy U-Net làm Generator
 - Mô hình U-Net càn quét các cuộc thi ảnh y tế giai đoạn 2018-2020
 - Mô hình sinh khác nổi tiếng là PixelCNN++ sử dụng ResNet

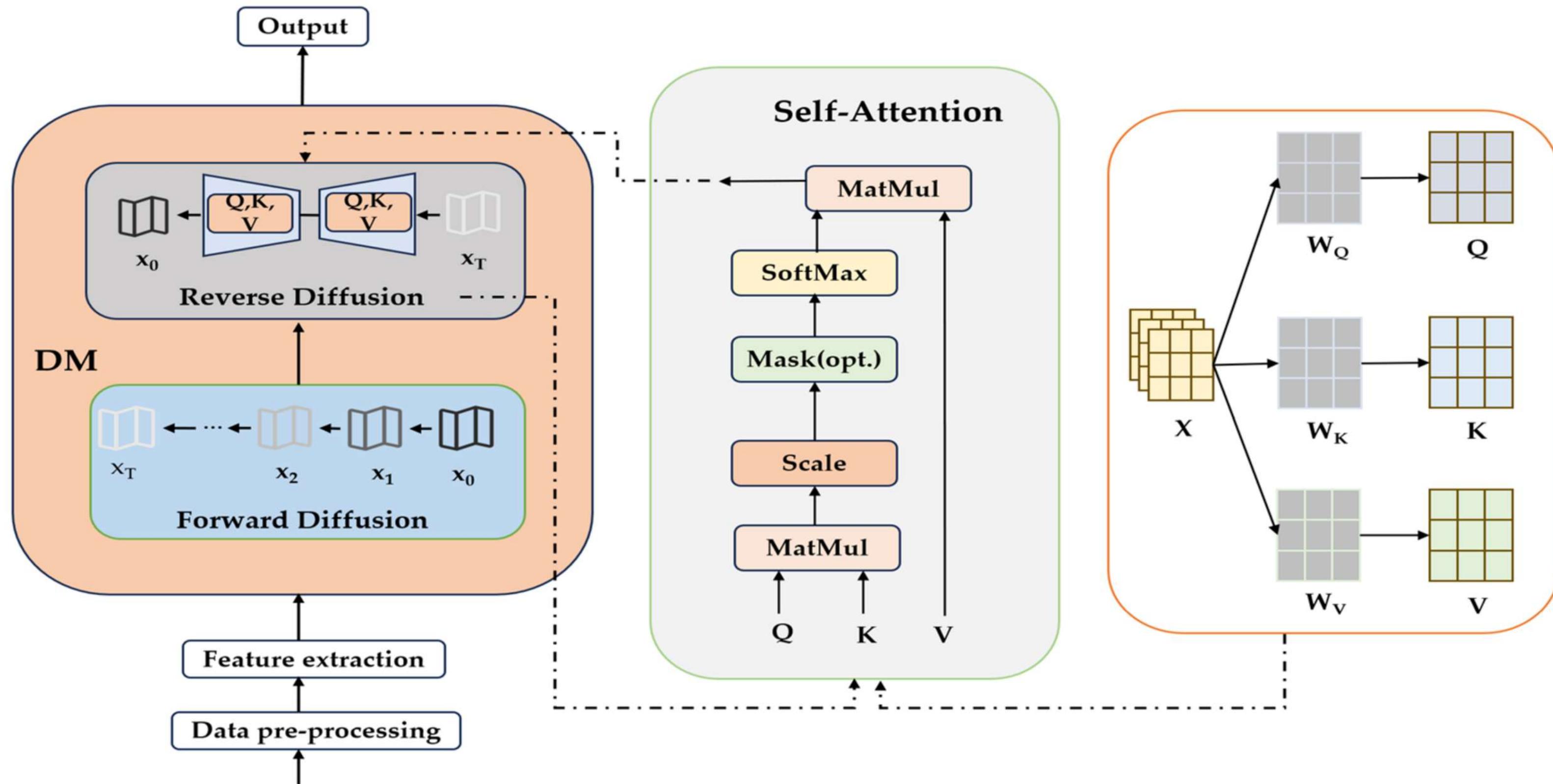


Time Embedding

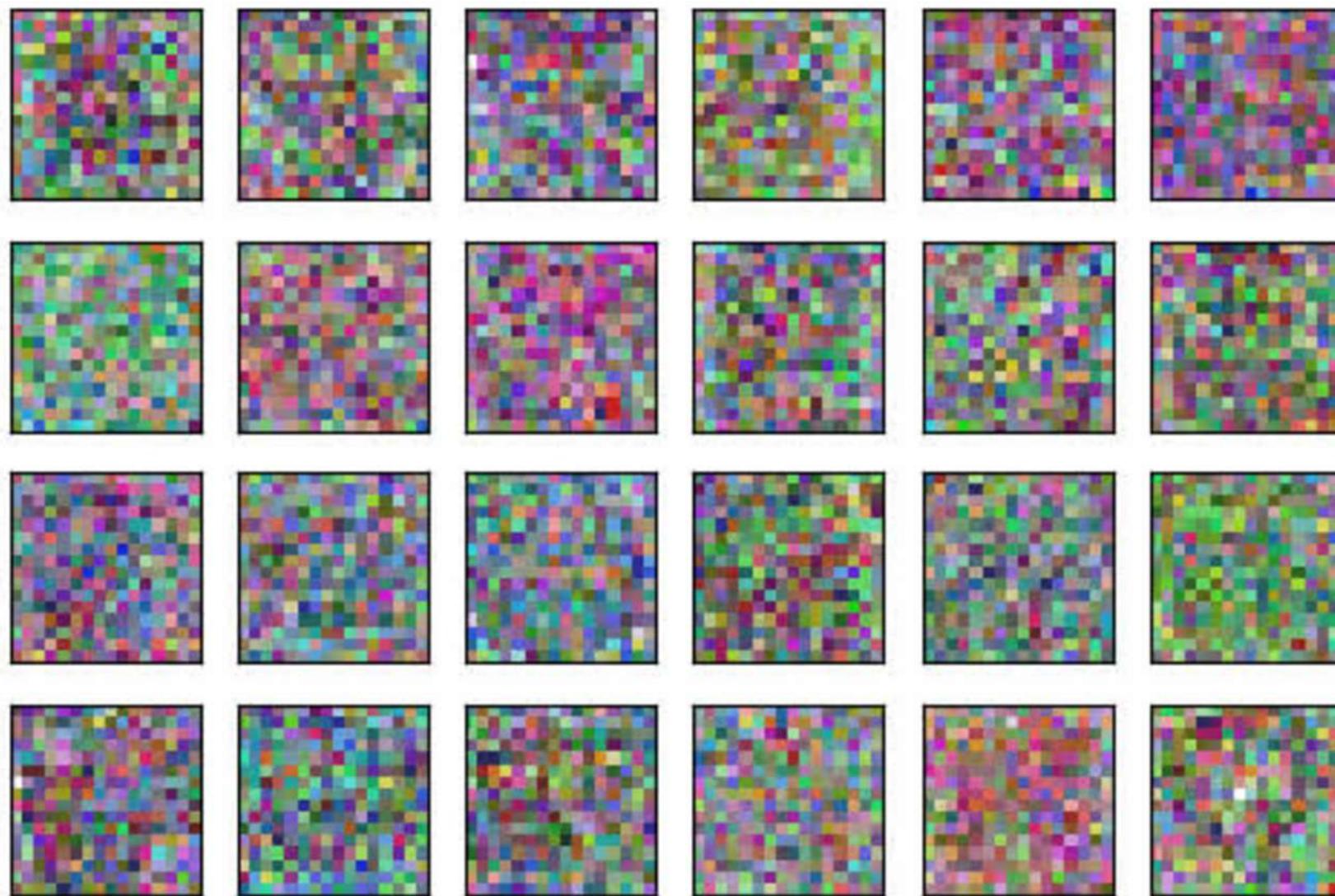
- Mạng U-Net chỉ có một bộ trọng số dùng chung, nhưng quá trình khử nhiễu ở step 1 và step 100 khác nhau hoàn toàn
- Tác giả mượn ý tưởng Positional Encoding từ Transformer. Giá trị thời gian t được chuyển thành một vector embedding, qua một lớp Linear rồi cộng dồn vào ở mọi ResNet Block



Self-Attention



- Giai đoạn Sampling, mô hình sẽ biến một nhiễu vô nghĩa thành một bức ảnh hoàn chỉnh



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  $\mathbf{x}_0$ 
```

Noise-Conditioned Score Network

- Hai nhược điểm của DDPM:
 - Thời gian sampling lớn do phải khử nhiễu từng bước một
 - Cũng một nhiễu nhưng mô hình có thể sampling ra hai ảnh khác nhau
- Noise-Conditioned Score Network (NCSN) ra đời năm 2019 với ý tưởng phá hủy dữ liệu và khôi phục dần dần thông qua việc để mạng học một scoring function

$$x_t = x_0 + \sigma_t \varepsilon, \varepsilon \sim N(0, 1)$$

$$s(x_t) \approx \frac{d}{dx} \log p(x_t)$$



Noise-Conditioned Score Network

- Việc DDPM học nhiễu và NCSN học score function về cơ bản là học cùng một thứ, chung nguyên lý toán học

$$x_t = \sqrt{\alpha_t}x_0 + \sqrt{1 - \alpha_t}\varepsilon \sim N(\sqrt{\alpha_t}x_0, \sqrt{1 - \alpha_t})$$

$$s(x_t) = \frac{d}{dx} \log p(x_t) = \frac{d}{dx} \left(-\frac{\|x_t - \sqrt{\alpha_t}x_0\|^2}{2(\sqrt{1 - \alpha_t})^2} \right)$$

$$= -\frac{x_t - \sqrt{\alpha_t}x_0}{(\sqrt{1 - \alpha_t})^2} = -\frac{\varepsilon(x_t, t)}{\sqrt{1 - \alpha_t}}$$



Denoising Diffusion Implicit Models

- Công thức tổng quát giai đoạn Sampling:

$$x_s = \sqrt{\frac{\alpha_s}{\alpha_t}}(x_t - \sqrt{1 - \alpha_t}\varepsilon(x_t)) + \sqrt{1 - \alpha_t - \sigma_{t \rightarrow s}^2}\varepsilon(x_t) + \sigma_{t \rightarrow s}z$$

- Với DDPM: $\sigma_{t \rightarrow s}^2 = \frac{1 - \alpha_s}{1 - \alpha_t} \left(1 - \frac{\alpha_t}{\alpha_s}\right)$
- Với DDIM: $\sigma_{t \rightarrow s}^2 = 0$

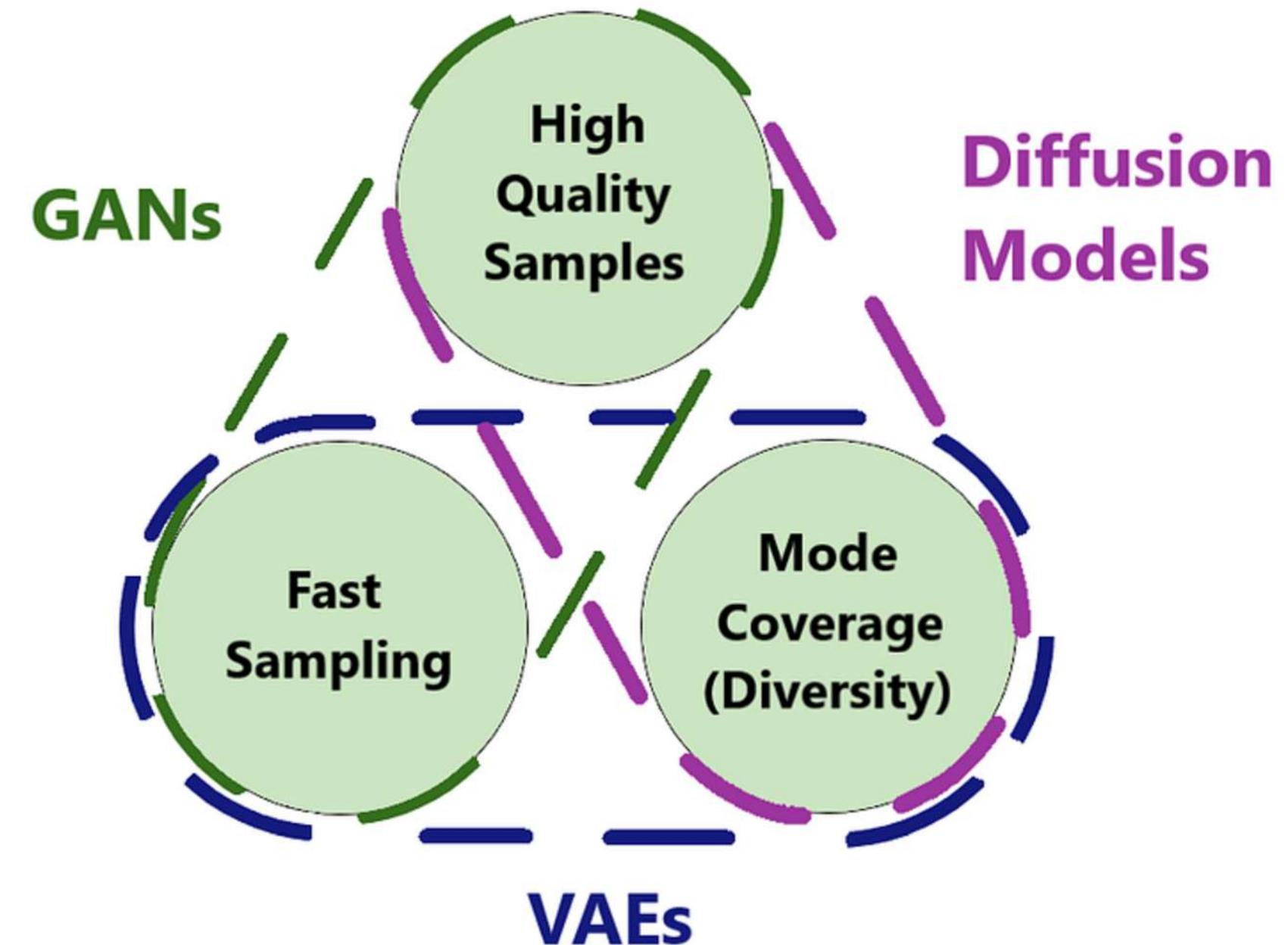


- Ưu điểm của Diffusion Models:
 - Tạo ra ảnh có độ chi tiết cực cao, sắc nét, xử lý tốt các cấu trúc phức tạp
 - Không bị "Mode Collapse" hay "Posterior Collapse"
 - Hàm mất mát đơn giản, huấn luyện dễ, khó thất bại
- Nhược điểm của Diffusion Models:
 - Các mô hình hiện đại cũng mất 10-20 bước để sinh ảnh
 - Tốn kém tài nguyên tính toán, tốn thời gian
 - Cần lượng dữ liệu khổng lồ để học được phân phối xác suất phức tạp

- Từ sự ra đời của DDPM năm 2020 đến hiện tại có hàng loạt các mô hình Diffusion khác cải tiến so với mô hình gốc:
 - Latent Diffusion Model: Nén ảnh vào một không gian bé hơn sau đó mới thực hiện quá trình thêm/khử nhiễu
 - DDIM, DPM++: Thực hiện các phép tính cho phép nhảy cóc trong quá trình sampling, giảm còn chưa tới 20 bước
 - Diffusion Transformers: Thay thế U-Net bằng Vision Transformer

Generative Learning Trilemma

- Tam giác bất khả thi: thể hiện rằng chưa có một mô hình sinh nào thực sự đạt được cả 3 tiêu chí là Nhanh - Đẹp - Đa dạng



Ứng dụng trong bài toán thực tế

- Bộ dữ liệu: FFHQ (Flickr-Faces-HQ Dataset)
- Bao gồm 52000 ảnh mặt người đa dạng giới tính, tuổi tác, màu da,... thu thập trên nền tảng Flickr - cộng đồng các nhiếp ảnh gia



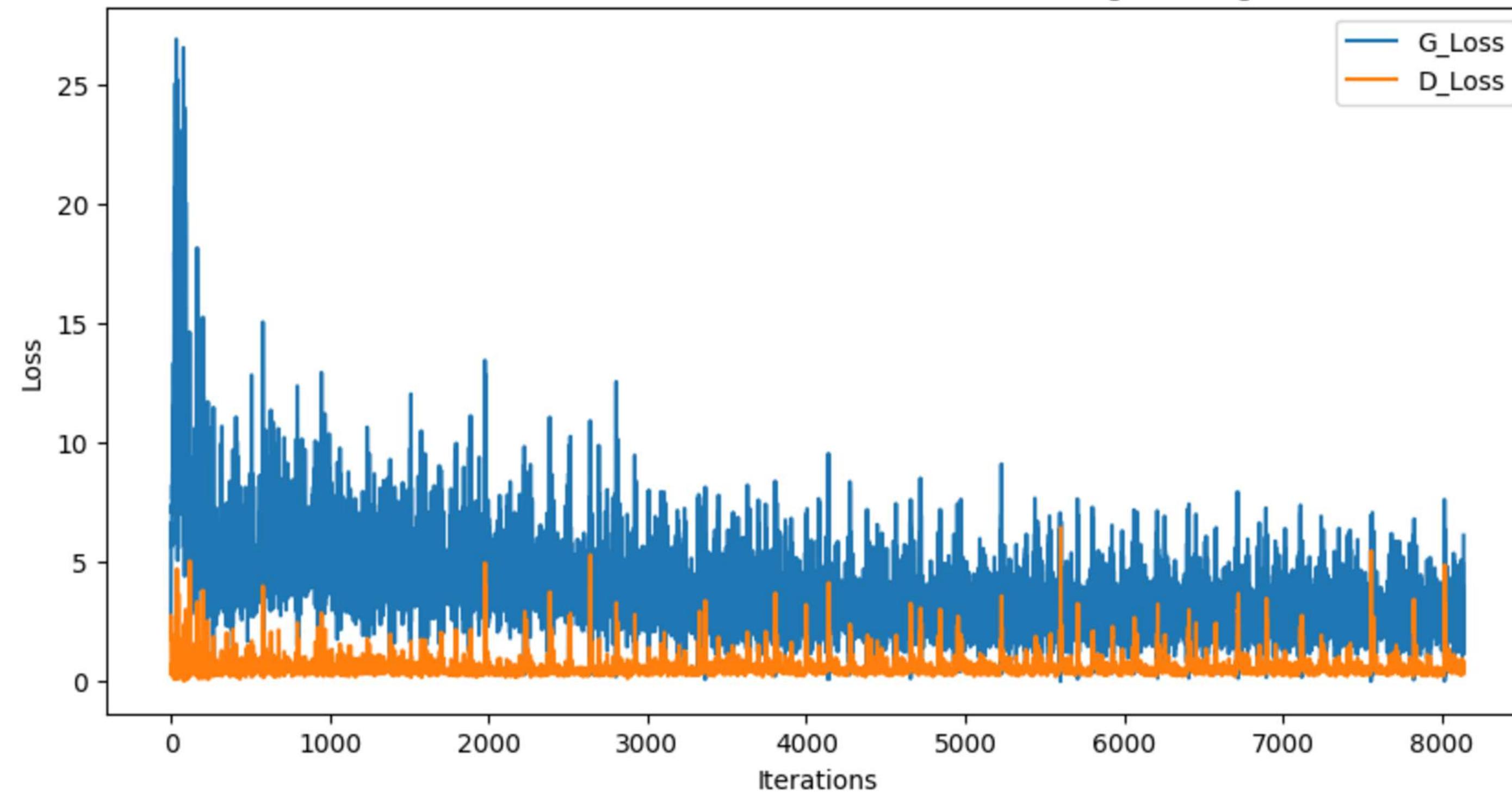
Ứng dụng trong bài toán thực tế

- Các chỉ số sử dụng để đánh giá VAE, DCGAN và DDIM là:
 - Precision: Đánh giá chất lượng ảnh tái tạo
 - Recall: Đánh giá độ đa dạng, bao phủ của ảnh tái tạo
 - FID: So sánh phân phối ảnh tái tạo và ảnh thật

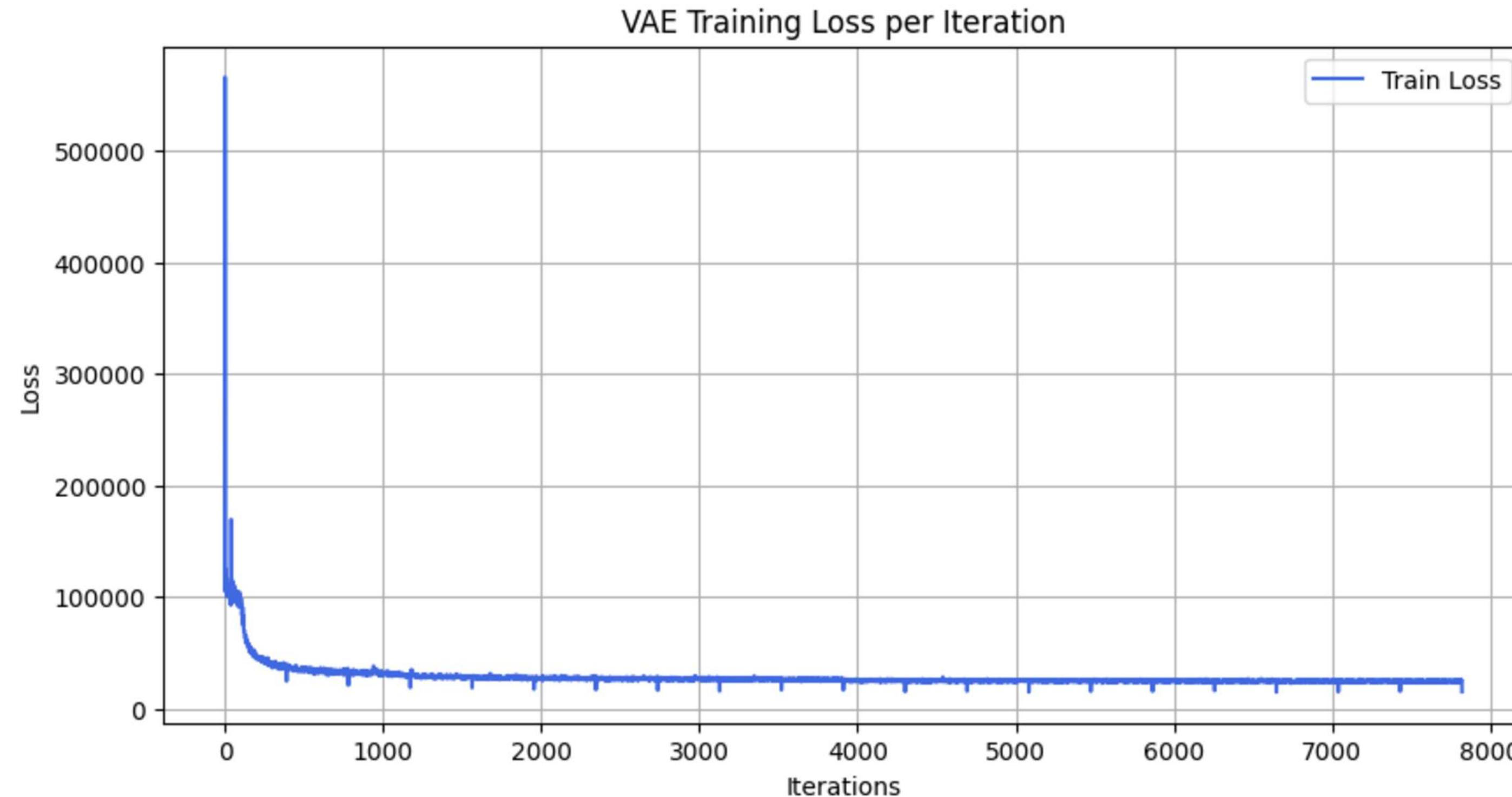


Ứng dụng trong bài toán thực tế

DCGAN Generator and Discriminator Loss During Training

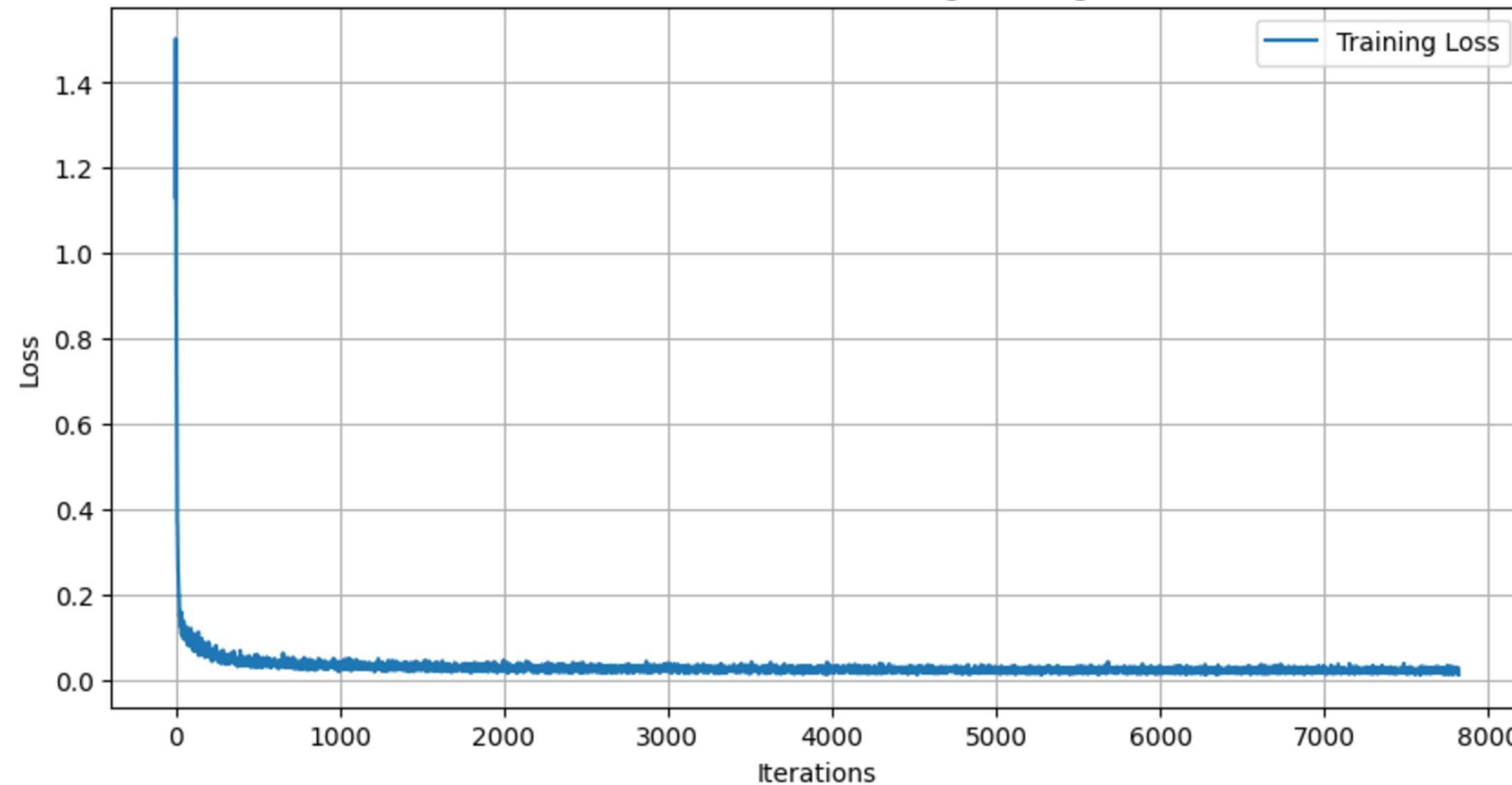


Ứng dụng trong bài toán thực tế

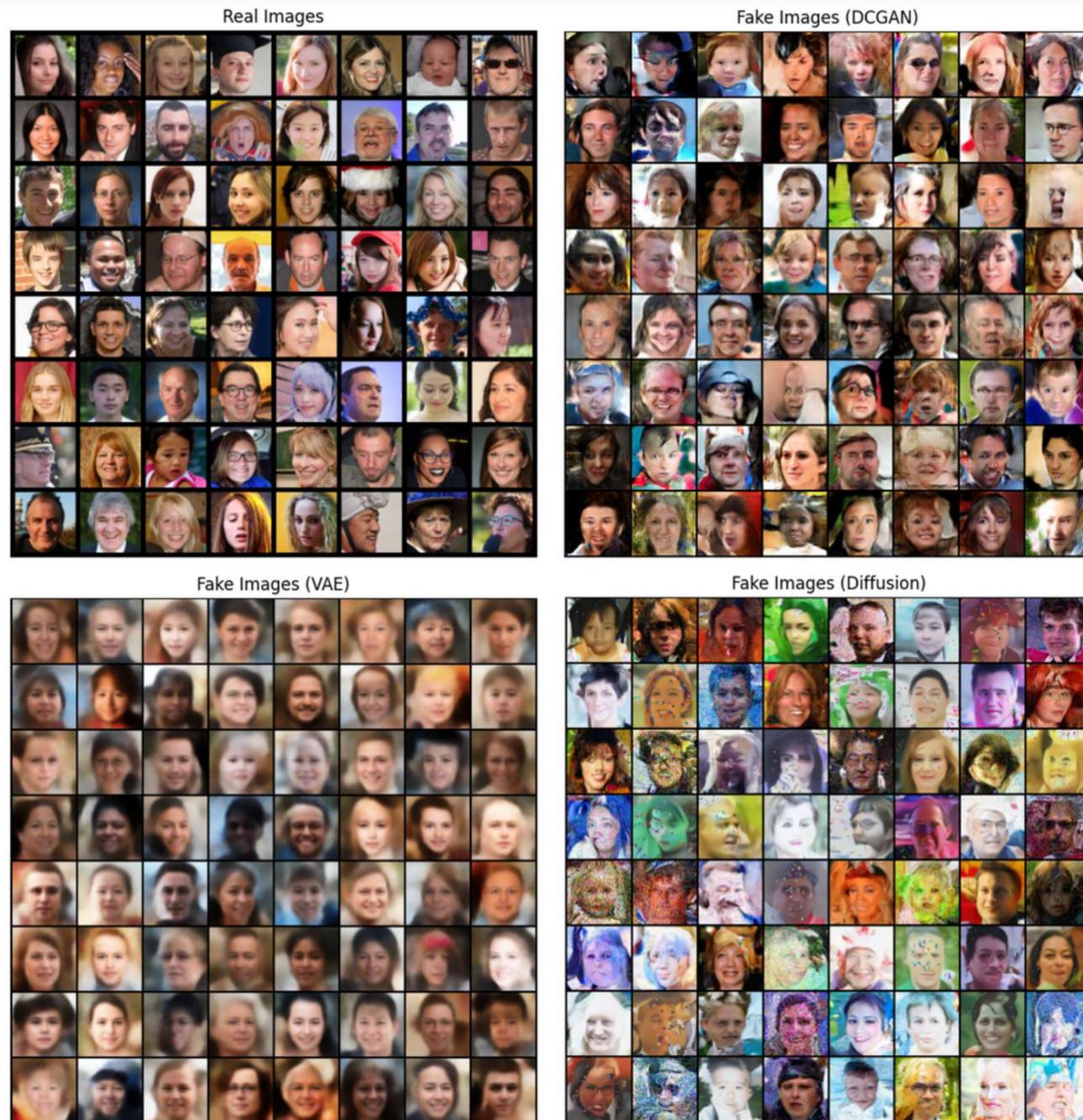


Ứng dụng trong bài toán thực tế

Diffusion Model Loss during Training



Ứng dụng trong bài toán thực tế



Ứng dụng trong bài toán thực tế

Model	Precision	Recall	FID
VAE	0.0042	0.0011	327.2013
DCGAN	0.0924	0.0192	149.5057
Diffusion	0.0540	0.2042	69.9397



Ứng dụng trong bài toán thực tế



A large, faint watermark of the HUST logo is visible across the entire background of the slide.

HUST

THANK YOU !