

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

Generative Adversarial Network

ONE LOVE. ONE FUTURE.

Nội dung chính

- GAN
 - Minimax Objective
 - Failure Models
- DCGAN
- WGAN
- CGAN
- Ứng dụng bài toán thực tế



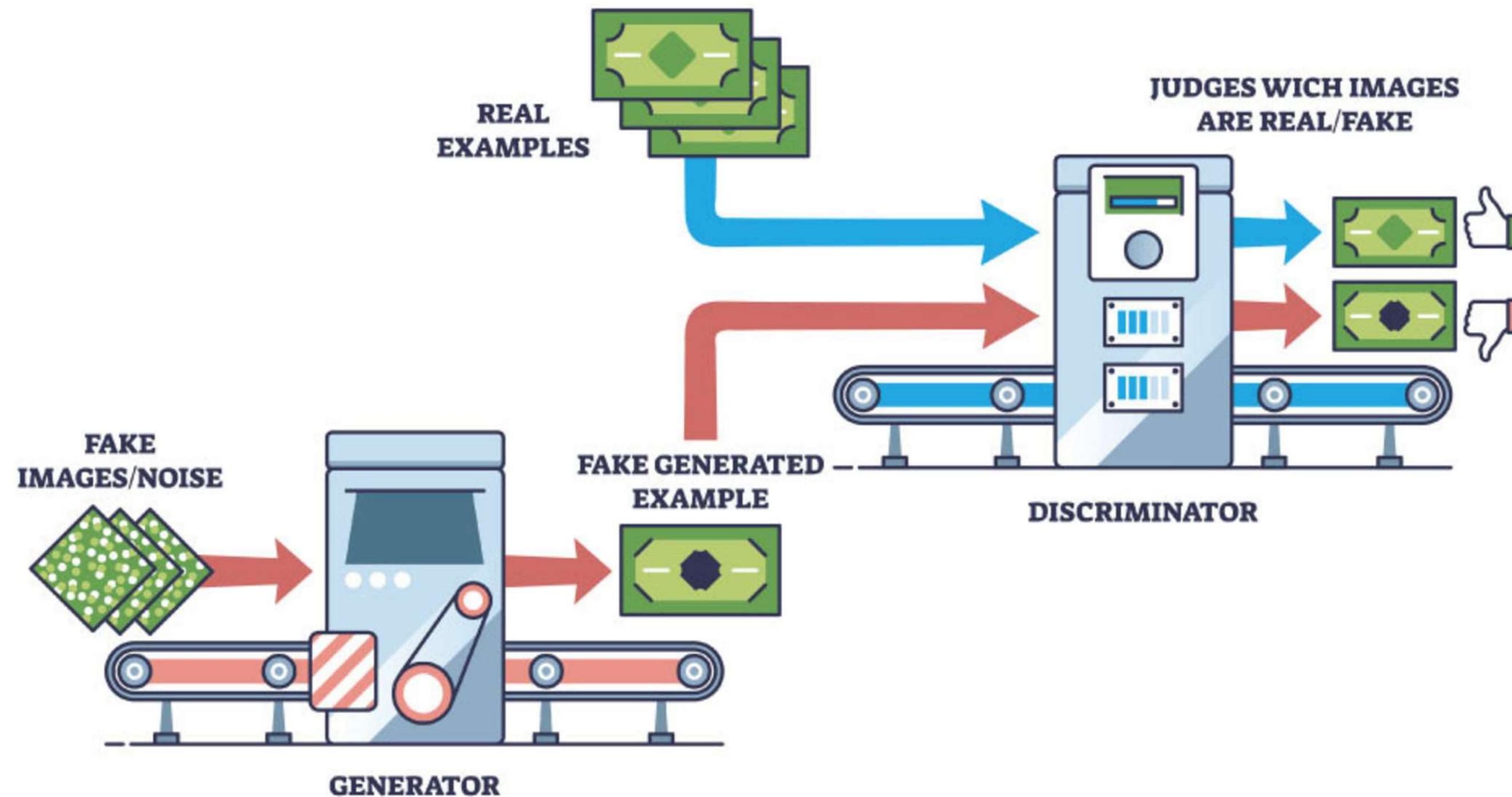
- Với một tập dữ liệu X và một tập nhãn tương ứng Y :
 - Discriminative model học mối quan hệ giữa đầu vào và nhãn bằng cách mô hình hóa xác suất có điều kiện $P(Y|X)$ từ đó phân biệt các dữ liệu với nhau
 - Generative model học phân phối xác suất của dữ liệu $P(X, Y)$ từ đó có thể tạo các dữ liệu mới

Generative Adversarial Networks

- Generative Adversarial Networks (GAN) được thiết kế để học cách sinh dữ liệu mới giống với dữ liệu thật gồm hai mạng đối kháng nhau:
 - Generator: học phân phối dữ liệu và cố gắng sinh ra các mẫu dữ liệu trông giống thật nhất có thể
 - Discriminator: nhìn vào một mẫu và quyết định xem mẫu đó đến từ tập dữ liệu thật hay do Generator tạo ra



Minimax Objective



Minimax Objective

- GAN sử dụng một hàm mất mát như sau:

$$L(G, D) = E_x[\log D(x)] + E_z[\log(1 - D(G(z)))]$$

- Cố định Generator, cập nhật Discriminator

$$\max L_D = \frac{1}{N} \sum_{i=1}^N (\log D(x_i) + \log(1 - D(G(z_i))))$$

- Cố định Discriminator, cập nhật Generator:

$$\min L_G = \frac{1}{N} \sum_{i=1}^N \log(1 - D(G(z_i))) \Leftrightarrow \max L'_G = \frac{1}{N} \sum_{i=1}^N \log D(G(z_i))$$



- **Ưu điểm của GAN:**
 - Không tối ưu trực tiếp trên dữ liệu huấn luyện vì Generator cập nhật gián tiếp thông qua Discriminator từ đó giúp Generator học phân phối dữ liệu tổng quát hơn
 - Đem lại hiệu quả tính toán cao vì Generator học ánh xạ từ không gian ngẫu nhiên sang không gian dữ liệu mà không cần mô hình hóa xác suất phức tạp nào

Failure Models

- Trên thực tế, GAN khá khó huấn luyện. Trong lý thuyết trò chơi, việc huấn luyện GAN giống như trò chơi phi hợp tác giữa hai người, khi sự phát triển của người này sẽ gây hại đến người kia
- Hiện tại không có thuật toán nào đảm bảo GAN hội tụ

Generator



Discriminator



Mode Collapse

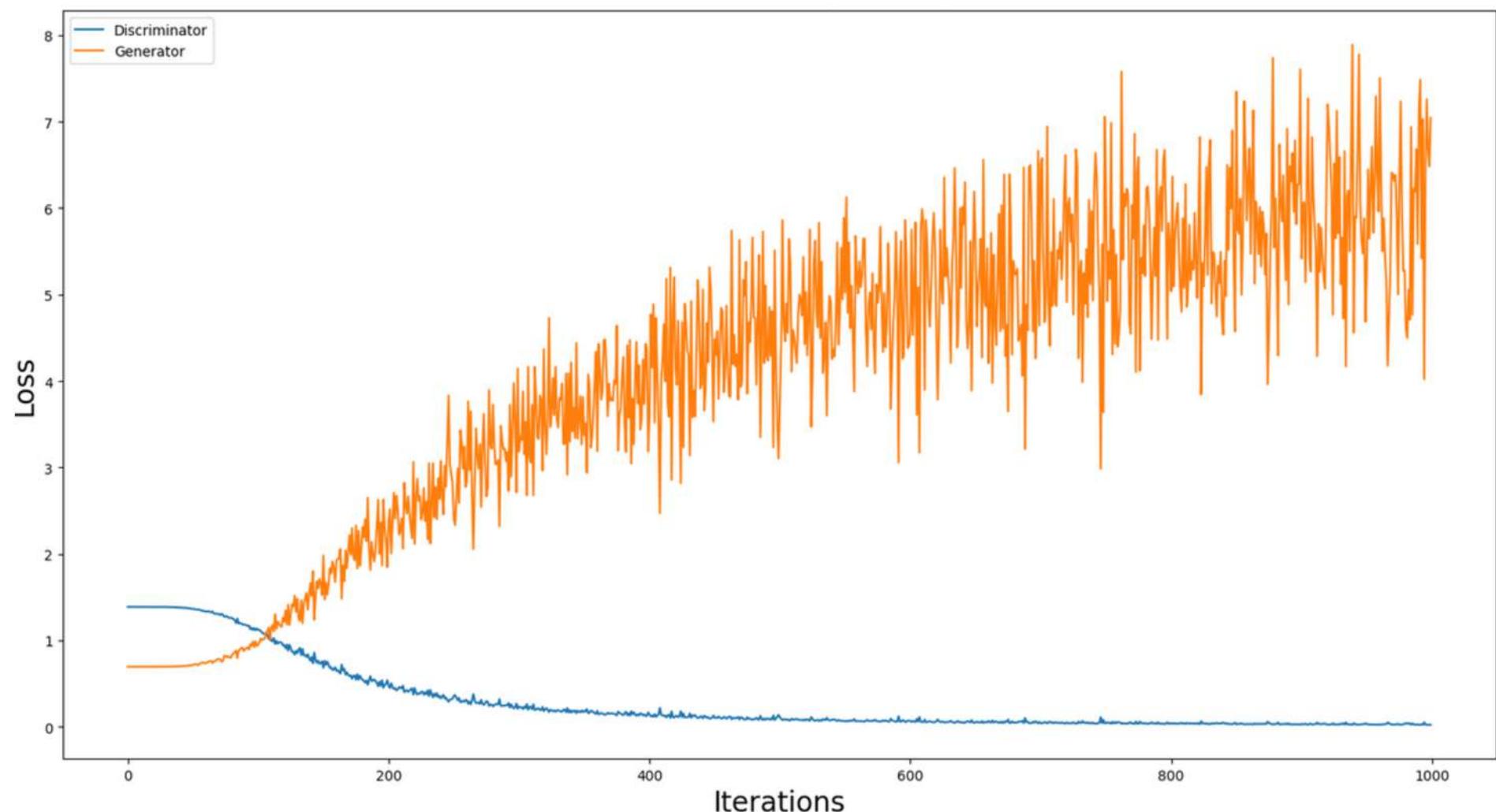
- Mode Collapse là hiện tượng Generator không sinh ra được các dữ liệu đa dạng như kỳ vọng
 - Partial collapse: Generator sinh ra một số ít kiểu ảnh khác nhau
 - Complete collapse: Tất cả ảnh Generator sinh ra giống hệt nhau



- Nguyên nhân:
 - Discriminator học quá chậm → Generator lợi dụng chỉ sinh ra số ít loại ảnh vừa đủ lừa
 - Hàm mất mát cơ bản dễ khiến Generator sinh các ảnh giống hệt nhau và cũng không có cơ chế phạt
 - Vòng luẩn quẩn: Generator chỉ sinh ảnh chó → Discriminator bị ép học thuộc coi ảnh toàn bộ chó là giả → Generator chuyển sang sinh chỉ ảnh mèo → ...
- Cách hạn chế:
 - Thay đổi hàm mất mát
 - Tăng độ phức tạp của Generator, cố tình làm suy yếu Discriminator

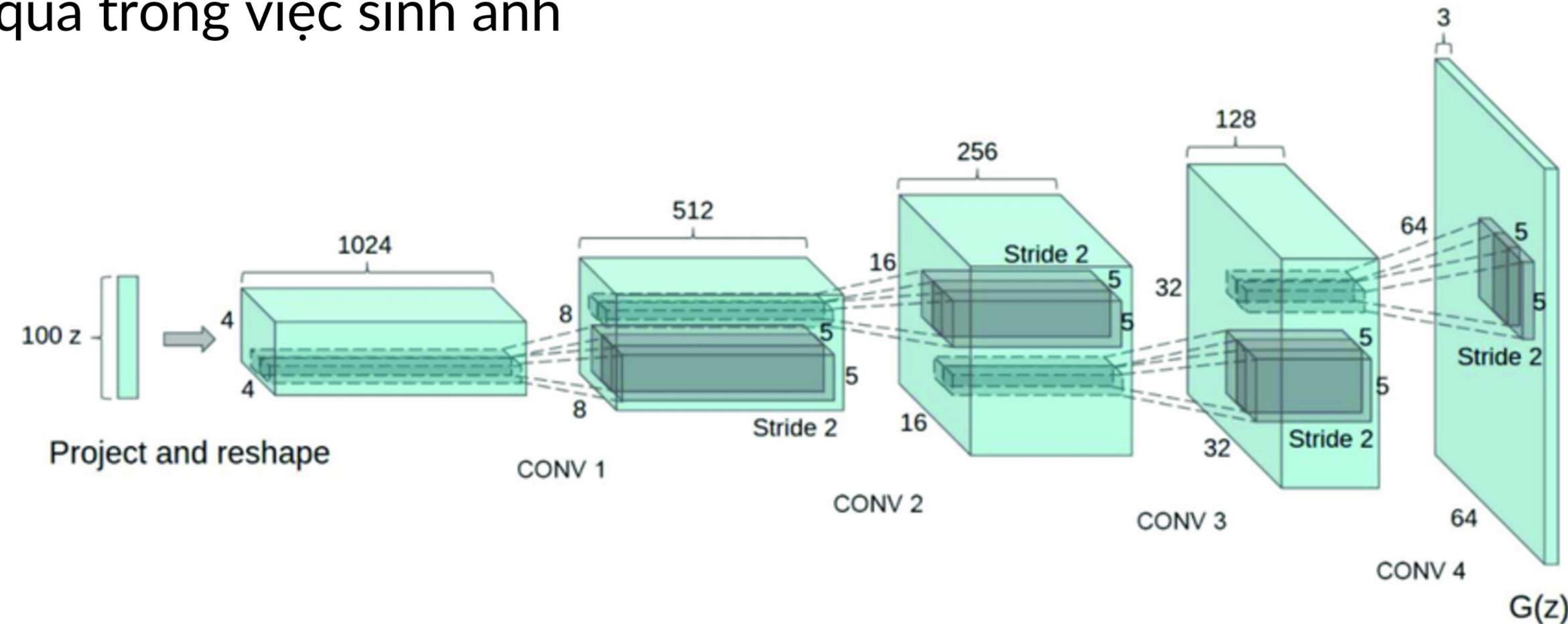
Convergence Failure

- Convergence Failure là hiện tượng khi quá trình huấn luyện GAN thất bại và không thể hội tụ do một bên chiếm ưu thế hoàn toàn
- Một số dấu hiệu nhận biết:
 - Hàm mất mát của Generator hoặc là giảm về 0 hoặc tăng liên tục
 - Các mẫu được sinh ra từ Generator có chất lượng rất kém



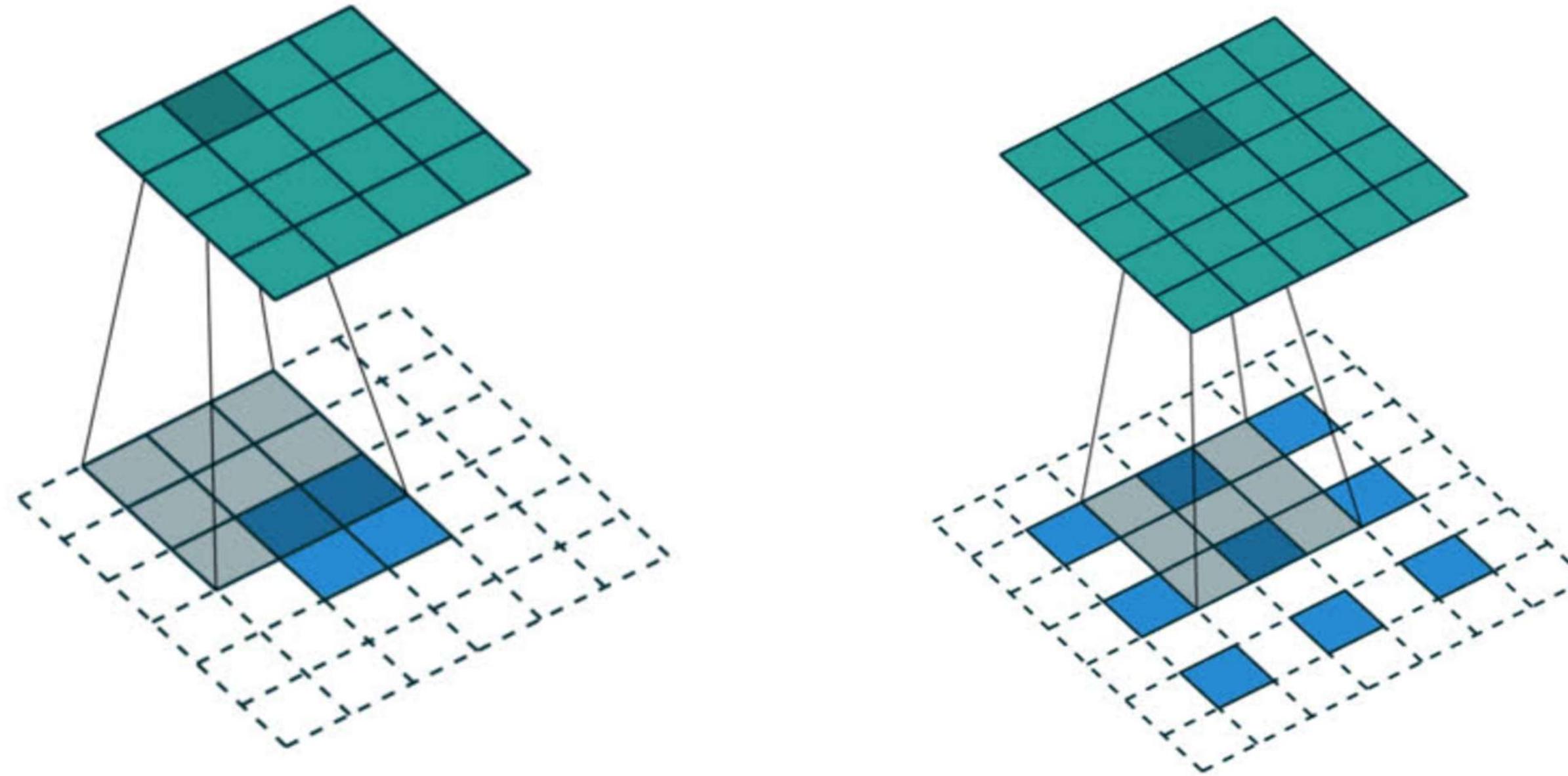
- Cách hạn chế:
 - Tăng độ phức tạp cho mạng bị áp đảo
 - Làm yếu mạng đang chiếm ưu thế bằng việc giảm bớt số tầng hoặc áp dụng dropout hay các kỹ thuật regularization
 - Cân bằng lại tốc độ học của hai mạng
 - Thay đổi tỉ lệ huấn luyện của hai mạng

- DCGAN ra đời nhằm giải quyết hạn chế của GAN truyền thống, dùng lớp Convolution thay vì MLP giúp cải thiện kết quả trong việc sinh ảnh



Transposed Convolution

- Transposed Convolution hay Deconvolution là một phép “ngược” với Convolution, dùng để tăng kích cỡ feature map



- **Ưu điểm của DCGAN:**
 - Sinh ảnh chất lượng cao hơn GAN truyền thống, có khả năng học đặc trưng không gian phong phú
 - Gradient ổn định hơn nhờ BatchNorm
- **Nhược điểm của DCGAN:**
 - Kiến trúc cứng nhắc yêu cầu kích thước ảnh cố định 64x64
 - Giúp huấn luyện ổn định và sinh ảnh tốt hơn nhưng không khắc phục các hạn chế như Mode Collapse hay Convergence Failure

- Hàm mất mát của GAN truyền thống:

$$L(G, D) = E_x[\log D(x)] + E_z [\log (1 - D(G(z)))]$$

- Cố định Generator, Discriminator tối ưu:

$$D^0 = \frac{P_{real}(x)}{P_{real}(x) + P_{generate}(x)}$$

- Thay Discriminator tối ưu:

$$\begin{aligned} L(G, D^0) &= E_{x \sim P_r} [\log \frac{P_r(x)}{P_r(x) + P_g(x)}] + E_{x \sim P_g} [\log \frac{P_g(x)}{P_r(x) + P_g(x)}] \\ &= -2\log 2 + JS(P_r || P_g) \end{aligned}$$

Wasserstein Distance

- WGAN thay hàm mất mát truyền thống thay hàm đo khoảng cách Wasserstein

$$L(G, D) = E_x[D(x)] - E_z[D(G(z))]$$

- Cố định Generator, cập nhật Discriminator:

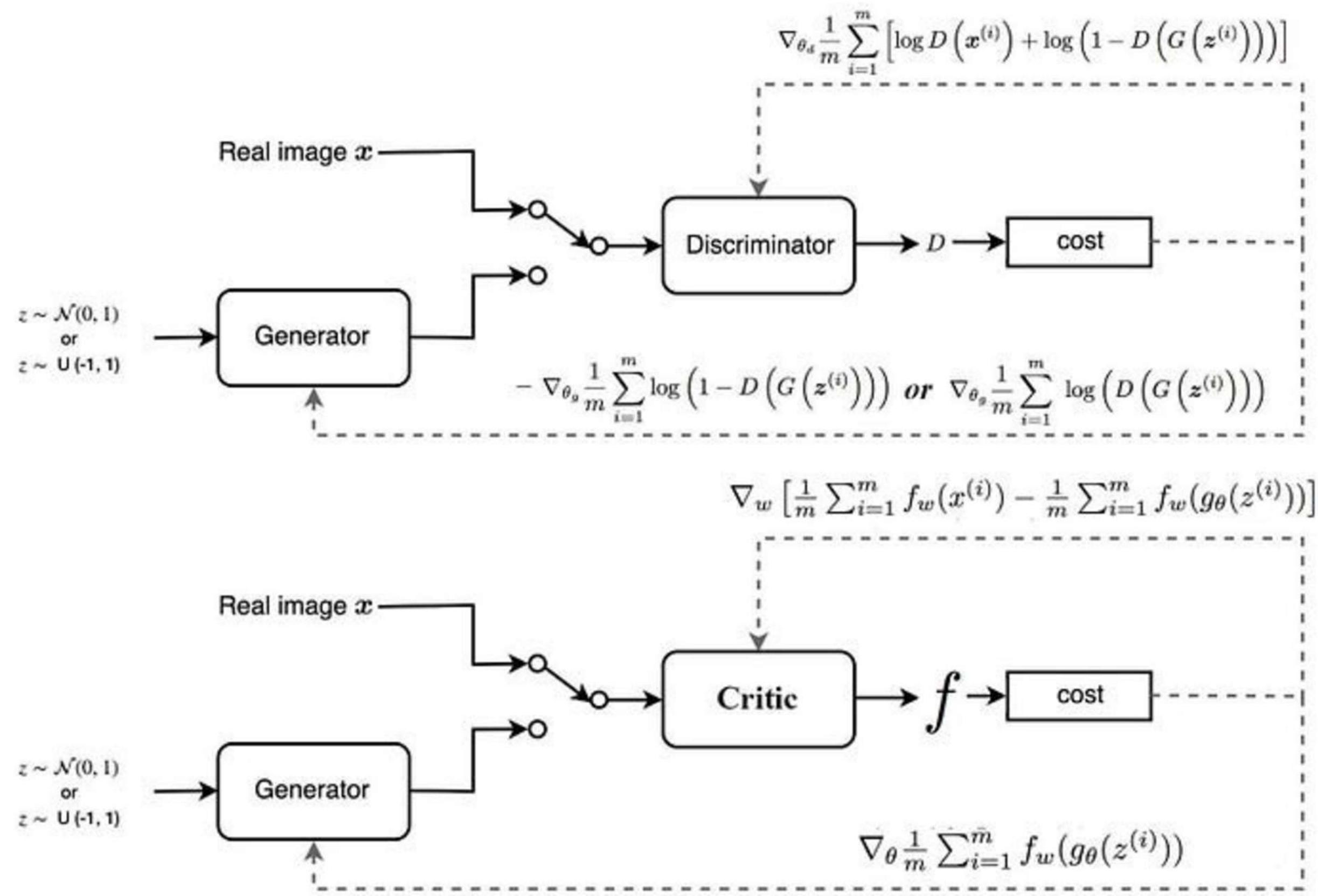
$$\max L_D = \frac{1}{N} \sum_{i=1}^N (D(x_i) - D(G(z_i)))$$

- Cố định Discriminator, cập nhật Generator:

$$\min L_G = \frac{1}{N} \sum_{i=1}^N D(G(z_i))$$



WGAN



- WGAN gốc giới hạn Gradient bằng cơ chế Weight Clipping, đảm bảo toàn bộ trọng số nằm trong một khoảng cố định

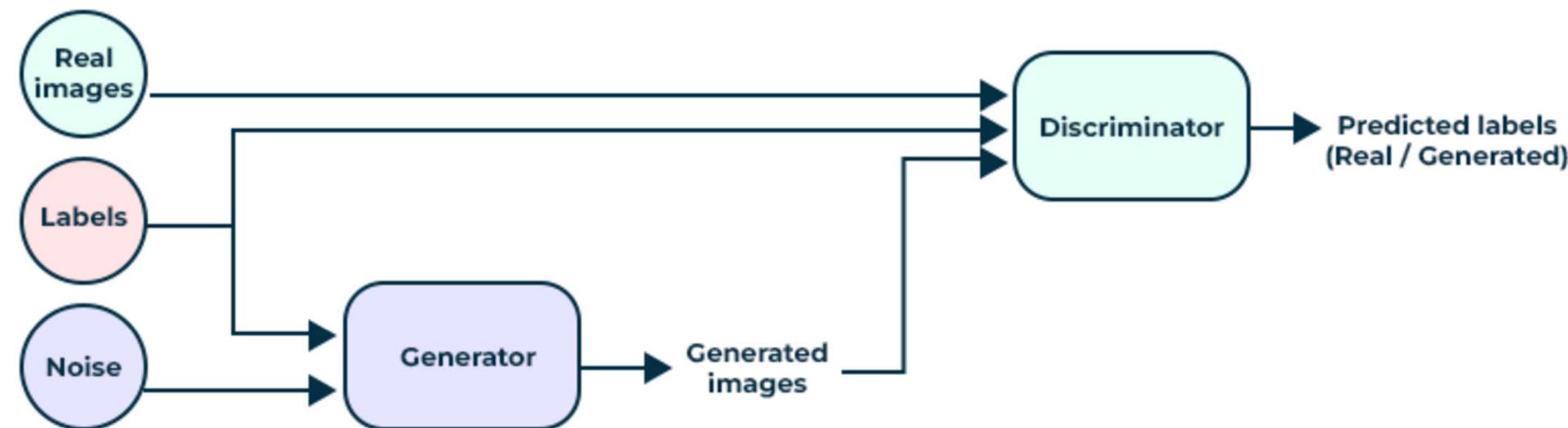
$$w_i \in [-c, c], c < 1$$

- Weight Clipping hạn chế các trọng số, làm mạng khó biểu diễn các mối quan hệ phức tạp, tham số c khó chọn cho phù hợp
- WGAN-GP xóa bỏ Weight Clipping và bổ sung vào hàm mất mát của Critic (Discriminator) một hàm gọi là Gradient Penalty

$$GP = \lambda E_x \left[\left(\left\| \frac{dD(x)}{dx} \right\|_2 - a \right)^2 \right]$$

- Ưu điểm của WGAN:
 - Hàm mất mát có ý nghĩa, huấn luyện ổn định hơn GAN truyền thống
 - Hạn chế hiện tượng Mode Collapse
- Nhược điểm của WGAN:
 - Để gradient cho Generator chính xác, Critic phải train nhiều bước, tốn thời gian, lâu hơn GAN truyền thống
 - Nhạy với các tham số
 - Vẫn còn các hạn chế như Convergence Failure

- Conditional GAN (CGAN) là một biến thể của GAN, trong đó quá trình sinh dữ liệu được điều kiện hóa bởi một thông tin bổ sung (label, class, thuộc tính...) từ đó giúp mô hình sinh ra dữ liệu theo đúng điều kiện yêu cầu



- CGAN gốc bổ sung thông tin nhãn bằng cách biến label thành vector one-hot và concat vào vector đầu vào của Generator và Discriminator
- Hạn chế của phương pháp dùng one-hot:
 - Vector thừa không mang nhiều thông tin
 - Không phù hợp với bài toán có số nhãn lớn
 - Không thực sự tốt cho dữ liệu ảnh
 - Thông tin nhãn chỉ xuất hiện một lần, mạng dễ quên
- Phương pháp hiện đại:
 - Đưa qua một lớp embedding trước khi concat
 - Thông tin nhãn bổ sung vào mọi lớp của Generator

- Ưu điểm của CGAN:
 - Cho phép sinh ảnh theo nhãn hoặc điều kiện cụ thể
 - Khi dùng embedding thay vì one-hot, mạng học được quan hệ giữa các nhãn
 - Một số bài toán giúp hạn chế Mode Collapse
- Nhược điểm của CGAN:
 - Phụ thuộc vào nhãn, yêu cầu dữ liệu được đánh nhãn phân loại cẩn thận
 - Nhiều tham số hơn GAN, chi phí tính toán tăng
 - Vẫn còn các hạn chế như Mode Collapse hay Convergence Failure

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

- Precision và Recall là cách để đo chất lượng và sự đa dạng của mô hình generative (đặc biệt là GAN)
 - Precision: Mô hình có sinh ảnh giống ảnh thật không?
 - Recall: Mô hình có sinh được đủ sự đa dạng của dữ liệu thật không?
- Lấy N ảnh thật và N ảnh do mô hình sinh ra, sử dụng InceptionV3 chuyển 2N ảnh thành vector embedding và sử dụng KNN ($k = 3$)
 - Precision = % ảnh giả nằm trong vùng của ảnh thật
 - Recall = % ảnh thật nằm trong vùng của ảnh giả



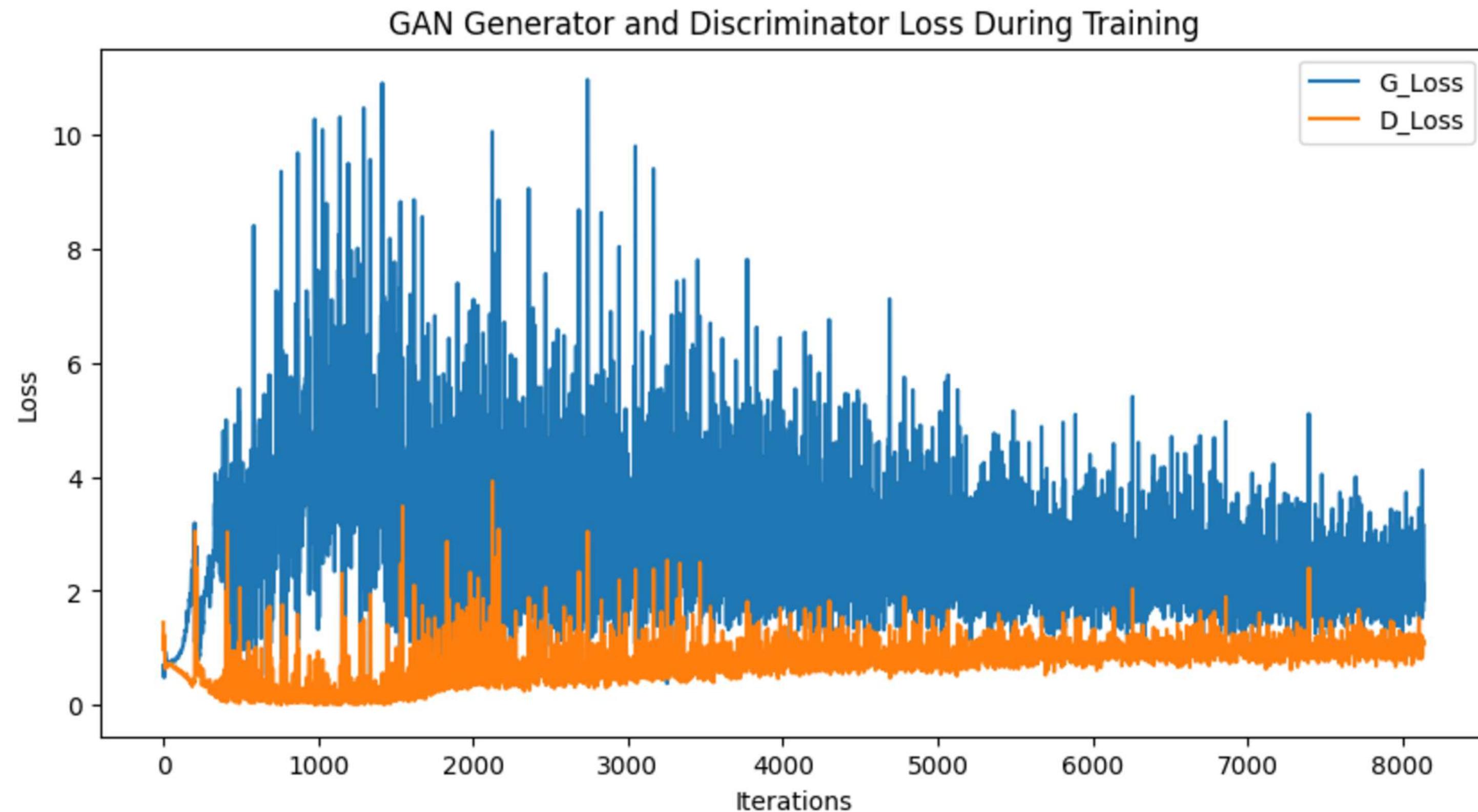
Ứng dụng trong bài toán thực tế

- FID (Frechet Inception Distance) đo độ khác nhau giữa phân phối của ảnh thật và ảnh giả
- Giả sử phân phối ảnh thật và giả đều tuân theo phân phối chuẩn, FID là công thức tính khoảng cách giữa hai phân phối → FID càng thấp càng tốt:

$$FID = \|\mu_r - \mu_f\|^2 + Trace(\sigma_r + \sigma_f - 2(\sigma_r \sigma_f)^{\frac{1}{2}})$$

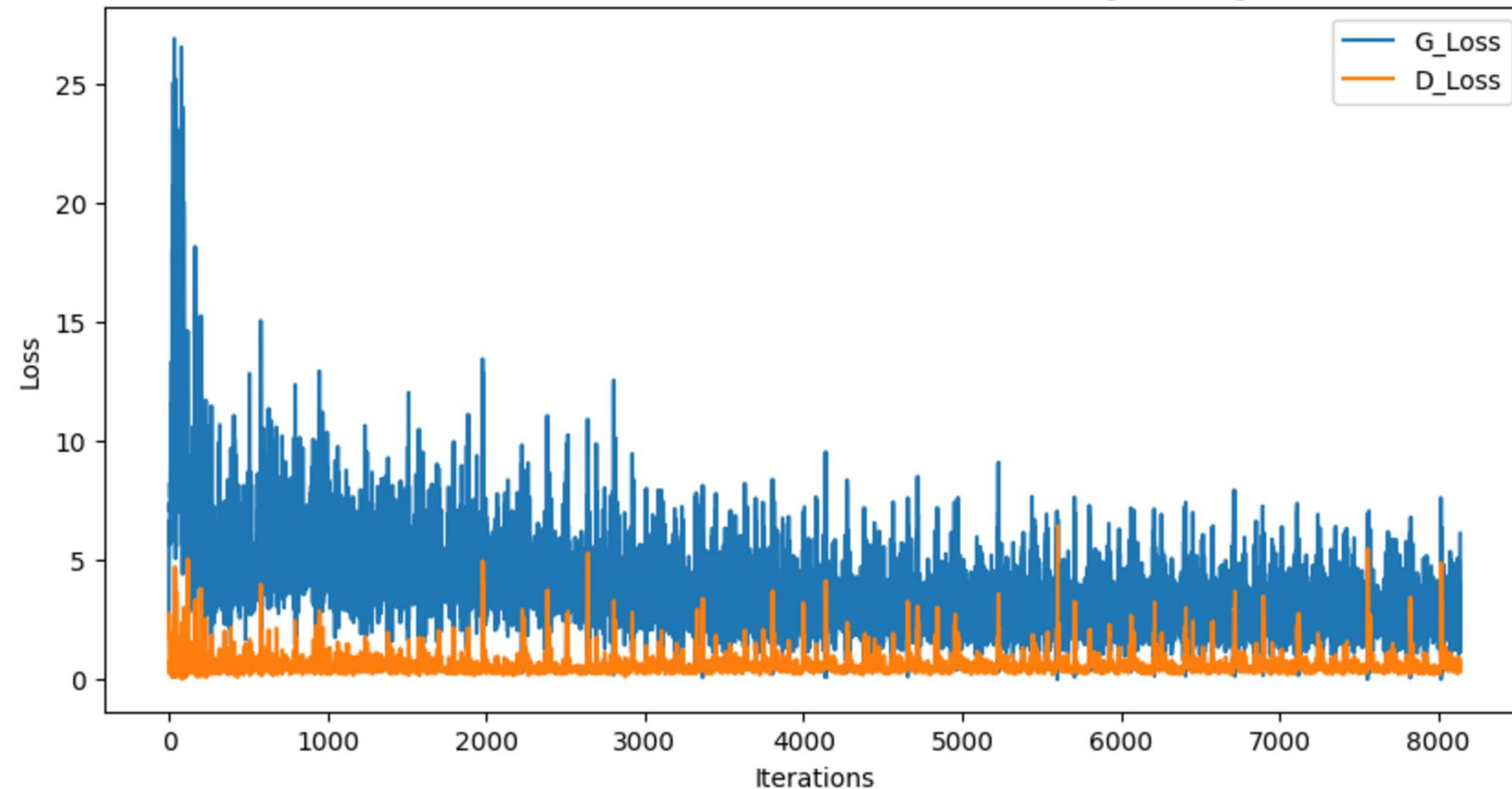


Ứng dụng trong bài toán thực tế

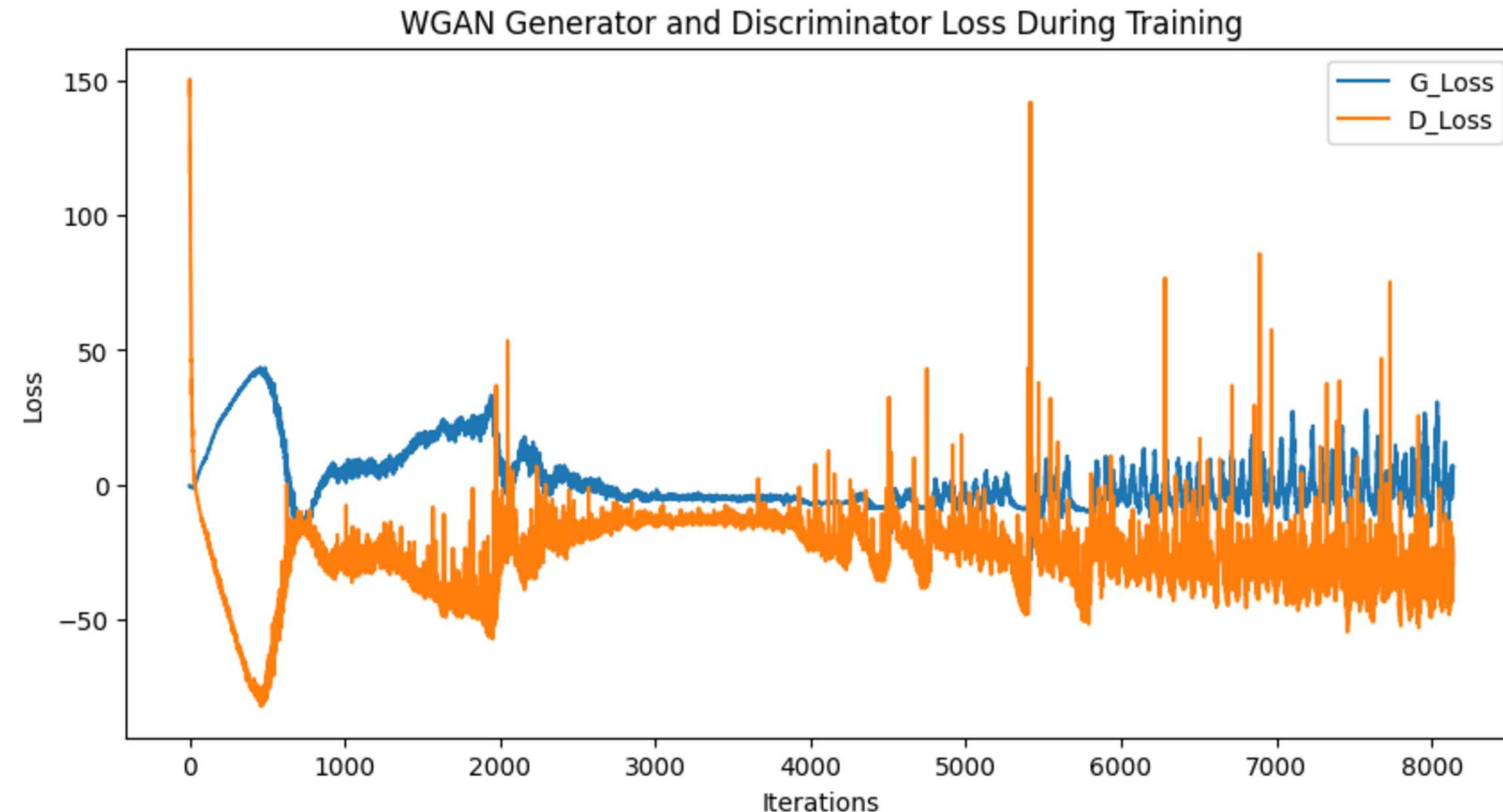


Ứng dụng trong bài toán thực tế

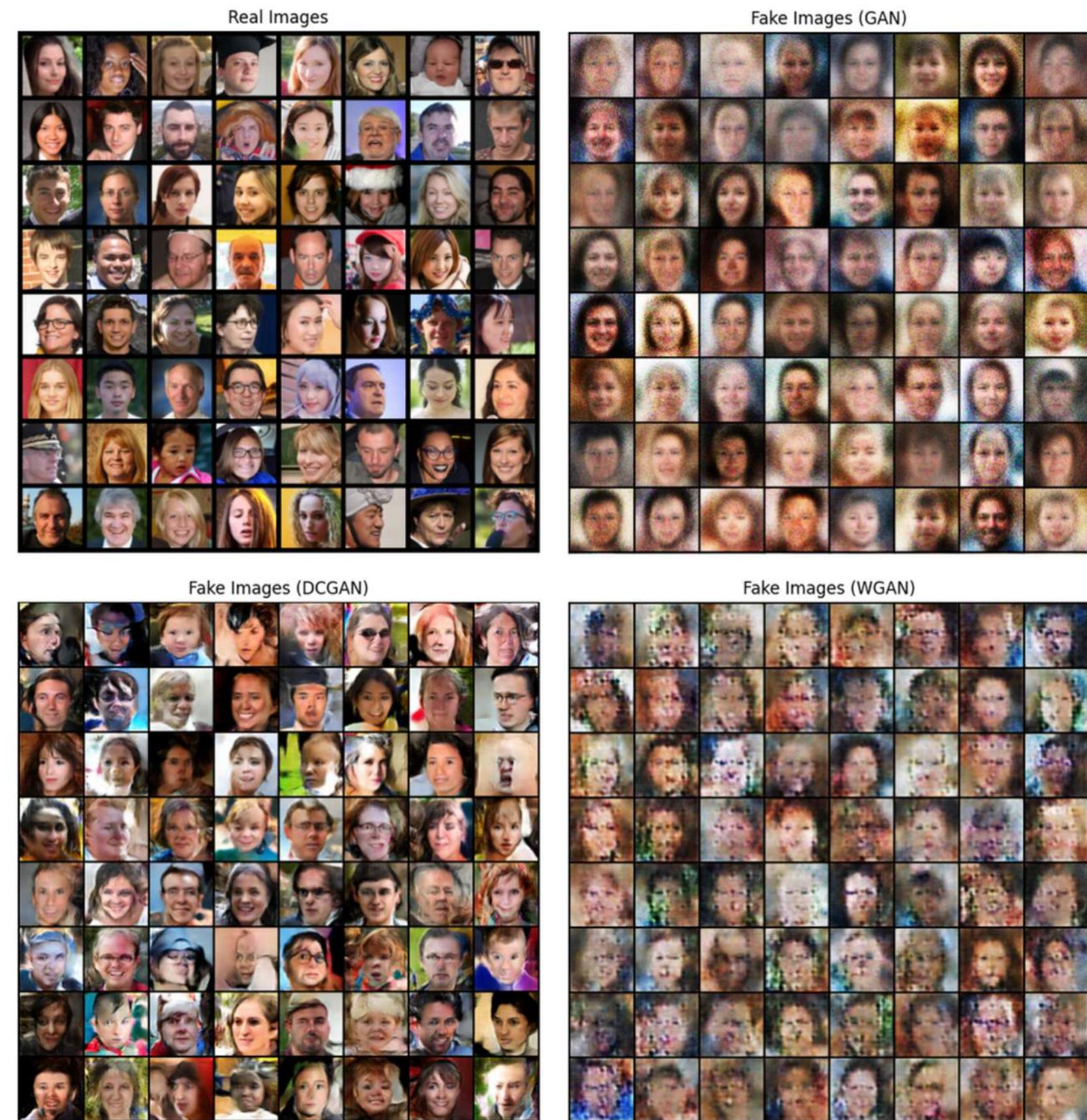
DCGAN Generator and Discriminator Loss During Training



Ứng dụng trong bài toán thực tế



Ứng dụng trong bài toán thực tế

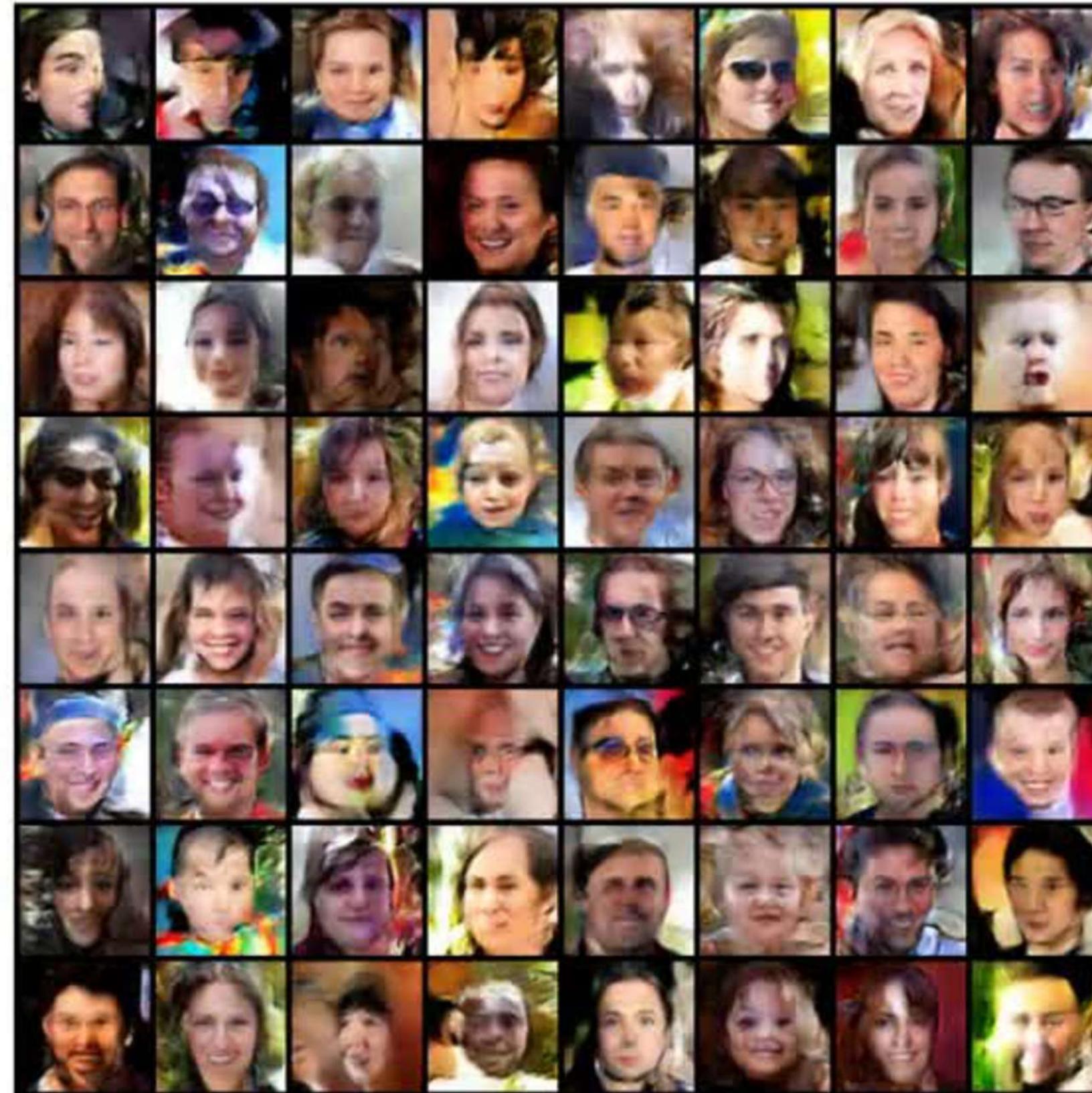


Ứng dụng trong bài toán thực tế

Model	Precision	Recall	FID
GAN	0.0104	0.0013	250.5502
DCGAN	0.0924	0.0192	149.5057
WGAN-GP	0.0088	0.0006	288.7550



Ứng dụng trong bài toán thực tế



Ứng dụng trong bài toán thực tế

- Bộ dữ liệu: Fashion MNIST
- Bao gồm 70000 ảnh kích cỡ 28x28 grayscale về quần áo, phụ kiện chia làm 10 nhãn



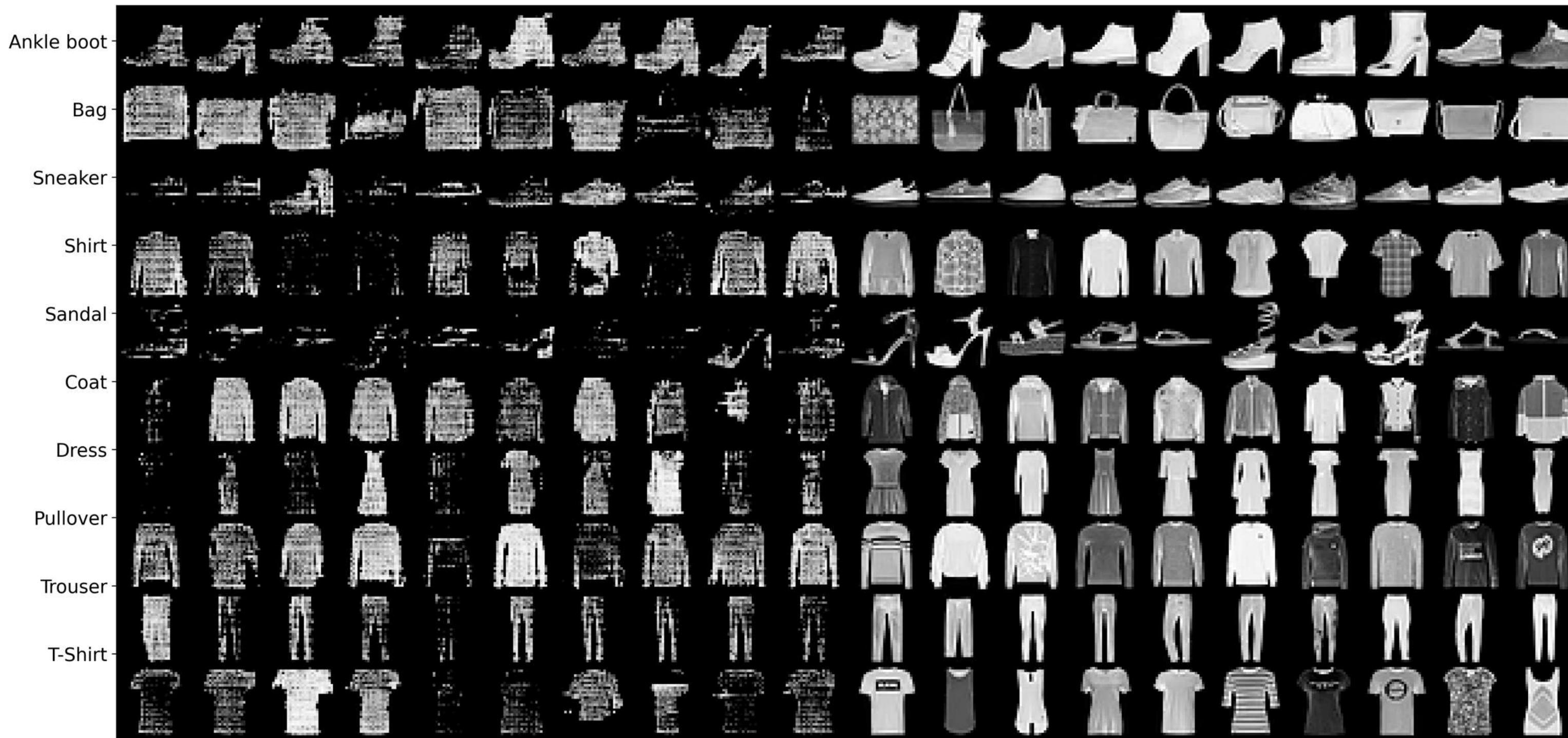
Ứng dụng trong bài toán thực tế

Precision: 0.0991 Recall: 0.0013 FID: 160.9237



Ứng dụng trong bài toán thực tế

Precision: 0.1906 Recall: 0.0088 FID: 113.3874

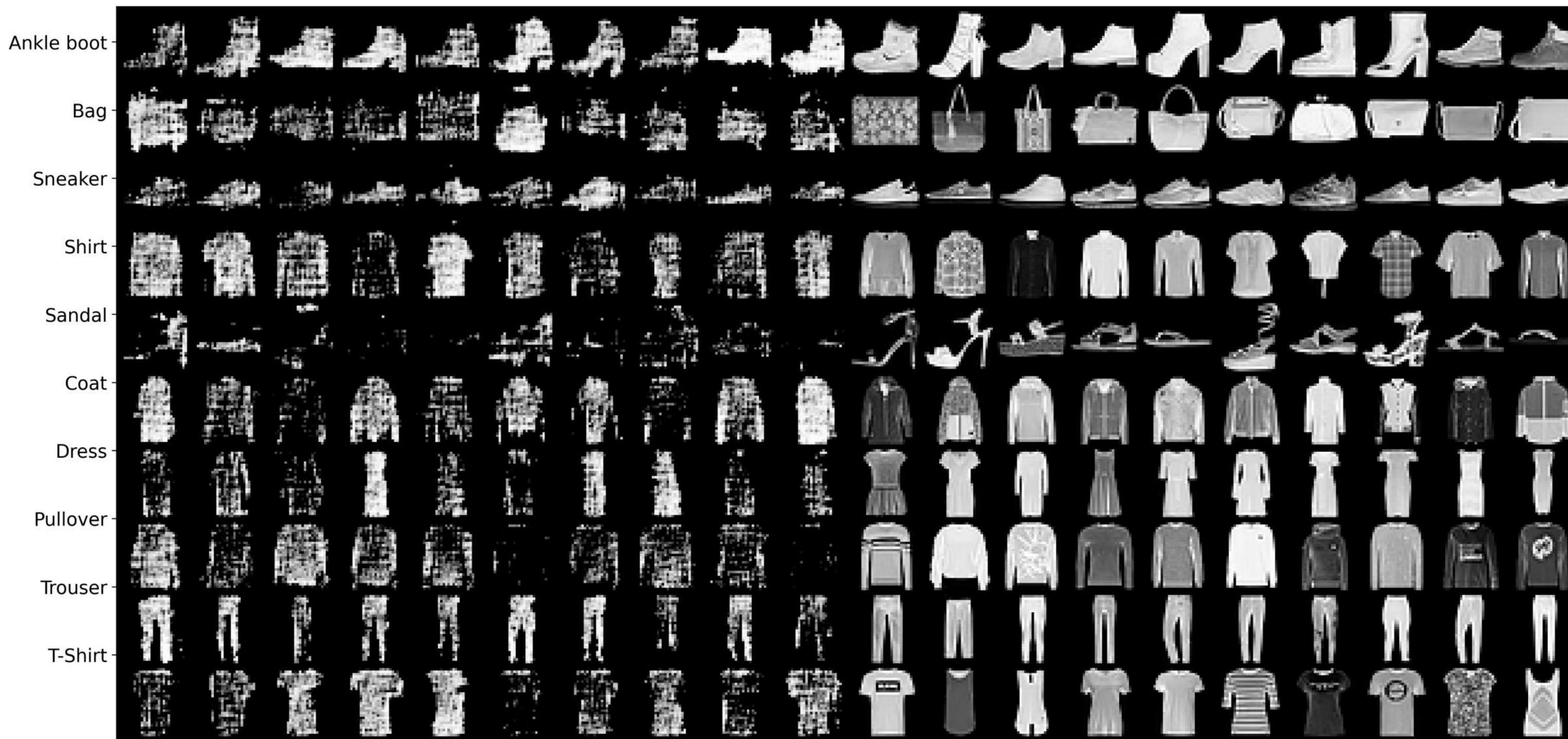


Ứng dụng trong bài toán thực tế

Precision: 0.0950

Recall: 0.0062

FID: 126.2724



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

HUST

THANK YOU !