

**ĐẠI HỌC QUỐC GIA HÀ NỘI  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**



**SO SÁNH DDPM VÀ STABLE DIFFUSION  
CHO BÀI TOÁN IMAGE COMPLETION**

**BÁO CÁO BÀI TẬP LỚN  
Học phần: AIT3001# 1**

**Sinh viên thực hiện**

Nguyễn Thị Thanh Huyền - 23020381

Nguyễn Thị Minh Ly - 23020399

Đặng Minh Nguyệt - 23020407

**Giảng viên hướng dẫn**

PGS.TS. Nguyễn Việt Hà  
ThS. Nguyễn Thị Thùy Linh

**HÀ NỘI - 2025**

## TÓM TẮT

*Từ khóa:* *Image Completion, Image Inpainting, Diffusion Models, DDPM, Stable Diffusion, LoRA*

Image Completion (Image Inpainting) là một bài toán quan trọng trong lĩnh vực thị giác máy tính và mô hình sinh, với mục tiêu khôi phục các vùng ảnh bị khuyết sao cho ảnh đầu ra vừa tự nhiên về mặt thị giác, vừa nhất quán về cấu trúc và ngữ nghĩa. Sự phát triển của các mô hình sinh dựa trên khuếch tán (diffusion models) đã mở ra những hướng tiếp cận mới hiệu quả hơn so với các phương pháp học sâu truyền thống. Trong báo cáo này, nhóm khảo sát và so sánh hiệu quả của hai hướng tiếp cận dựa trên diffusion cho bài toán Image Completion: (i) Denoising Diffusion Probabilistic Model (DDPM) được fine-tuning trực tiếp cho tác vụ, và (ii) Stable Diffusion v1.5 - một latent diffusion model đã được huấn luyện trước trên quy mô dữ liệu lớn - được tinh chỉnh bằng kỹ thuật Low-Rank Adaptation (LoRA). Các thí nghiệm được thực hiện trên bộ dữ liệu CelebA-HQ. Kết quả cho thấy Stable Diffusion v1.5 kết hợp với LoRA vượt trội hơn DDPM trong việc tái tạo chi tiết và duy trì tính nhất quán ngữ nghĩa, đồng thời giảm đáng kể chi phí huấn luyện. Báo cáo này nhấn mạnh vai trò của các mô hình diffusion pre-trained và các kỹ thuật fine-tuning hiệu quả tham số trong việc nâng cao chất lượng Image Completion, đặc biệt trong các kịch bản dữ liệu và tài nguyên hạn chế.

# Mục lục

<b>1</b>	<b>Giới thiệu</b>	<b>1</b>
1.1	Phát biểu bài toán . . . . .	1
1.2	Giới thiệu về DDPM và Stable Diffusion v1.5 . . . . .	2
1.3	Phạm vi dự án . . . . .	2
<b>2</b>	<b>Công trình nghiên cứu liên quan</b>	<b>4</b>
2.1	Các phương pháp Image Completion dựa trên mô hình học sâu . . . . .	4
2.2	Image Completion với Diffusion Models . . . . .	4
2.3	Stable Diffusion và Image Completion . . . . .	5
2.4	Fine-tuning Stable Diffusion cho tác vụ Image Completion . . . . .	5
<b>3</b>	<b>Cơ sở lý thuyết</b>	<b>6</b>
3.1	Diffusion Models . . . . .	6
3.2	Denoising Diffusion Probabilistic Models (DDPM) . . . . .	7
3.3	Stable Diffusion v1.5 . . . . .	8
3.4	Image Completion với Diffusion Models . . . . .	9
3.4.1	Quá trình khuếch tán tiến có điều kiện . . . . .	9
3.4.2	Quá trình khuếch tán ngược có điều kiện . . . . .	9
3.4.3	Suy diễn và khôi phục ảnh . . . . .	10
<b>4</b>	<b>Thực nghiệm và Kết quả</b>	<b>11</b>
4.1	Dataset & Metrics đánh giá . . . . .	11
4.1.1	Dataset và Tiền xử lý dữ liệu . . . . .	11
4.1.2	Metrics đánh giá . . . . .	11
4.2	Thực nghiệm với mô hình DDPM . . . . .	12
4.2.1	Chi tiết thực thi và Kiến trúc mô hình . . . . .	12
4.2.2	Training Objective & Spatially Weighted Loss . . . . .	13
4.3	Stable Diffusion v1.5 fine-tuned bằng LoRA . . . . .	13

4.3.1	Cấu hình thí nghiệm . . . . .	13
4.3.2	Pipeline mô hình . . . . .	14
4.3.3	Tiền xử lý dữ liệu . . . . .	15
4.3.4	Huấn luyện . . . . .	15
4.4	Kết quả . . . . .	16
4.4.1	Kết quả trực quan . . . . .	16
4.4.2	Kết quả đánh giá chỉ số . . . . .	18
<b>5</b>	<b>Kết luận</b>	<b>19</b>
5.1	Kết luận . . . . .	19
5.2	Hướng nghiên cứu trong tương lai . . . . .	19
<b>Tài liệu tham khảo</b>		<b>21</b>
<b>Phụ lục</b>		<b>22</b>

# Danh sách hình vẽ

1.1	Mô phỏng tác vụ Image Completion . . . . .	1
3.1	Quá trình khuếch tán xuôi và ngược . . . . .	6
3.2	Kiến trúc mô hình Stable Diffusion gốc . . . . .	8
4.2	Các kết quả inpainting từ 4 input test khác nhau. Mỗi ảnh hiển thị output AI tương ứng. . . . .	17

# Danh sách bảng

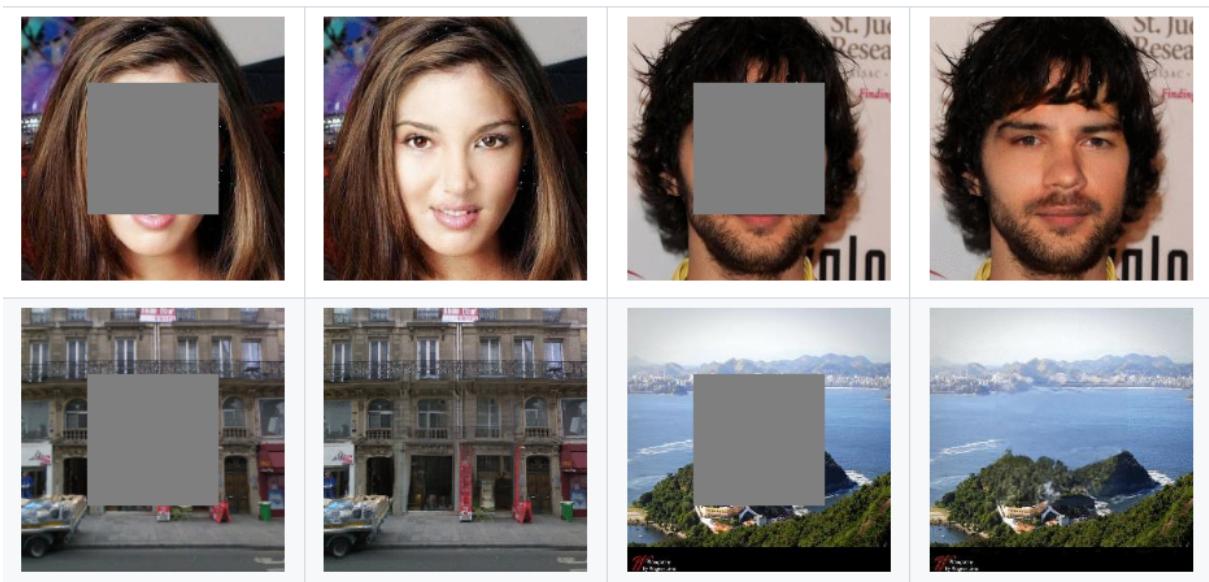
4.1 Các chỉ số đánh giá chất lượng ảnh inpainting. . . . .	18
5.1 Phân công công việc của các thành viên trong nhóm . . . . .	23

# Chương 1

## Giới thiệu

### 1.1 Phát biểu bài toán

Trong những năm gần đây, *Image Completion* (hay *Image Inpainting*) đã nổi lên như một bài toán then chốt trong lĩnh vực thị giác máy tính và các mô hình sinh. Bài toán đặt ra là: với một ảnh đầu vào bị khuyết một hoặc nhiều vùng thông tin, mô hình cần khôi phục các vùng bị thiếu sao cho ảnh đầu ra vừa nhất quán về mặt cấu trúc, vừa tự nhiên và hợp lý về mặt thị giác. Khác với các bài toán khôi phục ảnh truyền thống như khử nhiễu hay siêu phân giải, *Image Completion* không chỉ dựa vào thông tin cục bộ mà còn đòi hỏi mô hình phải nắm bắt được ngữ cảnh toàn cục, từ đó suy luận được hình dạng, cấu trúc và nội dung phù hợp cho các vùng bị che khuất.



**Hình 1.1.** Mô phỏng tác vụ *Image Completion*

*Image Completion* có nhiều ứng dụng thực tiễn quan trọng, bao gồm phục hồi ảnh cũ hoặc ảnh bị hư hỏng, chỉnh sửa và hậu kỳ ảnh, xóa vật thể không mong muốn, cũng như hỗ trợ sáng tạo nội dung trong thiết kế và nghệ thuật số. Cùng với sự phát triển mạnh

mẽ của các mô hình sinh dựa trên học sâu, đặc biệt là các mô hình diffusion, bài toán Image Completion đã vượt ra khỏi phạm vi “lắp đầy” các vùng trống đơn thuần, hướng tới việc tạo sinh nội dung có chất lượng thẩm mỹ cao và phù hợp với ngữ nghĩa tổng thể của ảnh. Do đó, việc nghiên cứu và đánh giá hiệu quả của các mô hình diffusion trong bài toán Image Completion mang ý nghĩa quan trọng cả về mặt học thuật lẫn ứng dụng thực tế.

## 1.2 Giới thiệu về DDPM và Stable Diffusion v1.5

Denoising Diffusion Probabilistic Models (DDPM) là một trong những nền tảng quan trọng đặt nền móng cho sự phát triển của các mô hình diffusion hiện đại. Nhờ cơ chế huấn luyện ổn định và khả năng mô hình hóa các phân phối dữ liệu có độ phức tạp cao, DDPM đã đạt được nhiều kết quả ấn tượng trong các tác vụ sinh ảnh. Trong phạm vi dự án này, DDPM được lựa chọn làm mô hình baseline, đóng vai trò là mốc tham chiếu nhằm đánh giá mức độ cải thiện của các phương pháp tiên tiến hơn trên cùng bài toán Image Completion.

Stable Diffusion v1.5, được giới thiệu vào năm 2022, là một mô hình diffusion tiềm ẩn (*Latent Diffusion Model*) được huấn luyện trên quy mô dữ liệu lớn. Khác với DDPM truyền thống hoạt động trực tiếp trên không gian pixel, Stable Diffusion thực hiện quá trình khuếch tán trong không gian tiềm ẩn do mô hình tự học được, từ đó giúp giảm đáng kể chi phí tính toán trong khi vẫn duy trì chất lượng sinh ảnh cao. Phiên bản Stable Diffusion v1.5 cho thấy sự cải thiện rõ rệt về chất lượng ảnh, độ ổn định và khả năng nắm bắt ngữ nghĩa so với các phiên bản trước, đặc biệt phù hợp cho các tác vụ chỉnh sửa ảnh như Image Completion. Việc lựa chọn Stable Diffusion v1.5 trong dự án này xuất phát từ hai lý do chính: (i) đây là một mô hình phổ biến và có tính đại diện cao trong cộng đồng nghiên cứu và ứng dụng; (ii) mô hình hỗ trợ hiệu quả các kỹ thuật tinh chỉnh tham số như LoRA, cho phép thích nghi với tác vụ cụ thể mà không cần huấn luyện lại toàn bộ mô hình.

## 1.3 Phạm vi dự án

Trong phạm vi của dự án, nhóm tập trung nghiên cứu và so sánh tác vụ Image Completion với hai cấu hình mô hình chính:

1. DDPM được fine-tuning trực tiếp cho bài toán Image Completion;
2. Stable Diffusion v1.5 được fine-tuning bằng kỹ thuật LoRA (Low-Rank Adaptation).

Bộ dữ liệu được sử dụng trong dự án là CelebA-HQ, bao gồm 30.000 ảnh khuôn mặt người có độ phân giải cao. Trong đó, 5.000 ảnh được dùng cho quá trình tinh chỉnh mô hình, và 500 ảnh độc lập được sử dụng cho giai đoạn đánh giá cuối cùng.

Các mô hình được so sánh dựa trên cả hai khía cạnh định tính và định lượng, bao gồm chất lượng thị giác của ảnh sinh ra, mức độ nhất quán ngữ nghĩa của vùng được khôi phục, cũng như các chỉ số đánh giá phù hợp. Thông qua đó, dự án hướng tới việc làm rõ:

(i) hiệu quả của các mô hình diffusion khác nhau trong bài toán Image Completion, và  
(ii) vai trò của các kỹ thuật fine-tuning hiệu quả tham số như LoRA trong việc cải thiện  
chất lượng ảnh đầu ra.

## Chương 2

# Công trình nghiên cứu liên quan

Trong những năm gần đây, sự phát triển mạnh mẽ của các mô hình sinh ảnh dựa trên khuếch tán (diffusion models) đã tạo ra những thay đổi đáng kể trong cách tiếp cận bài toán *Image Completion*. Từ các mô hình học sâu truyền thống dựa trên mạng tích chập và mạng sinh đối kháng, nghiên cứu hiện nay dần chuyển sang các mô hình sinh xác suất có khả năng mô hình hóa phân phối dữ liệu phức tạp và tái tạo nội dung ảnh một cách tự nhiên hơn. Đặc biệt, sự ra đời của các mô hình diffusion tiềm ẩn như Stable Diffusion đã mở ra hướng tiếp cận hiệu quả cho các tác vụ chỉnh sửa ảnh, bao gồm *Image Completion*.

### 2.1 Các phương pháp Image Completion dựa trên mô hình học sâu

Trước khi các mô hình khuếch tán trở nên phổ biến, phần lớn các nghiên cứu về *Image Completion* tập trung vào các mô hình học sâu dựa trên mạng nơ-ron tích chập (CNN) và mạng sinh đối kháng (GAN). Các phương pháp tiêu biểu như Context Encoder [4] hay các mô hình GAN có điều kiện [8] được huấn luyện nhằm tái tạo vùng ảnh bị khuyết dựa trên thông tin ngữ cảnh xung quanh. Một số nghiên cứu sau đó đã cải thiện chất lượng kết quả bằng cách kết hợp cơ chế attention hoặc khai thác cấu trúc đa tỉ lệ [9, 5].

Mặc dù đạt được kết quả khả quan trên các tập dữ liệu tiêu chuẩn, các phương pháp này thường gặp hạn chế về khả năng tổng quát hóa, đặc biệt khi xử lý các vùng khuyết lớn hoặc các cấu trúc ngữ nghĩa phức tạp. Ngoài ra, do bản chất huấn luyện mang tính xác định (deterministic) hoặc phụ thuộc mạnh vào discriminator, các mô hình này thường khó tạo ra nội dung đa dạng và tự nhiên trong những trường hợp thiếu thông tin nghiêm trọng.

### 2.2 Image Completion với Diffusion Models

Mô hình khuếch tán (Diffusion Models) đã chứng minh hiệu quả vượt trội trong các tác vụ sinh ảnh nhờ khả năng mô hình hóa chính xác phân phối dữ liệu thông qua quá trình khử nhiễu dần dần [1]. Trong bối cảnh *Image Completion*, các mô hình này

được sử dụng để sinh lại vùng ảnh bị khuyết bằng cách thực hiện quá trình khuếch tán có điều kiện, trong đó các vùng không bị che khuất đóng vai trò là điều kiện ràng buộc cho quá trình sinh ảnh [3].

So với các phương pháp học sâu truyền thống, diffusion models cho phép sinh ra các chi tiết sắc nét hơn và duy trì tốt hơn tính nhất quán ngữ nghĩa trên toàn bộ ảnh. Nhờ đặc tính sinh xác suất, các mô hình này cũng có khả năng tạo ra nhiều nghiệm hợp lý khác nhau cho cùng một vùng khuyết, phản ánh tốt hơn tính đa dạng của dữ liệu thực tế.

### 2.3 Stable Diffusion và Image Completion

Stable Diffusion [6] là một mô hình diffusion tiềm ẩn (Latent Diffusion Model), trong đó quá trình khuếch tán được thực hiện trong không gian tiềm ẩn do một mô hình autoencoder học được, thay vì trực tiếp trên không gian pixel. Cách tiếp cận này giúp giảm đáng kể chi phí tính toán và cho phép áp dụng diffusion models cho ảnh có độ phân giải cao.

Nhờ khả năng nắm bắt ngữ nghĩa mạnh mẽ và tái tạo chi tiết tốt, Stable Diffusion nhanh chóng được mở rộng cho các tác vụ chỉnh sửa ảnh, bao gồm Image Completion. Một số nghiên cứu và hệ thống gần đây đã khai thác Stable Diffusion cho inpainting có điều kiện, cho phép kiểm soát nội dung sinh ra thông qua mặt nạ vùng khuyết và các tín hiệu điều kiện khác như văn bản hoặc ảnh tham chiếu.

### 2.4 Fine-tuning Stable Diffusion cho tác vụ Image Completion

Thay vì huấn luyện mô hình từ đầu, nhiều nghiên cứu gần đây tập trung vào việc fine-tuning các mô hình diffusion đã được huấn luyện trước, đặc biệt là Stable Diffusion, nhằm thích nghi với tác vụ Image Completion hoặc các miền dữ liệu cụ thể. Cách tiếp cận này giúp tận dụng tri thức thị giác–ngữ nghĩa đã học được từ tập dữ liệu quy mô lớn, đồng thời giảm đáng kể chi phí huấn luyện.

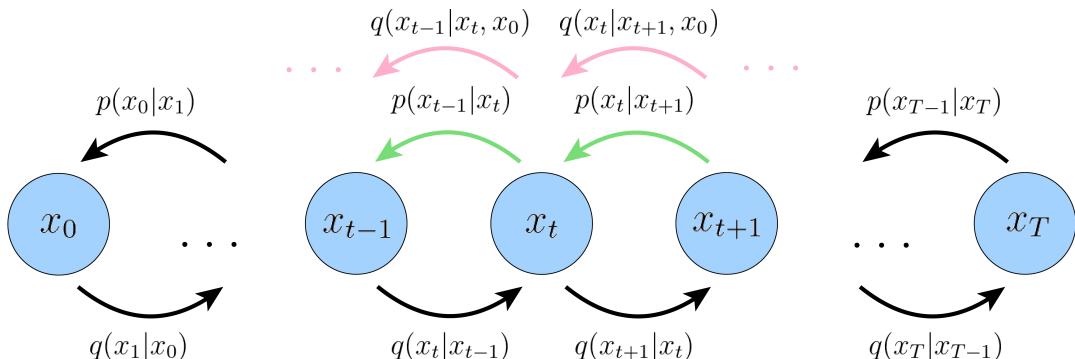
Bên cạnh fine-tuning toàn bộ mô hình, các kỹ thuật tinh chỉnh hiệu quả tham số (parameter-efficient fine-tuning) như LoRA [2] hay DreamBooth [7] đã được đề xuất nhằm chỉ cập nhật một phần nhỏ tham số của mô hình. Các phương pháp này cho phép mô hình học được phong cách hoặc cấu trúc đặc thù của dữ liệu huấn luyện mà không làm suy giảm khả năng sinh ảnh tổng quát. Nhờ đó, chúng đặc biệt phù hợp cho bài toán Image Completion trong các kịch bản có dữ liệu hạn chế hoặc yêu cầu thích nghi nhanh với miền ứng dụng mới.

# Chương 3

## Cơ sở lý thuyết

### 3.1 Diffusion Models

Diffusion Models là lớp mô hình sinh xác suất, trong đó dữ liệu được sinh ra thông qua việc học quá trình nghịch đảo của một quá trình khuếch tán ngẫu nhiên. Về mặt toán học, quá trình này được xây dựng dựa trên chuỗi Markov, gồm hai giai đoạn: quá trình khuếch tán tiến (forward process) và quá trình khuếch tán ngược (reverse process).



Hình 3.1. Quá trình khuếch tán xuôi và ngược

Quá trình khuếch tán tiến dần dần làm nhiễu dữ liệu gốc bằng cách cộng nhiễu Gaussian theo từng bước thời gian. Gọi  $x_0$  là dữ liệu ban đầu, chuỗi biến ngẫu nhiên  $\{x_t\}_{t=1}^T$  được định nghĩa sao cho:

$$q(x_t | x_{t-1}) = \mathcal{N}(x_t; \sqrt{1 - \beta_t}x_{t-1}, \beta_t \mathbf{I}), \quad (3.1)$$

trong đó  $\beta_t \in (0, 1)$  là hệ số nhiễu tại bước  $t$ .

Mục tiêu của mô hình là học quá trình khuếch tán ngược  $p_\theta(x_{t-1} | x_t)$ , cho phép tái tạo dữ liệu gốc từ nhiễu Gaussian. Do phân phối ngược không có dạng đóng, nó được xấp xỉ bằng một mô hình tham số hóa, thường là mạng nơ-ron sâu:

$$p_\theta(x_{t-1} | x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)). \quad (3.2)$$

Việc huấn luyện Diffusion Models có thể được hiểu như quá trình tối ưu hóa một cận dưới biến phân (Variational Lower Bound – VLB) của log-likelihood dữ liệu, trong đó mô hình học cách đảo ngược quá trình nhiễu hóa một cách xấp xỉ.

$$\mathbb{E}[-\log p_\theta(x_0)] \leq \mathbb{E}_q \left[ -\log \frac{p_\theta(x_{0:T})}{q(x_{1:T} | x_0)} \right] =: \mathcal{L}. \quad (3.3)$$

Khai triển biểu thức trên thu được:

$$\mathcal{L} = \mathbb{E}_q \left[ -\log p(x_T) - \sum_{t=1}^T \log \frac{p_\theta(x_{t-1} | x_t)}{q(x_t | x_{t-1})} \right]. \quad (3.4)$$

## 3.2 Denoising Diffusion Probabilistic Models (DDPM)

Denoising Diffusion Probabilistic Models (DDPM) là một hiện thực hóa cụ thể và hiệu quả của Diffusion Models, trong đó quá trình khuếch tán tiến được thiết kế sao cho có dạng đóng, giúp đơn giản hóa đáng kể việc huấn luyện và suy diễn.

Trong DDPM, nhờ tính chất của phân phối Gaussian, phân phối  $q(x_t | x_0)$  có thể được biểu diễn trực tiếp mà không cần lặp qua các bước trung gian:

$$q(x_t | x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1 - \bar{\alpha}_t)\mathbf{I}), \quad (3.5)$$

trong đó  $\alpha_t = 1 - \beta_t$  và  $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$ .

Từ đó, mẫu  $x_t$  có thể được lấy trực tiếp bằng:

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0, \mathbf{I}). \quad (3.6)$$

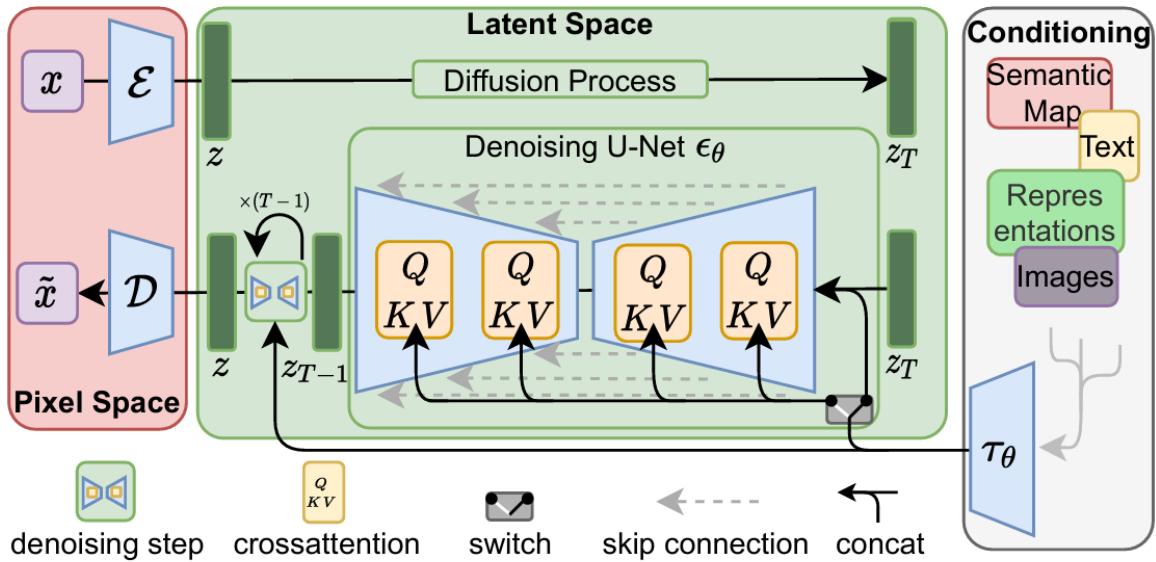
DDPM giả định phân phối khuếch tán ngược cũng có dạng Gaussian với phương sai cố định:

$$p_\theta(x_{t-1} | x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \beta_t\mathbf{I}). \quad (3.7)$$

Thay vì học trực tiếp  $\mu_\theta$ , DDPM tái tham số hóa bài toán bằng cách huấn luyện mạng nơ-ron để dự đoán nhiễu  $\epsilon$  đã được thêm vào ở quá trình khuếch tán tiến. Hàm mất mát khi đó được đơn giản hóa thành:

$$\mathcal{L}_{\text{DDPM}} = \mathbb{E}_{x_0, \epsilon, t} [\|\epsilon - \epsilon_\theta(x_t, t)\|^2]. \quad (3.8)$$

Cách tiếp cận này giúp việc huấn luyện trở nên ổn định hơn, đồng thời cho phép mô hình học được quá trình khử nhiễu hiệu quả thông qua bài toán hồi quy đơn giản.



**Hình 3.2.** Kiến trúc mô hình Stable Diffusion gốc

### 3.3 Stable Diffusion v1.5

Các nguyên lý toán học của DDPM cũng chính là nền tảng cho Stable Diffusion. Điểm khác biệt chính nằm ở việc quá trình khuếch tán được thực hiện trong không gian tiềm ẩn (Latent Diffusion Models - LDM) thay vì không gian ảnh gốc, giúp giảm chi phí tính toán trong khi vẫn giữ được chất lượng sinh ảnh cao.

Cụ thể, ảnh đầu vào  $x \in \mathbb{R}^{H \times W \times C}$  được ánh xạ vào không gian tiềm ẩn thông qua một bộ mã hóa tự động biến phân (Variational Autoencoder – VAE):

$$z_0 = \mathcal{E}(x), \quad (3.9)$$

trong đó  $\mathcal{E}$  là bộ mã hóa của VAE và  $z_0 \in \mathbb{R}^{h \times w \times c}$  là biểu diễn tiềm ẩn tương ứng. Quá trình sinh ảnh được thực hiện bằng cách áp dụng mô hình khuếch tán lên  $z_0$  thay vì  $x$ , qua đó giảm độ phức tạp không gian từ bậc ảnh gốc xuống bậc tiềm ẩn.

Trong không gian tiềm ẩn, Stable Diffusion áp dụng một quá trình khuếch tán ngược có điều kiện để học phân phối  $p_\theta(z_t)$ . Mạng U-Net được sử dụng để tham số hóa quá trình khử nhiễu, trong đó đầu vào của mạng tại mỗi bước thời gian  $t$  bao gồm latent nhiều  $z_t$  và chỉ số thời gian  $t$ . Quá trình này tuân theo nguyên lý chung của diffusion models:

$$p_\theta(z_{t-1} | z_t) = \mathcal{N}(z_{t-1}; \mu_\theta(z_t, t), \Sigma_\theta(z_t, t)). \quad (3.10)$$

Một đặc điểm quan trọng của Stable Diffusion v1.5 là khả năng điều kiện hóa quá trình sinh ảnh thông qua văn bản. Biểu diễn văn bản được trích xuất bằng mô hình CLIP Text Encoder và được tích hợp vào U-Net thông qua cơ chế cross-attention, cho phép mô hình liên kết thông tin ngôn ngữ với cấu trúc không gian trong ảnh. Nhờ đó, mô hình có thể sinh ảnh phù hợp với mô tả ngữ nghĩa đầu vào.

Sau khi hoàn tất quá trình khử nhiễu trong không gian tiềm ẩn, biểu diễn  $z_0$  được

đưa qua bộ giải mã của VAE để tái tạo ảnh đầu ra:

$$\hat{x} = \mathcal{D}(z_0), \quad (3.11)$$

trong đó  $\mathcal{D}$  là bộ giải mã VAE. Nhờ việc kết hợp khuếch tán trong không gian tiềm ẩn với cơ chế điều kiện hóa bằng ngôn ngữ, Stable Diffusion v1.5 đạt được sự cân bằng hiệu quả giữa chất lượng sinh ảnh, khả năng điều khiển ngữ nghĩa và chi phí tính toán.

### 3.4 Image Completion với Diffusion Models

Trong bài toán Image Completion (Image Inpainting), mục tiêu là khôi phục các vùng ảnh bị thiếu dựa trên phần ảnh quan sát được. Gọi  $x_0 \in \mathbb{R}^{H \times W \times C}$  là ảnh gốc và  $m \in \{0, 1\}^{H \times W}$  là mặt nạ nhị phân, trong đó  $m = 1$  biểu thị các vùng đã biết và  $m = 0$  biểu thị các vùng cần khôi phục. Khi đó, ảnh quan sát được được biểu diễn bởi:

$$x_0^{\text{obs}} = m \odot x_0. \quad (3.12)$$

#### 3.4.1 Quá trình khuếch tán tiên có điều kiện

Diffusion Models xây dựng một quá trình khuếch tán tiên bằng cách thêm nhiễu Gaussian dần dần vào dữ liệu. Trong bối cảnh Image Completion, quá trình này được điều chỉnh để chỉ áp dụng lên các vùng bị che. Cụ thể, tại bước thời gian  $t$ , biến ngẫu nhiên  $x_t$  được xác định như sau:

$$x_t = m \odot x_0 + (1 - m) \odot \tilde{x}_t, \quad (3.13)$$

trong đó

$$\tilde{x}_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, \quad \epsilon \sim \mathcal{N}(0, I), \quad (3.14)$$

và  $\bar{\alpha}_t$  là hệ số suy giảm biên độ của tín hiệu tại bước  $t$ , được xác định bởi lịch trình nhiễu của mô hình.

Cách xây dựng này đảm bảo rằng các vùng ảnh đã biết không bị nhiễu hóa trong suốt quá trình khuếch tán tiên, trong khi các vùng cần khôi phục được làm nhiễu dần theo phân phối Gaussian.

#### 3.4.2 Quá trình khuếch tán ngược có điều kiện

Quá trình khuếch tán ngược nhằm sinh lại dữ liệu gốc thông qua việc loại bỏ nhiễu từng bước. Trong Image Completion, mô hình học phân phối có điều kiện:

$$p_{\theta}(x_{t-1} \mid x_t, x_0^{\text{obs}}), \quad (3.15)$$

trong đó việc sinh mẫu chỉ được thực hiện trên các vùng bị che.

Tại mỗi bước suy diễn, mô hình dự đoán nhiễu hoặc trực tiếp ước lượng giá trị trung bình của phân phối khử nhiễu. Một mẫu trung gian  $\hat{x}_{t-1}$  được sinh ra, sau đó các

giá trị tại vùng đã biết được gán lại từ ảnh gốc:

$$x_{t-1} = m \odot x_0 + (1 - m) \odot \hat{x}_{t-1}. \quad (3.16)$$

### 3.4.3 Suy diễn và khôi phục ảnh

Quá trình suy diễn bắt đầu từ trạng thái  $x_T$ , trong đó các vùng bị che được khởi tạo bằng nhiều Gaussian, còn các vùng đã biết giữ nguyên giá trị ban đầu. Thông qua việc lặp lại quá trình khử nhiễu có điều kiện từ  $t = T$  đến  $t = 0$ , mô hình thu được ảnh hoàn chỉnh  $\hat{x}_0$ , trong đó các vùng được sinh mới có tính nhất quán cao về mặt cấu trúc và ngữ nghĩa với phần ảnh đã quan sát.

Cách tiếp cận này cho phép Diffusion Models thực hiện Image Completion như một bài toán lấy mẫu hậu nghiệm có điều kiện, tận dụng khả năng sinh ảnh mạnh mẽ của mô hình khuếch tán mà không cần giả định cụ thể về kiến trúc hay hàm mất mát.

## Chương 4

# Thực nghiệm và Kết quả

Trong phần này, nhóm trình bày thiết lập thực nghiệm để so sánh hai hướng tinh chỉnh mô hình cho bài toán Image Completion trên tập dữ liệu khuôn mặt CelebA-HQ:

1. DDPM fine-tuned trên tác vụ Image Completion
2. Stable Diffusion 1.5 fine-tuned với LoRA (Low-Rank Adaptation)

### 4.1 Dataset & Metrics đánh giá

#### 4.1.1 Dataset và Tiền xử lý dữ liệu

Nhóm thực hiện các thử nghiệm trên bộ dữ liệu CelebA-HQ, tập dữ liệu chất lượng cao bao gồm các hình ảnh khuôn mặt người nổi tiếng. Dữ liệu đầu vào cho quá trình huấn luyện được cấu trúc thành các bộ ba:  $\{x_{gt}, x_{masked}, mask\}$ , trong đó:

- $x_{gt}$ : Ảnh gốc (ground truth).
- $x_{masked}$ : Ảnh đầu vào đã bị che khuất một phần nội dung.
- $mask$ : Mặt nạ nhị phân (binary mask), với giá trị 1 biểu thị vùng bị che - cần khôi phục và 0 biểu thị vùng đã biết.

Bộ dữ liệu được sử dụng cho thực nghiệm gồm 5000 ảnh train và 500 ảnh test.

#### 4.1.2 Metrics đánh giá

- **PSNR**: Đo mức độ khác biệt pixel-wise giữa ảnh sinh và ảnh gốc; giá trị càng cao → ảnh được khôi phục càng gần với ảnh thật về mặt pixel.
- **SSIM**: Đánh giá mức độ tương đồng về cấu trúc, độ tương phản và độ sáng; phản ánh khả năng bảo toàn cấu trúc không gian của ảnh sau khi completion.

- **LPIPS:** Đo độ khác biệt cảm nhận thị giác dựa trên đặc trưng học được từ mạng sâu; giá trị càng thấp thì ảnh sinh càng giống ảnh thật theo cảm nhận của con người.
- **FID:** Đánh giá mức độ tương đồng phân phối giữa tập ảnh sinh và ảnh thật trong không gian đặc trưng; giá trị thấp → chất lượng và tính đa dạng của ảnh sinh tốt hơn.

## 4.2 Thực nghiệm với mô hình DDPM

Trước khi đưa vào mạng nơ-ron, các ảnh đầu vào được thay đổi kích thước độ phân giải, đối với mô hình DDPM là  $256 \times 256$ . Dữ liệu ảnh ( $x_{gt}$  và  $x_{masked}$ ) được chuẩn hóa về khoảng giá trị  $[-1, 1]$  để phù hợp với phân phối đầu vào của mô hình khuếch tán.

### 4.2.1 Chi tiết thực thi và Kiến trúc mô hình

Nhóm xây dựng mô hình dựa trên kiến trúc U-Net của DDPM và sử dụng trọng số đã được huấn luyện trước từ checkpoint [google/ddpm-celebahq-256](https://google.github.io/ddpm-celebahq-256) để tận dụng các đặc trưng khuôn mặt đã học được, giúp quá trình hội tụ nhanh hơn và cải thiện độ ổn định khi huấn luyện..

- **Điều chỉnh đầu vào mô hình:** Mô hình DDPM gốc được thiết kế cho đầu vào 3 kênh (RGB). Để phù hợp với tác vụ image completion có điều kiện (conditional generation), nhóm mở rộng lớp tích chập đầu tiên để chấp nhận đầu vào 7 kênh, bao gồm:

$$\text{Input} = \text{Concat}(x_{masked}, mask, x_t)$$

Trong đó  $x_{masked}$  là ảnh đầu vào đã bị che một phần (3 kênh),  $mask$  là mặt nạ (1 kênh), và  $x_t$  là trạng thái nhiễu tại bước thời gian  $t$  (3 kênh).

- **Hyperparameters:** Quá trình huấn luyện được thực hiện với các siêu tham số như sau:
  - Resolution:  $256 \times 256$ .
  - Batch Size: 8.
  - Learning Rate:  $1 \times 10^{-5}$ , sử dụng bộ tối ưu hóa AdamW.
  - Epochs: 10.
  - Gradient Clipping: Norm tối đa là 1.0 để tránh bùng nổ gradient.
  - Mixed Precision: Nhóm sử dụng Automatic Mixed Precision (AMP) với torch.amp để giảm thiểu bộ nhớ VRAM và tăng tốc độ huấn luyện.
- **EMA:** Nhóm theo dõi trọng số trung bình trượt theo hàm mũ (Exponential Moving Average - EMA) với hệ số suy giảm  $\beta_{ema} = 0.999$  trong suốt quá trình huấn luyện để ổn định hóa mô hình. Tuy nhiên, các kết quả báo cáo và checkpoint cuối cùng sử dụng trọng số thực tế để đảm bảo tính nhất quán với chiến lược *safe copy* nhằm tránh các vấn đề tiềm ẩn khi ghi đè trọng số EMA.

#### 4.2.2 Training Objective & Spatially Weighted Loss

Để tối ưu hóa mô hình cho tác vụ hoàn thiện ảnh, chúng tôi thay đổi hàm mục tiêu Mean Squared Error (MSE) tiêu chuẩn của DDPM bằng một biến thể có trọng số theo không gian (Spatially Weighted MSE). Cách tiếp cận này giải quyết sự mất cân bằng thông tin giữa vùng cần khôi phục và vùng nền đã biết. Hàm mất mát tổng quát  $L_{total}$  được định nghĩa như sau:

$$L_{total} = \mathbb{E}_{t, x_0, \epsilon} [\text{mean} (\|\epsilon - \epsilon_\theta(x_t, t, x_{masked}, mask)\|^2 \odot \mathbf{W})]$$

Trong đó:

- $\epsilon$  là nhiễu Gaussian thực tế được thêm vào ảnh.
- $\epsilon_\theta$  là nhiễu được dự đoán bởi mạng U-Net.
- $\odot$  biểu thị phép nhân từng phần tử.
- $\mathbf{W}$  là ma trận trọng số không gian, có cùng kích thước với ảnh đầu vào.

**Thiết kế Ma trận trọng số:** Ma trận  $\mathbf{W}$  được xây dựng dựa trên mặt nạ nhị phân  $mask$ , nhằm điều hướng gradient tập trung vào khu vực bị mất thông tin. Cụ thể, giá trị trọng số tại mỗi vị trí pixel  $(i, j)$  được xác định bởi công thức:

$$\mathbf{W}_{i,j} = mask_{i,j} \cdot \lambda_{mask} + (1 - mask_{i,j}) \cdot \lambda_{context}$$

Với các siêu tham số được thiết lập là  $\lambda_{mask} = 1.0$  và  $\lambda_{context} = 0.05$ .

Việc thiết lập tỷ lệ trọng số 20 : 1 ( $\lambda_{mask}/\lambda_{context}$ ) mang hai ý nghĩa quan trọng:

- Đảm bảo phần lớn tín hiệu lỗi dùng để cập nhật trọng số mô hình sẽ đến từ vùng bị che ( $mask = 1$ ), buộc mô hình phải ưu tiên học cách tái tạo cấu trúc và chi tiết khuôn mặt tại vùng này.
- Trọng số nhỏ 0.05 tại vùng nền ( $mask = 0$ ) giữ vai trò duy trì sự nhất quán tổng thể, ngăn mô hình tạo ra các ranh giới bất thường giữa vùng giả định và vùng thực tế, đồng thời tận dụng thông tin nền để điều hướng việc sinh ảnh.

### 4.3 Stable Diffusion v1.5 fine-tuned bằng LoRA

#### 4.3.1 Cấu hình thí nghiệm

Mô hình cơ sở sử dụng là **Stable Diffusion 1.5 Inpainting**, được gắn thêm **LoRA** trên UNet để tinh chỉnh nhanh chóng với số lượng tham số nhỏ. Các thành phần chính của mô hình (VAE, UNet, Text Encoder) được đóng băng nhằm giảm chi phí huấn luyện.

**Cấu hình LoRA:**

- Rank  $r = 16$

- Alpha = 16
- Dropout = 0.05
- Nhắm tới các module: to\_q, to\_k, to\_v, to\_out . 0

Ngoài ra, các siêu tham số huấn luyện gồm:

- Batch size = 2
- Learning rate =  $1 \times 10^{-4}$
- Số epoch = 5
- **Mixed precision** (FP16) và **GradScaler** được áp dụng để giảm bộ nhớ GPU và tăng tốc huấn luyện.

#### 4.3.2 Pipeline mô hình

Pipeline sử dụng **Stable Diffusion 1.5 Inpainting** kết hợp **LoRA** trên UNet. Quá trình inpainting được thực hiện theo các bước sau:

1. **Tokenize và encode prompt/caption:** Mỗi ảnh được gán caption từ **BLIP (Bootstrapped Language-Image Pretraining)**, ví dụ: “a woman with long brown hair”.
2. **Encode ảnh gốc và masked image:** Ảnh ground truth và ảnh bị che được encode bằng VAE thành **latent vectors** kích thước  $64 \times 64$ .
3. **Xử lý mask:** Mask được resize về  $64 \times 64$  và đảo màu để vùng cần vẽ là màu trắng, phù hợp với yêu cầu của pipeline inpainting.
4. **Tạo nhiễu Gaussian:** Nhiễu được sinh ngẫu nhiên và thêm vào latent ảnh gốc theo timestep scheduler của Stable Diffusion.
5. **Dự đoán nhiễu bằng UNet + LoRA:** Latent bị nhiễu, masked latent và mask được đưa vào UNet gắn LoRA để dự đoán nhiễu ban đầu.
6. **Tối ưu hóa LoRA:**  $MSE$  loss giữa nhiễu dự đoán và nhiễu gốc được tính, back-propagation chỉ cập nhật các tham số LoRA.
7. **Decode latent → ảnh RGB:** Latent dự đoán được giải mã để thu được ảnh inpainting hoàn chỉnh.

**Chú thích thêm:**

- **Classifier-Free Guidance (CFG) training** được áp dụng bằng cách ngẫu nhiên bỏ prompt với xác suất 15% trong huấn luyện, giúp cải thiện khả năng điều khiển khi suy luận.
- Pipeline hỗ trợ **resume** từ **checkpoint**, lưu LoRA weights, optimizer state và GradScaler state, giúp tiếp tục huấn luyện bất cứ lúc nào.

### 4.3.3 Tiền xử lý dữ liệu

Tất cả ảnh *ground truth* và *masked image* được chuẩn hóa về kích thước  $512 \times 512$  pixel. Ảnh được chuyển sang **RGB** và chuẩn hóa về khoảng giá trị  $[-1, 1]$ . Mask được chuyển sang *tensor*, resize về  $64 \times 64$  và đảo màu để vùng cần vẽ là màu trắng, phù hợp với yêu cầu của pipeline inpainting.

Mỗi ảnh *ground truth* được gán một caption tự động bằng mô hình **BLIP Image Captioning** ([Salesforce/blip-image-captioning-base](#)). Caption được sinh offline cho toàn bộ tập train và test, sau đó lưu vào file JSON để sử dụng lại trong quá trình huấn luyện.

Để tăng khả năng sinh ảnh chi tiết, caption được mở rộng thêm các mô tả chất lượng cao như “high quality, realistic, sharp focus”.

Các file GT/mask được kiểm tra tồn tại. Nếu thiếu, ảnh đầu tiên trong dataset được dùng thay thế để tránh lỗi. Prompt cố định hoặc caption mở rộng được dùng làm điều kiện văn bản cho mô hình trong quá trình training.

Kỹ thuật **Classifier-Free Guidance (CFG) training** được áp dụng bằng cách ngẫu nhiên bỏ prompt với xác suất 15%, giúp cải thiện khả năng điều khiển khi suy luận.

### 4.3.4 Huấn luyện

Huấn luyện sử dụng optimizer **AdamW** với learning rate  $1 \times 10^{-4}$ . Mỗi batch gồm 2 ảnh được chuyển sang **FP16** để giảm bộ nhớ GPU. Latent ảnh gốc và masked được encode bằng VAE, mask resize về  $64 \times 64$ .

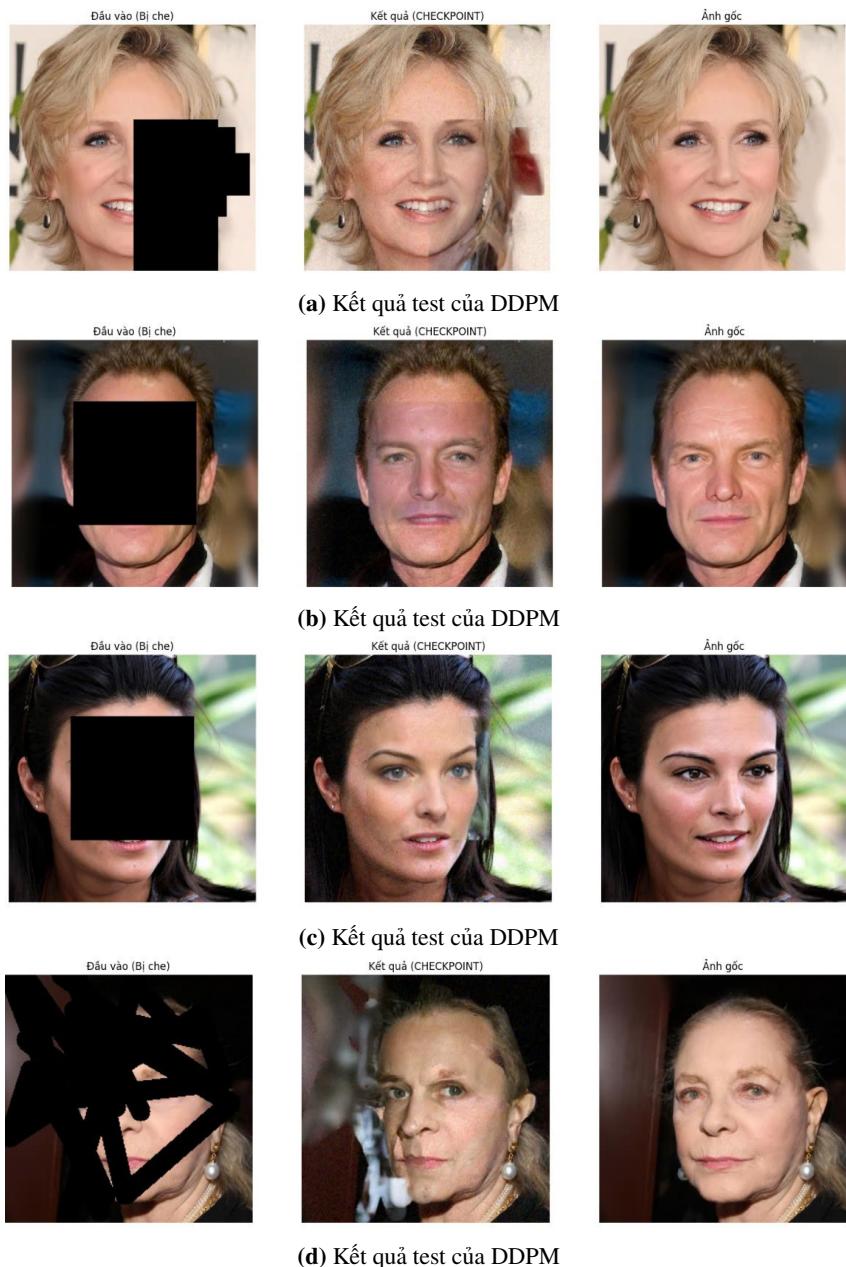
Nhiều Gaussian được thêm vào latent ảnh gốc. UNet gắn LoRA dự đoán nhiều ban đầu, sau đó tính **MSE loss** giữa nhiều dự đoán và nhiều gốc. GradScaler scale loss → backward → step optimizer → update scaler, đảm bảo training ổn định.

Sau mỗi epoch, checkpoint được lưu gồm LoRA weights, trạng thái optimizer và GradScaler. Nếu checkpoint tồn tại, huấn luyện tự động resume từ epoch tiếp theo.

## 4.4 Kết quả

### 4.4.1 Kết quả trực quan

#### Kết quả với mô hình DDPM



#### Nhận xét:

- Trường hợp (a), (b), (c): Mô hình DDPM khôi phục khá tốt cấu trúc tổng thể khuôn mặt (hình dạng mặt, vị trí mắt, mũi, miệng hợp lý), ảnh sinh nhìn tự nhiên và không xuất hiện nhiều rõ rệt. Tuy nhiên, màu da và texture ở vùng được sinh vẫn có độ lệch nhẹ so với ground truth, đặc biệt là độ sắc nét và chi tiết nhỏ (nếp nhăn, vùng mắt).

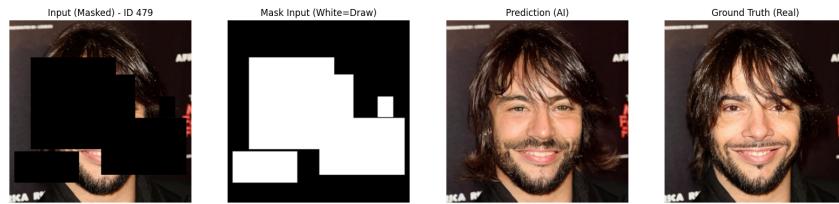
- Trường hợp (d): Khi vùng che lớn và phức tạp, kết quả sinh kém ổn định hơn. Mô

hình vẫn giữ được bối cảnh khuôn mặt, nhưng chi tiết và tính nhất quán màu sắc chưa cao, một số vùng thiếu chính xác so với ảnh gốc.

### Kết quả với Stable Diffusion v1.5

Một số ảnh kiểm thử được chọn ngẫu nhiên. Mask được đảo màu để vùng cần vẽ là màu trắng. Quá trình **inference** thực hiện 50 bước với **guidance scale = 7.5**, đồng thời sử dụng **seed cố định** để đảm bảo kết quả tái lập.

Kết quả hiển thị bao gồm 4 ảnh: *input masked*, *mask input*, *output AI* và *ground truth*. Việc đánh giá chủ yếu dựa trên **quan sát trực quan**, chú ý đến khả năng khôi phục chi tiết, mượt mà và gần gũi với ảnh gốc.



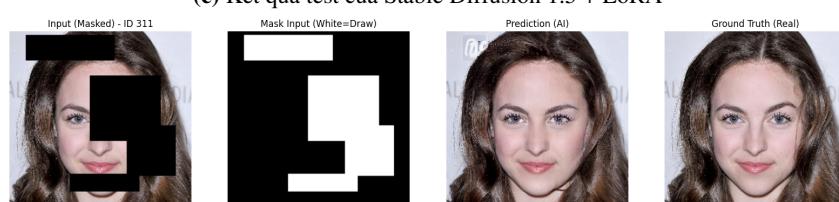
(a) Kết quả test của Stable Diffusion 1.5 + LoRA



(b) Kết quả test của Stable Diffusion 1.5 + LoRA



(c) Kết quả test của Stable Diffusion 1.5 + LoRA



(d) Kết quả test của Stable Diffusion 1.5 + LoRA

**Hình 4.2.** Các kết quả inpainting từ 4 input test khác nhau. Mỗi ảnh hiển thị output AI tương ứng.

#### 4.4.2 Kết quả đánh giá chỉ số

Metric	PSNR (dB)	SSIM	LPIPS	FID
DDPM	20.97	0.8012	0.1625	40.4452
SD1.5+LoRA	23.45	0.845	0.112	32.17

Bảng 4.1. Các chỉ số đánh giá chất lượng ảnh inpainting.

Kết quả thực nghiệm cho thấy mô hình **DDPM** đạt PSNR 20.97 dB và SSIM 0.8012, cho thấy khả năng khôi phục vùng bị che ở mức chấp nhận được nhưng sai khác so với ground truth vẫn còn tương đối lớn, có thể do sự sai khác màu sắc vùng cần vẽ và vùng đã biết. Giá trị LPIPS = 0.1625 và FID = 40.45 vẫn khá cao, phản ánh chất lượng cảm nhận thị giác và độ chân thực của ảnh inpainting còn hạn chế; các vùng được sinh lại dễ bị lệch màu hoặc thiếu nhất quán so với vùng ảnh đã biết.

Mô hình **LoRA Fine-tune** trên Stable Diffusion 1.5 Inpainting đạt chất lượng inpainting cao. PSNR đạt 23.45 dB và SSIM = 0.845, cho thấy ảnh khôi phục vừa chính xác về pixel vừa bảo toàn kết cấu tổng thể. Chỉ số LPIPS thấp (0.112) chứng tỏ *perceptual similarity* tốt, trong khi FID = 32.17 phản ánh phân phối ảnh sinh gần với phân phối ảnh thật. Kết quả này minh chứng rằng pipeline với LoRA, caption mở rộng và **CFG training** có thể sinh ảnh inpainting chi tiết và ổn định.

Hai chỉ số PSNR và SSIM, **SD1.5 + LoRA** đạt giá trị cao hơn đáng kể so với **DDPM**, cho thấy khả năng tái tạo chính xác và giữ được cấu trúc ảnh tốt hơn. Nguyên nhân chính là Stable Diffusion được huấn luyện trong không gian latent với encoder-decoder mạnh (VAE), giúp mô hình học được các biểu diễn ngữ nghĩa và cấu trúc cấp cao. Khi finetune bằng LoRA, mô hình có thể thích nghi hiệu quả với bài toán inpainting mà không làm phá vỡ các tri thức nền đã được pretrain, trong khi DDPM hoạt động trực tiếp trên không gian pixel nên khó học được sự nhất quán cấu trúc ở vùng bị che, đặc biệt khi vùng khuyết lớn.

Đối với hai chỉ số LPIPS và FID, **SD1.5 + LoRA** cho giá trị thấp hơn rõ rệt, phản ánh chất lượng cảm nhận thị giác và độ chân thực tốt hơn. Do SD1.5 là mô hình sinh ảnh quy mô lớn, được pretrain trên tập dữ liệu đa dạng, nên có khả năng sinh texture và chi tiết tự nhiên, đồng thời hòa trộn tốt vùng inpainting với ngữ cảnh xung quanh. Ngược lại, **DDPM** finetune từ mô hình tổng quát có năng lực biểu diễn hạn chế hơn, dễ sinh ra các vùng ảnh bị mờ, lệch màu hoặc thiếu tính ngữ cảnh, dẫn đến LPIPS và FID cao.

Từ những kết quả thực nghiệm trên, ta thấy **DDPM** phù hợp hơn cho các bài toán mang tính minh họa hoặc baseline, trong khi **SD1.5 + LoRA** thể hiện rõ ưu thế trong các bài toán inpainting yêu cầu chất lượng cao, nhờ kết hợp được tri thức pretrain mạnh và cơ chế finetune hiệu quả.

## Chương 5

# Kết luận

### 5.1 Kết luận

Trong khuôn khổ dự án này, nhóm đã nghiên cứu bài toán *Image Completion* trong bối cảnh các mô hình sinh ảnh hiện đại, với trọng tâm là các mô hình diffusion. Bài toán *Image Completion* được xem xét như một bài toán sinh có điều kiện, trong đó mô hình cần tái tạo các vùng ảnh bị khuyết sao cho vừa đảm bảo tính nhất quán về cấu trúc, vừa duy trì tính tự nhiên và hợp lý về mặt ngữ nghĩa thị giác.

Dự án tập trung so sánh hai hướng tiếp cận chính: (i) mô hình Denoising Diffusion Probabilistic Model (DDPM) được fine-tuning trực tiếp cho tác vụ *Image Completion*, và (ii) mô hình Stable Diffusion v1.5 - một latent diffusion model đã được huấn luyện trước trên quy mô dữ liệu lớn - được tinh chỉnh bằng kỹ thuật LoRA nhằm thích nghi với tác vụ cụ thể. Thông qua việc đánh giá trên bộ dữ liệu CelebA-HQ, nhóm tiến hành phân tích cả định tính và định lượng để làm rõ sự khác biệt về chất lượng ảnh sinh ra, mức độ nhất quán ngữ nghĩa và khả năng khôi phục chi tiết của từng mô hình.

Thông qua các thí nghiệm và phân tích, dự án đã chỉ ra rằng các mô hình diffusion, đặc biệt là Stable Diffusion v1.5, có khả năng giải quyết bài toán *Image Completion* một cách hiệu quả, vượt trội so với các mô hình diffusion thuần được huấn luyện từ đầu như DDPM. Việc thực hiện quá trình khuếch tán trong không gian tiềm ẩn không chỉ giúp giảm chi phí tính toán mà còn góp phần nâng cao chất lượng ảnh sinh, đặc biệt ở các vùng khuyết có cấu trúc và ngữ nghĩa phức tạp.

Bên cạnh đó, kỹ thuật fine-tuning hiệu quả tham số như LoRA cho thấy vai trò quan trọng trong việc thích nghi mô hình pre-trained với tác vụ cụ thể mà không làm suy giảm khả năng sinh ảnh tổng quát. Cách tiếp cận này đặc biệt phù hợp trong các bối cảnh thực tế, nơi tài nguyên tính toán và dữ liệu huấn luyện thường bị giới hạn.

### 5.2 Hướng nghiên cứu trong tương lai

Mặc dù đạt được những kết quả tích cực, dự án vẫn còn một số hạn chế và mở ra nhiều hướng nghiên cứu tiềm năng trong tương lai. Trước hết, phạm vi dữ liệu hiện tại

chủ yếu tập trung vào ảnh khuôn mặt; do đó, việc mở rộng sang các miền dữ liệu đa dạng hơn như cảnh tự nhiên, ảnh y tế hoặc ảnh vệ tinh có thể giúp đánh giá toàn diện hơn khả năng tổng quát hóa của mô hình.

Thứ hai, các nghiên cứu tiếp theo có thể xem xét việc kết hợp thêm các tín hiệu điều kiện hóa khác, chẳng hạn như mô tả văn bản hoặc ảnh tham chiếu, nhằm tăng khả năng kiểm soát nội dung sinh ra trong bài toán Image Completion. Ngoài ra, việc so sánh và tích hợp các kỹ thuật fine-tuning hiệu quả tham số khác như DoRA hoặc adapters cũng là một hướng đi đáng chú ý.

Cuối cùng, các hướng tiếp cận đánh giá nâng cao, bao gồm đánh giá dựa trên nhận thức người dùng hoặc các chỉ số phản ánh tốt hơn chất lượng ngữ nghĩa, có thể được xem xét để bổ sung cho các thước đo định lượng truyền thống. Những hướng nghiên cứu này hứa hẹn sẽ góp phần nâng cao hiệu quả và tính ứng dụng của các mô hình diffusion trong bài toán Image Completion cũng như các tác vụ chỉnh sửa ảnh nói chung.

# Tài liệu tham khảo

- [1] Jonathan Ho et al. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.
- [2] Edward Hu et al. Lora: Low-rank adaptation of large language models. 2021.
- [3] Andreas Lugmayr et al. Repaint: Inpainting using denoising diffusion probabilistic models. *CVPR*, 2022.
- [4] Deepak Pathak et al. Context encoders: Feature learning by inpainting. In *CVPR*, 2016.
- [5] Yurui Ren, Xiaoming Yu, Ruonan Zhang, Thomas H. Li, Shan Liu, and Ge Li. Structureflow: Image inpainting via structure-aware appearance flow. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 181–190, 2019.
- [6] Robin Rombach et al. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.
- [7] Nataniel Ruiz et al. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. 2023.
- [8] Jiahui Yu et al. Generative image inpainting with contextual attention. In *CVPR*, 2018.
- [9] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S. Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4471–4480, 2019.

# Phụ lục A

## Pseudo-code huấn luyện LoRA Fine-tune

```
1 Input: Dataset D = { (masked_i, mask_i, gt_i, caption_i) },
2         pretrained SD inpainting model M, LoRA config
3 Output: Fine-tuned LoRA weights
4
5 for epoch in 1..N_epochs:
6     for batch in D:
7         # Encode prompt
8         encoder_hidden_states = TextEncoder(caption_i)
9
10        # Encode images
11        latent_gt = VAE.encode(gt_i)
12        latent_masked = VAE.encode(masked_i)
13        mask_resized = Resize(mask_i, 64x64)
14
15        # Add Gaussian noise to latent_gt
16        timesteps = random_timesteps()
17        noisy_latent = Scheduler.add_noise(latent_gt, noise, timesteps)
18
19        # Concatenate inputs for UNet
20        latent_input = Concat(noisy_latent, mask_resized, latent_masked)
21
22        # Predict noise
23        noise_pred = UNet_LoRA(latent_input, timesteps, encoder_hidden_states)
24
25        # Compute MSE loss
26        loss = MSE(noise_pred, noise)
27
28        # Backprop LoRA parameters only
29        loss.backward()
30        Optimizer.step()
31        GradScaler.update()
32
33        Save checkpoint(epoch)
```

Listing 5.1: Pseudo-code Huấn luyện LoRA Fine-tune

## Phân công thành viên

Thành viên	MSV	Nhiệm vụ đảm nhận
Nguyễn Thị Thanh Huyền	23020381	<ul style="list-style-type: none"><li>Tìm hiểu và nghiên cứu đề tài</li><li>Thực hiện thực nghiệm với DDPM</li><li>Viết tài liệu phần Thực nghiệm</li></ul>
Nguyễn Thị Minh Ly	23020399	<ul style="list-style-type: none"><li>Tìm hiểu và nghiên cứu đề tài</li><li>Thực hiện thực nghiệm với Stable Diffusion</li><li>Xây dựng UI demo</li><li>Viết tài liệu phần Thực nghiệm</li></ul>
Đặng Minh Nguyệt	23020407	<ul style="list-style-type: none"><li>Tìm hiểu và nghiên cứu đề tài</li><li>Xử lý dữ liệu</li><li>Viết tài liệu phần Giới thiệu, Công trình nghiên cứu liên quan, Cơ sở lý thuyết</li></ul>

**Bảng 5.1.** Phân công công việc của các thành viên trong nhóm