

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN  
KHOA KHOA HỌC MÁY TÍNH

TRẦN ĐÌNH KHÁNH ĐĂNG

KHOÁ LUẬN TỐT NGHIỆP

TĂNG CƯỜNG KHẢ NĂNG CHUYỂN KIẾU  
CHỮ ĐA NGÔN NGỮ TRONG BÀI TOÁN ONE-  
SHOT BẰNG MÔ HÌNH KHUẾCH TÁN

Enhancing One-shot Cross-Script Font Style  
Transfer using Diffusion Model

CỦ NHÂN NGÀNH KHOA HỌC MÁY TÍNH

TP.HỒ CHÍ MINH, 2025

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN  
KHOA KHOA HỌC MÁY TÍNH

TRẦN ĐÌNH KHÁNH ĐĂNG - 22520195

KHOÁ LUẬN TỐT NGHIỆP

TĂNG CƯỜNG KHẢ NĂNG CHUYÊN KIỀU  
CHỮ ĐA NGÔN NGỮ TRONG BÀI TOÁN ONE-  
SHOT BẰNG MÔ HÌNH KHUẾCH TÁN

Enhancing One-shot Cross-Script Font Style  
Transfer using Diffusion Model

CỦ NHÂN NGÀNH KHOA HỌC MÁY TÍNH

GIÁNG VIÊN HƯỚNG DẪN:  
TS. Dương Việt Hằng

TP.HỒ CHÍ MINH, 2025

# **THÔNG TIN HỘI ĐỒNG CHẤM KHOÁ LUẬN TỐT NGHIỆP**

Hội đồng chấm khoá luận tốt nghiệp, thành lập theo Quyết định số 20 ngày 01  
của Hiệu trưởng Trường Đại học Công nghệ Thông tin.

Chủ tịch

**TS. Ngô Đức Thành**

Thư ký

**ThS. Nguyễn Thanh Sơn**

Uỷ viên

**TS. Mai Tiến Dũng**

## LỜI CẢM ƠN

Lời đầu tiên, em xin gửi lời tri ân sâu sắc đến Ban Giám hiệu cùng Quý Thầy/Cô Khoa Khoa học Máy tính, Trường Đại học Công nghệ Thông tin – ĐHQG-HCM. Môi trường học thuật năng động và chuyên nghiệp tại đây không chỉ trang bị cho em nền tảng kiến thức vững chắc mà còn là nơi rèn giữa tư duy, giúp em trưởng thành hơn từng ngày.

Đặc biệt, em xin bày tỏ lòng biết ơn chân thành nhất đến cô Dương Việt Hằng. Trong suốt quá trình thực hiện khoá luận, Cô không chỉ là người hướng dẫn tận tâm, định hướng chuyên môn mà còn là người truyền cảm hứng, luôn kiên nhẫn giải đáp và động viên em vượt qua những giai đoạn khó khăn nhất. Sự chỉ dẫn quý báu của Cô chính là kim chỉ nam giúp em hoàn thành tốt đề tài nghiên cứu này.

Bên cạnh đó, con xin cảm ơn gia đình – điểm tựa tinh thần vững chắc, và cảm ơn bạn bè, những người đồng đội đã luôn kè vai sát cánh, chia sẻ và hỗ trợ em trong suốt chặng đường sinh viên vừa qua.

Kính chúc Quý Thầy/Cô, gia đình và bạn bè luôn dồi dào sức khỏe, hạnh phúc và gặt hái được nhiều thành công trong sự nghiệp cũng như cuộc sống.

Em xin chân thành cảm ơn!

# MỤC LỤC

<b>Thông tin hội đồng chấm khoá luận tốt nghiệp .....</b>	i
<b>Lời cảm ơn .....</b>	ii
<b>Mục lục .....</b>	iii
<b>Danh mục hình ảnh .....</b>	viii
<b>Danh mục bảng .....</b>	x
<b>Danh mục từ viết tắt .....</b>	xii
<b>Danh mục giải thuật .....</b>	xiii
<b>Tóm tắt .....</b>	xiv
<b>Tóm tắt .....</b>	xv
<b>Chương 1. Giới thiệu .....</b>	1
1.1. Giới thiệu bài toán .....	1
1.2. Mô tả bài toán .....	2
1.3. Mục tiêu của đề tài .....	4
1.4. Đối tượng và phạm vi nghiên cứu .....	4
1.4.1. Đối tượng nghiên cứu .....	4
1.4.2. Phạm vi nghiên cứu .....	5
1.5. Cấu trúc của khoá luận .....	6
<b>Chương 2. Cơ sở lý thuyết .....</b>	8
2.1. Một số phương pháp tiếp cận cho bài toán Sinh phông chữ (Font Generation) .....	8
2.1.1. Các phương pháp dựa trên GAN (Generative Adversarial Networks) .....	8
2.1.1.1. DG-Font (Deformable Generative Network, CVPR 2021) .	8

2.1.1.2. CF-Font (Content Fusion, CVPR 2023) .....	9
2.1.1.3. DFS (Few-Shot Text Style Transfer via Deep Feature Similarity, TIP 2020) .....	10
2.1.1.4. FTransGAN (Few-shot Font Style Transfer between Different Languages, WACV 2021) .....	13
2.1.2. Mô hình khuếch tán (Diffusion Models) .....	13
2.1.2.1. Quá trình Khuếch tán xuôi (Forward Diffusion Process) .	14
2.1.2.2. Quá trình Khuếch tán ngược (Reverse Diffusion Process) .....	15
2.1.2.3. Hàm mất mát (Loss function) .....	16
2.1.2.4. FontDiffuser (AAAI 2024) .....	16
2.2. Một số phương pháp tiếp cận cho bài toán Biểu diễn Phong cách (Style Representation) .....	16
2.2.1. Neural Style Transfer truyền thống .....	17
2.2.2. Học tương phản (Contrastive Learning) .....	17
2.3. Thách thức trong bài toán Cross-Lingual: Chuyển đổi Hai chiều giữa Latin và Hán tự .....	19
2.3.1. Vấn đề Chênh lệch Mật độ Thông tin (Information Density Asymmetry) .....	19
2.3.2. Khoảng cách Hình thái học (Morphological Gap) .....	20
<b>Chương 3. Phương pháp đề xuất .....</b>	<b>21</b>
3.1. Kiến trúc nền tảng FontDiffuser .....	21
3.1.1. Giai đoạn 1 - Tái tạo cấu trúc (Reconstruction Phase) .....	23
3.1.2. Giai đoạn 2 - Tinh chỉnh phong cách (Style Refinement Phase) ...	25
3.1.2.1. Kiến trúc Khai thác Phong cách .....	25
3.1.2.2. Cơ chế Học Tương phản và Hàm Mất mát .....	27

3.1.3. Kết hợp vào Mục tiêu Huấn luyện .....	28
3.2. Cải tiến đề xuất: Cross-Lingual Style Contrastive Refinement (CL-SCR) .	29
3.2.1. Hạn chế của SCR trong bối cảnh đa ngôn ngữ .....	29
3.2.2. Thiết kế mô-đun CL-SCR .....	30
3.2.2.1. Chiến lược lấy mẫu mở rộng .....	30
3.2.2.2. Cơ chế tính toán Loss hỗn hợp .....	31
3.2.2.3. Quy trình huấn luyện Pha 2 cải tiến .....	33
3.3. Đề xuất thuật toán tính CL-SCR .....	34
<b>Chương 4. Thực nghiệm và Đánh giá kết quả .....</b>	<b>36</b>
4.1. Bộ dữ liệu (Datasets) .....	36
4.1.1. Cấu trúc .....	36
4.1.2. Tiền xử lý và Chuẩn hoá .....	37
4.2. Thiết lập Thực nghiệm .....	38
4.2.1. Cấu hình Huấn luyện (Implementation Details) .....	38
4.2.2. Kịch bản Đánh giá (Evaluation Scenarios) .....	40
4.3. Các thước đo đánh giá (Evaluation Metrics) .....	40
4.3.1. Chỉ số Định lượng (Quantitative Metrics) .....	40
4.3.1.1. L1 (Mean Absolute Error) .....	41
4.3.1.2. SSIM (Structural Similarity Index) .....	41
4.3.1.3. LPIPS (Learned Perceptual Image Patch Similarity) .....	42
4.3.1.4. FID (Fréchet Inception Distance) .....	42
4.3.1.5. Phân tích mối tương quan và Vai trò của bộ độ đo .....	43
4.3.2. Đánh giá Định tính (Qualitative Evaluation) .....	43

4.3.2.1. Quy trình Phân tích Trực quan (Visual Analysis Protocol) .....	43
4.3.2.2. Thiết kế Khảo sát Người dùng (User Study Design) .....	44
4.4. Kết quả Thực nghiệm và Thảo luận .....	45
4.4.1. So sánh Định lượng .....	47
4.4.1.1. Tác vụ chuyển đổi phong cách từ chữ Latin sang ảnh nguồn Hán (e2c) .....	48
4.4.1.2. Tác vụ chuyển đổi phong cách từ chữ Hán sang ảnh nguồn Latin (c2e) .....	52
4.4.2. So sánh Định tính .....	56
4.4.2.1. Phân tích Trực quan .....	56
4.4.2.2. Đánh giá Cảm nhận Người dùng .....	57
4.5. Nghiên cứu Bóc tách (Ablation Study) .....	58
4.5.1. Ảnh hưởng của các mô-đun trong FontDiffuser .....	58
4.5.2. Ảnh hưởng của Tăng cường dữ liệu (Data Augmentation) .....	62
4.5.3. Ảnh hưởng của Chế độ hàm loss (Loss Mode) .....	64
4.5.4. Ảnh hưởng của Số lượng mẫu âm (Negative Sample Numbers) ...	68
4.5.5. Ảnh hưởng của Alpha và Beta .....	71
4.5.6. Ảnh hưởng của Trọng số hướng dẫn (Guidance Scale) .....	74
<b>Chương 5. Kết luận và Hướng phát triển .....</b>	<b>79</b>
5.1. Kết quả đạt được .....	79
5.2. Các định hướng phát triển .....	80
<b>Tài liệu tham khảo .....</b>	<b>81</b>
<b>Công trình khoa học .....</b>	<b>86</b>

<b>Phụ lục .....</b>	<b>87</b>
A. Chi tiết Kiến trúc mạng UNet .....	87
B. Chi tiết Kiến trúc mô-đun CL-SCR .....	88
C. Các siêu tham số Tiền huấn luyện CL-SCR .....	90
D. Các siêu tham số huấn luyện .....	91
E. Các tham số quá trình suy luận .....	93
F. Chi phí Tính toán và Thời gian .....	94
F.1. Thời gian Huấn luyện .....	94
F.2. Tốc độ Suy diễn .....	96

## DANH MỤC HÌNH ẢNH

Hình 1.1 Ví dụ minh họa các ảnh tham chiếu nội dung. ....	2
Hình 1.2 Ví dụ minh họa các ảnh tham chiếu phong cách. ....	3
Hình 1.3 Minh họa quy trình sinh ảnh trong bài toán chuyển đổi phong cách ký tự: mô hình nhận ảnh tham chiếu nội dung và ảnh tham chiếu phong cách làm đầu vào, sau đó sinh ra ảnh ký tự mới giữ nguyên nội dung nhưng mang phong cách của ảnh tham chiếu; kết quả được so sánh với ảnh chuẩn để đánh giá chất lượng. ....	3
Hình 2.1 Kiến trúc mạng DG-Font. Mô-đun FDSC đóng vai trò nòng cốt trong việc học biến dạng hình học giữa các ký tự. (Trích từ ) .....	9
Hình 2.2 Minh họa cơ chế Content Fusion: Các đặc trưng từ tập font cơ sở (Source) được tổ hợp tuyển tính dựa trên bộ trọng số dự đoán (Weights) để xấp xỉ cấu trúc hình học của font mục tiêu. (Trích từ ) .....	10
Hình 2.3 Kiến trúc mạng DFS với thành phần cốt lõi là Ma trận Tương đồng (SM) giúp điều hướng dòng chảy thông tin phong cách. (Trích từ ) .....	12
Hình 2.4 Tổng quan kiến trúc FTransGAN. (Trích từ ) .....	13
Hình 2.5 Quá trình Khuếch tán xuôi. (Trích từ ) .....	14
Hình 2.6 Quá trình Khuếch tán ngược. (Trích từ ) .....	15
Hình 2.7 So sánh cấu trúc hình thái giữa chữ Latin và Hán tự. Chữ Latin được tổ chức theo bố cục tuyển tính dựa trên baseline, với các tham số hình học như x-height và cap height, đồng thời có độ rộng ký tự biến thiên. Ngược lại, Hán tự tuân theo cấu trúc khói vuông cố định (body frame), trong đó các nét bút được phân bố để cân bằng và lắp đầy không gian nội khói. Sự khác biệt căn bản về hệ quy chiếu hình học này tạo nên khoảng cách hình thái học giữa hai hệ chữ. (Trích từ ) .....	20
Hình 3.1 Mô hình tổng thể của FontDiffuser gồm 2 giai đoạn: Tái tạo cấu trúc (Trái) và Tinh chỉnh phong cách (Phải). (Trích từ ) .....	21
Hình 3.2 Ví dụ về ảnh nội dung. ....	22
Hình 3.3 Ví dụ về ảnh phong cách. ....	22
Hình 3.4 Ví dụ về ảnh đầu ra. ....	22
Hình 3.5 Khối MCA (Multi-scale Content Aggregation). (Trích từ ) .....	24
Hình 3.6 Đặc trưng Content ở các khói khác nhau. (Trích từ ) .....	24
Hình 3.7 Kiến trúc của mô-đun SCR. (Trích từ ) .....	26
Hình 3.8 Kiến trúc của mô-đun Tinh chỉnh Tương phản Phong cách Đa ngôn ngữ (CL-SCR), minh họa với ví dụ cụ thể về việc sinh ký tự Latin ‘d’ theo phong cách đích ‘Z’. Với đầu vào là mẫu sinh $x_{\text{sample}}$ đóng vai trò là anchor, mô-đun sử dụng bộ trích xuất phong cách (Style Extractor) chia sẻ trọng số để ánh xạ các đầu vào vào không gian vector tiềm ẩn. Các	

vector đặc trưng trích xuất được ký hiệu là $V_{\text{sample}}$ (anchor), $V_p$ (đại diện cho các mẫu dương $x_{\text{intra}}^+$ và $x_{\text{cross}}^+$ ), và $V_n$ (đại diện cho các tập mẫu âm $X_{\text{N,intra}}^-$ và $X_{\text{N,cross}}^-$ ). Mục tiêu huấn luyện sử dụng cơ chế giám sát kép để đảm bảo phong cách ‘Z’ được nắm bắt chính xác: 1) Nhánh nội miền ( $L_{\text{intra}}$ ): Căn chỉnh $V_{\text{sample}}$ với đặc trưng nhãn $V_p$ (từ $x_{\text{intra}}^+$ ) đồng thời đẩy xa các đặc trưng âm $V_n$ (từ $X_{\text{N,intra}}^-$ ); và 2) Nhánh xuyên miền ( $L_{\text{cross}}$ ): Thu hẹp khoảng cách phong cách bằng cách kéo $V_{\text{sample}}$ lại gần đặc trưng dương xuyên ngôn ngữ $V_p$ (từ ký tự Hán $x_{\text{cross}}^+$ ). Mục tiêu cuối cùng $L_{\text{CL-SCR}}$ kết hợp các lực này để đảm bảo sự chuyển giao thẩm mỹ nhất quán qua các hệ chữ viết. ....	31
Hình 4.1 Minh họa hai hệ chữ trong cùng một bộ font. ....	37
Hình 4.2 Ví dụ về ảnh nội dung và ảnh tham chiếu. ....	44
Hình 4.3 Ví dụ về các kết quả mà người tham gia khảo sát có thể chọn. ....	45
Hình 4.4 Ví dụ ba loại độ phức tạp. (Trích từ ) ....	47
Hình 4.5 So sánh ảnh sinh trên tập SFUC cho kịch bản Latin → Hán tự (e2c) giữa các phương pháp và ground truth. ....	50
Hình 4.6 So sánh ảnh sinh trên tập UFSC cho kịch bản Latin → Hán tự (e2c) giữa các phương pháp và ground truth. ....	51
Hình 4.7 So sánh ảnh sinh trên tập SFUC cho kịch bản Hán tự → Latin (c2e) giữa các phương pháp và ground truth. ....	54
Hình 4.8 So sánh ảnh sinh trên tập UFSC cho kịch bản Hán tự → Latin (c2e) giữa các phương pháp và ground truth. ....	55
Hình 4.9 Biểu đồ so sánh tỷ lệ ưu tiên của người dùng giữa phương pháp đề xuất (Ours) và các phương pháp SOTA khác. Kết quả cho thấy sự vượt trội về độ hài lòng thị giác của mô hình tích hợp CL-SCR. ....	57

## DANH MỤC BẢNG

Bảng 4.1	Bảng phân loại các kịch bản dựa trên độ phức tạp của ký tự. ....	46
Bảng 4.2	Kết quả Định lượng cho Latin → Hán tự (e2c) trên SFUC. Mũi tên chỉ hướng tốt hơn (thấp hơn hoặc cao hơn). ....	48
Bảng 4.3	Kết quả Định lượng cho Latin → Hán tự (e2c) trên UFSC. Mũi tên chỉ hướng tốt hơn (thấp hơn hoặc cao hơn). ....	48
Bảng 4.4	Kết quả Định lượng cho Hán tự → Latin (c2e) trên SFUC. Mũi tên chỉ hướng tốt hơn (thấp hơn hoặc cao hơn). ....	52
Bảng 4.5	Kết quả Định lượng cho Hán tự → Latin (c2e) trên UFSC. Mũi tên chỉ hướng tốt hơn (thấp hơn hoặc cao hơn). ....	52
Bảng 4.6	Ảnh hưởng của các thành phần M, R, S và CL đến hiệu năng mô hình trên tác vụ Latin → Hán tự (e2c). ....	59
Bảng 4.7	Ảnh hưởng của các thành phần M, R, S và CL đến hiệu năng mô hình trên tác vụ Hán tự → Latin (c2e). ....	59
Bảng 4.8	So sánh kết quả sinh ảnh giữa các mô-đun khác nhau trên tập dữ liệu chưa từng thấy cho hai hướng tác vụ (e2c và c2e). ....	61
Bảng 4.9	Ảnh hưởng của tăng cường dữ liệu đối với hiệu năng mô hình trên tác vụ Latin → Hán tự (e2c). ....	62
Bảng 4.10	Ảnh hưởng của tăng cường dữ liệu đối với hiệu năng mô hình trên tác vụ Hán tự → Latin (c2e). ....	62
Bảng 4.11	So sánh kết quả sinh ảnh giữa mô hình có và không áp dụng tăng cường dữ liệu trên tập dữ liệu chưa từng thấy cho hai hướng tác vụ (e2c và c2e). ....	64
Bảng 4.12	Ảnh hưởng của các chế độ loss đối với hiệu năng mô hình trên tác vụ Latin → Hán tự (e2c). ....	65
Bảng 4.13	Ảnh hưởng của các chế độ loss đối với hiệu năng mô hình trên tác vụ Hán tự → Latin (c2e). ....	65
Bảng 4.14	So sánh kết quả sinh ảnh giữa các chế độ mât mát khác nhau trên tập dữ liệu chưa từng thấy cho hai hướng tác vụ (e2c và c2e). ....	67
Bảng 4.15	Ảnh hưởng của số lượng mẫu âm đối với hiệu năng mô hình trên tác vụ Latin → Hán tự (e2c). ....	68
Bảng 4.16	Ảnh hưởng của số lượng mẫu âm đối với hiệu năng mô hình trên tác vụ Hán tự → Latin (c2e). ....	68
Bảng 4.17	So sánh kết quả sinh ảnh giữa các số lượng mẫu âm khác nhau trên tập dữ liệu chưa từng thấy cho cả hai hướng tác vụ (e2c và c2e). .	70
Bảng 4.18	Ảnh hưởng của alpha và beta đối với hiệu năng mô hình trên tác vụ Latin → Hán tự (e2c). ....	71

Bảng 4.19	Ảnh hưởng của alpha và beta đối với hiệu năng mô hình trên tác vụ Hán tự → Latin (c2e). . . . .	71
Bảng 4.20	So sánh kết quả sinh ảnh giữa các alpha và beta khác nhau trên tập dữ liệu chưa từng thấy cho cả hai hướng tác vụ (e2c và c2e). . . . .	73
Bảng 4.21	Ảnh hưởng của trọng số hướng dẫn đối với hiệu năng mô hình trên tác vụ Latin → Hán tự (e2c). . . . .	74
Bảng 4.22	Ảnh hưởng của trọng số hướng dẫn đối với hiệu năng mô hình trên tác vụ Hán tự → Latin (c2e). . . . .	75
Bảng 4.23	So sánh kết quả sinh ảnh giữa các trọng số hướng dẫn khác nhau trên tập dữ liệu chưa từng thấy cho cả hai hướng tác vụ (e2c và c2e). . . . .	77
Bảng A.1	Chi tiết kiến trúc mạng UNet trong FontDiffuser. Trong đó: MCA là khối Tổng hợp nội dung đa quy mô, SI là khối Chèn phong cách (Style Insertion) sử dụng cơ chế Cross-Attention. . . . .	87
Bảng B.2	Chi tiết kiến trúc và luồng dữ liệu của mô-đun CL-SCR. Các ký hiệu $\text{ReLU}_1^x$ biểu thị lớp kích hoạt đầu tiên trong mỗi khối VGG. . . . .	88
Bảng C.3	Bảng tổng hợp các siêu tham số cho giai đoạn tiền huấn luyện CL-SCR. . . . .	90
Bảng D.4	Bảng tổng hợp các siêu tham số huấn luyện cho cả hai giai đoạn. . . . .	91
Bảng E.5	Bảng các tham số cấu hình cho quá trình suy luận (Inference). . . . .	93
Bảng F.6	Thời gian huấn luyện cho từng giai đoạn của phương pháp đề xuất (Ours). . . . .	94
Bảng F.7	So sánh tổng thời gian huấn luyện giữa phương pháp đề xuất và các Baseline. . . . .	95
Bảng F.8	So sánh tốc độ suy diễn. . . . .	96

## DANH MỤC TỪ VIẾT TẮT

<b>CF-Font</b>	Content Fusion Font
<b>CL-SCR</b>	Cross-Lingual Style Contrastive Refinement
<b>DCN</b>	Deformable Convolutional Networks
<b>DDPM</b>	Denoising Diffusion Probabilistic Models
<b>DFS</b>	Deep Feature Similarity
<b>DG-Font</b>	Deformable Generative Network
<b>DPM-Solver</b>	Diffusion Probabilistic Model Solver
<b>FID</b>	Fréchet Inception Distance
<b>GAN</b>	Generative Adversarial Network
<b>GT</b>	Ground Truth
<b>L1</b>	Mean Absolute Error
<b>LPIPS</b>	Learned Perceptual Image Patch Similarity
<b>MCA</b>	Multi-scale Content Aggregation
<b>MSE</b>	Mean Squared Error
<b>RSI</b>	Reference-Structure Interaction
<b>SCR</b>	Style Contrastive Refinement
<b>SFUC</b>	Seen Font, Unseen Character
<b>SOTA</b>	State-of-the-Art
<b>SSIM</b>	Structural Similarity Index
<b>UFSC</b>	Unseen Font, Seen Character

## **DANH MỤC GIẢI THUẬT**

Thuật toán 3.1 Thuật toán tính hàm mất mát CL-SCR ..... 34

## TÓM TẮT

**Tóm tắt:** Bài toán sinh phông chữ tự động là một nhánh quan trọng trong thị giác máy tính, nhằm tạo ra các ký tự mới với phong cách (style) đồng nhất từ một số lượng mẫu tối thiểu. FontDiffuser là một phương pháp tiên tiến dựa trên mô hình khuếch tán (Diffusion Model), cho phép sinh ảnh ký tự chất lượng cao và duy trì tính nhất quán về phong cách tốt hơn so với các mô hình GAN truyền thống.

Trong nghiên cứu này, em kế thừa pipeline huấn luyện hai giai đoạn của FontDiffuser (trong đó giai đoạn 2 sử dụng Style Contrastive Refinement – SCR) và **đề xuất mở rộng SCR sang bài toán cross-lingual**. Cụ thể, em thiết kế **cross-lingual SCR loss** nhằm học biểu diễn phong cách bất biến theo ngôn ngữ, đồng thời bổ sung cơ chế điều chỉnh trọng số giữa **intra-loss** và **cross-loss** để tối ưu chất lượng sinh font trong bối cảnh dữ liệu đa ngôn ngữ.

Hệ thống được bổ sung cơ chế checkpoint giúp tiếp tục huấn luyện từ trạng thái trước đó, hỗ trợ tập dữ liệu lớn và rút ngắn thời gian huấn luyện. Kết quả thực nghiệm cho thấy phương pháp đề xuất cải thiện đáng kể độ trung thành phong cách (style consistency) và chất lượng trực quan của ký tự sinh ra, đồng thời tăng khả năng tổng quát hoá khi áp dụng phong cách từ hệ chữ này sang hệ chữ khác.

**Từ khoá:** *FontDiffuser, Style Contrastive Refinement, Cross-lingual SCR, Diffusion Model, Font Generation*

## ABSTRACT

**Abstract:** Automatic font generation is an important research direction in computer vision, aiming to synthesize new characters with consistent stylistic properties from a minimal number of reference samples. FontDiffuser is a state-of-the-art approach based on diffusion models, capable of generating high-quality character images while preserving stylistic consistency more effectively than traditional GAN-based methods.

In this study, we inherit the two-stage training pipeline of FontDiffuser, in which the second stage employs Style Contrastive Refinement (SCR), and **propose an extension of SCR to the cross-lingual font generation setting**. Specifically, we design a **cross-lingual SCR loss** to learn language-invariant style representations, enabling effective style transfer across different writing systems. In addition, we introduce a weighting mechanism to balance the **intra-loss** and **cross-loss**, thereby optimizing font generation quality under multilingual data conditions.

Furthermore, a checkpointing mechanism is incorporated into the system to allow training to resume from previous states, improving scalability to large datasets and reducing overall training time. Experimental results demonstrate that the proposed method significantly enhances style fidelity and visual quality of the generated characters, while also improving generalization performance when transferring styles across different scripts.

**Keywords:** *FontDiffuser, Style Contrastive Refinement, Cross-lingual SCR, Diffusion Model, Font Generation*

# Chương 1

## Giới thiệu

### 1.1. Giới thiệu bài toán

Thiết kế phông chữ (Typeface design) từ lâu đã được xem là một loại hình nghệ thuật đòi hỏi sự kết hợp tinh tế giữa thẩm mĩ và kỹ thuật. Để tạo ra một bộ phông chữ hoàn chỉnh, các nhà thiết kế (typographers) phải vẽ thủ công hàng nghìn ký tự (glyphs) nhằm đảm bảo sự nhất quán về phong cách (style) như độ dày nét, hình dáng chân chữ (serif), và độ cong. Thách thức này càng trở nên lớn hơn đối với các hệ chữ tượng hình phức tạp như CJK (Chinese, Japanese, Korean), nơi số lượng ký tự có thể lên tới hàng chục nghìn. Do đó, các phương pháp truyền thống dựa trên nội suy (interpolation) hoặc vector hoá thủ công thường tốn kém nhiều chi phí, thời gian và khó mở rộng quy mô.

Trong bối cảnh đó, bài toán Sinh phông chữ tự động (Automatic Font Generation) đã trở thành một hướng nghiên cứu mũi nhọn trong lĩnh vực Thị giác máy tính (Computer Vision) và Học sâu (Deep Learning). Sự chuyển dịch từ các mô hình Generative Adversarial Networks (GANs)<sup>[1]</sup> sang Denoising Diffusion Probabilistic Models (DDPMs)<sup>[2], [3]</sup> gần đây đã tạo ra bước đột phá về chất lượng ảnh sinh. Các mô hình Diffusion, điển hình như FontDiffuser<sup>[4]</sup>, đã chứng minh khả năng vượt trội trong việc tái tạo các chi tiết nét chữ phức tạp và duy trì cấu trúc tò pô học của ký tự mà không gặp phải các vấn đề về mất ổn định khi huấn luyện (mode collapse) thường thấy ở GAN<sup>[1]</sup>.

Tuy nhiên, phần lớn các nghiên cứu hiện tại chỉ tập trung vào bài toán đơn ngôn ngữ (intra-lingual), tức là sinh chữ cái Latin từ mẫu Latin, hoặc sinh chữ Hán từ mẫu Hán<sup>[5], [6], [7], [8], [9], [10]</sup>. Một thách thức lớn hơn và vẫn còn nhiều “khoảng trống” nghiên cứu là bài toán sinh phông chữ đa ngôn ngữ (cross-lingual font generation).

Vấn đề cốt lõi của bài toán đa ngôn ngữ nằm ở “khoảng cách miền” (domain gap) giữa các hệ chữ viết. Ví dụ, việc chuyển đổi phong cách từ một chữ Hán (với cấu trúc

nét phức tạp, ô vuông) sang chữ cái Latin (cấu trúc đơn giản, tuyển tính) đòi hỏi mô hình phải có khả năng:

**Tách biệt hoàn toàn (Disentanglement)** giữa **nội dung (content)** và **phong cách (style)**.

Học được các **đặc trưng phong cách bất biến (invariant style features)** – những đặc điểm thẩm mỹ trừu tượng không phụ thuộc vào cấu trúc hình học của ngôn ngữ gốc.

Đây là một bài toán khó, bởi nếu không được xử lý tốt, mô hình thường có xu hướng “áp đặt” cấu trúc của ngôn ngữ nguồn lên ngôn ngữ đích, dẫn đến các ký tự bị biến dạng hoặc mất đi tính dễ đọc (legibility).

## 1.2. Mô tả bài toán

Phần này sẽ định nghĩa bài toán sinh phông chữ đa ngôn ngữ dưới dạng một bài toán chuyển đổi phong cách ảnh (Image-to-Image Translation)[11], [12], [13], [14] có điều kiện.

**Định nghĩa Đầu vào (Input):** Mô hình nhận vào hai luồng thông tin chính:

**Ảnh tham chiếu nội dung (Content Image -  $I_c$ ):** Là một **hình ảnh chứa ký tự mục tiêu c (target glyph)** được thể hiện dưới dạng phông chữ tiêu chuẩn (ví dụ: Arial hoặc Noto Sans), đóng vai trò **cung cấp thông tin về cấu trúc hình học và định danh của ký tự cần sinh** (ví dụ: chữ ‘A’, chữ ‘g’). Trong khuôn khổ bài toán cross-lingual,  $I_c$  được quy định thuộc **hệ ngôn ngữ đích** (Target Language, ví dụ: Latin).



**Hình 1.1** — Ví dụ minh họa các ảnh tham chiếu nội dung.

**Ảnh tham chiếu phong cách (Style Images -  $I_s$ ):** Là **tập hợp** một hoặc một vài **hình ảnh (k-shot)** chứa các **ký tự bất kỳ mang phong cách s mong muốn**, đóng vai trò **cung cấp các đặc trưng thẩm mỹ** (như nét xước, độ đậm nhạt, serif...). Trong bài toán cross-lingual,  $I_s$  thường thuộc **hệ ngôn ngữ nguồn**

(Source Language, ví dụ: Tiếng Trung Quốc), khác biệt hoàn toàn so với ngôn ngữ của  $I_c$ .



**Hình 1.2** — Ví dụ minh họa các ảnh tham chiếu phong cách.

**Định nghĩa Đầu ra (Output):**

**Ảnh được sinh ra (Generated Image -  $I_{\text{gen}}$ )**: Là hình ảnh kết quả thể hiện ký tự  $c$  nhưng mang phong cách  $s$ .

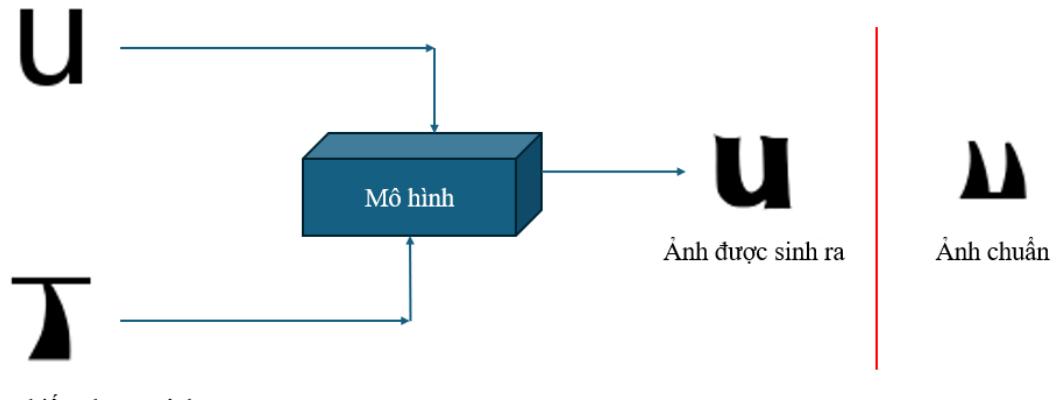
**Yêu cầu:**  $I_{\text{gen}}$  phải giữ được cấu trúc nội dung của  $I_c$  (đọc được là chữ gì) và mang đầy đủ đặc điểm thẩm mỹ của  $I_s$  (nhìn giống font mẫu).

**Mục tiêu toán học:** Mục tiêu là huấn luyện một hàm ánh xạ  $G$  (Generator/Diffusion Model) sao cho:

$$I_{\text{gen}} = G(I_c, I_s) \quad (1.1)$$

Thoả mãn điều kiện:  $\text{Content}(I_{\text{gen}}) \approx \text{Content}(I_c)$  và  $\text{Style}(I_{\text{gen}}) \approx \text{Style}(I_s)$ .

Ảnh tham chiếu nội dung



Ảnh tham chiếu phong cách

**Hình 1.3** — Minh họa quy trình sinh ảnh trong bài toán chuyển đổi phong cách ký tự: mô hình nhận ảnh tham chiếu nội dung và ảnh tham chiếu phong cách làm đầu vào, sau đó sinh ra ảnh ký tự mới giữ nguyên nội dung nhưng mang phong cách của ảnh tham chiếu; kết quả được so sánh với ảnh chuẩn để đánh giá chất lượng.

## 1.3. Mục tiêu của đề tài

Khoá luận này đề xuất mở rộng mô hình FontDiffuser để giải quyết bài toán **sinh phông chữ đa ngôn ngữ (Cross-lingual Font Generation)**, cụ thể:

Thiết kế quy trình (pipeline) cho phép chuyển đổi phong cách hai chiều linh hoạt: trích xuất phong cách từ hệ chữ Latin để áp dụng lên Hán tự và ngược lại.

Đề xuất cơ chế **Cross-Lingual Style Contrastive Refinement (CL-SCR)** cải tiến từ mô-đun SCR gốc, tích hợp chiến lược lấy mẫu âm đa dạng (cả nội miền và xuyên miền) nhằm buộc mô hình học được các đặc trưng phong cách bất biến, không phụ thuộc vào ngôn ngữ.

Thực hiện huấn luyện và tinh chỉnh mô hình khuếch tán trên dữ liệu song ngữ, đồng thời đánh giá toàn diện chất lượng đầu ra dựa trên các thước đo định lượng (LPIPS, FID, SSIM, L1) và khảo sát cảm nhận người dùng.

Mục tiêu cuối cùng là tạo ra một mô hình có khả năng sinh bộ font nhất quán đa ngôn ngữ chỉ từ một mẫu tham chiếu duy nhất, mở ra tiềm năng ứng dụng trong số hoá phông chữ, thiết kế tự động và cá nhân hoá chữ viết xuyên biên giới.

## 1.4. Đối tượng và phạm vi nghiên cứu

Để đảm bảo tính khả thi và tập trung sâu vào giải pháp kỹ thuật, đề tài xác định rõ đối tượng và giới hạn phạm vi nghiên cứu như sau:

### 1.4.1. Đối tượng nghiên cứu

**Mô hình lý thuyết và phát triển:** Trọng tâm nghiên cứu được đặt vào **Mô hình sinh ảnh dựa trên cơ chế khuếch tán (Diffusion Models)**, lấy kiến trúc FontDiffuser làm nền tảng cốt lõi để cải tiến. Đề tài tập trung nghiên cứu các kỹ thuật **điều hướng phong cách (Style Guidance)** và cơ chế **học tương phản (Contrastive Learning)** trong không gian khuếch tán nhằm giải quyết bài toán chuyển đổi đa ngôn ngữ.

**Mô hình đối chứng (Baseline):** Để thiết lập một hệ quy chiếu đánh giá toàn diện và làm nổi bật ưu thế của phương pháp đề xuất, khoá luận thực hiện so sánh với **hai nhóm phương pháp hiện có**. Nhóm thứ nhất bao gồm **các phương pháp dựa trên GAN[1] tiên tiến** như **DG-Font[5], CF-Font[6], DFS[15]** và **FTransGAN[16]**, nhằm chứng minh khả năng vượt trội của mô hình Khuếch tán trong việc tạo ra hình

ánh chất lượng cao và ổn định. Nhóm thứ hai là **mô hình FontDiffuser nguyên bản**[4], được sử dụng để đối sánh trực tiếp nhằm định lượng chính xác hiệu quả đóng góp của các cải tiến kỹ thuật được đề xuất trong khoá luận (như mô-đun CL-SCR) so với thuật toán ban đầu.

**Đổi tượng dữ liệu:** Khoá luận sử dụng **hai hệ chữ viết có đặc trưng hình thái đối lập**. Hệ chữ nguồn bao gồm các bộ phông chữ chứa ký tự Hán (theo chuẩn GB2312) với độ phức tạp cấu trúc đa dạng. Đối ứng với đó là hệ chữ đích gồm bộ 52 ký tự tiếng Anh cơ bản (26 chữ hoa và 26 chữ thường) thuộc hệ Latin.

#### 1.4.2. Phạm vi nghiên cứu

**Phạm vi về ngôn ngữ:** Đề tài tập trung nghiên cứu và thực nghiệm trên bài toán **chuyển đổi phong cách hai chiều (Bidirectional Transfer)** giữa Tiếng Anh (Latin) và Tiếng Trung Quốc (Hán). Việc lựa chọn cặp ngôn ngữ này nhằm giải quyết hai thách thức bổ trợ cho nhau. Ở hướng Latin sang Hán tự (e2c), thách thức nằm ở việc ngoại suy phong cách từ một hệ chữ đơn giản, cấu trúc thưa sang một hệ chữ phức tạp hơn rất nhiều, đòi hỏi mô hình phải học cách áp dụng phong cách lên các cấu trúc dày đặc mà không làm vỡ nét. Ngược lại, ở hướng Hán tự sang Latin (c2e), thách thức nằm ở việc trích xuất phong cách từ hệ chữ nhiều chi tiết để áp dụng lên hệ chữ đơn giản, buộc mô hình phải có khả năng tổng quát hoá cao để lọc bỏ các nhiễu cấu trúc.

Khoá luận tập trung vào bài toán chuyển đổi xuyên ngôn ngữ thay vì các tác vụ đơn ngữ (như Latin sang Latin hay Hán tự sang Hán tự) bởi lẽ bài toán đơn ngữ hiện nay đã đạt được trạng thái bão hòa với nhiều giải pháp tối ưu, trong khi bài toán đa ngôn ngữ vẫn tồn tại một “khoảng cách miền” (domain gap) lớn chưa được giải quyết triệt để. Việc chuyển đổi giữa hai hệ chữ có cấu trúc đối lập như Latin và Hán tự đặt ra những thách thức khoa học cao hơn về khả năng tách biệt (disentanglement) và suy diễn phong cách bất biến, từ đó chứng minh được tính ưu việt và khả năng tổng quát hoá của thuật toán đề xuất so với việc chỉ hoạt động trong một không gian hình thái đóng kín.

**Phạm vi về bài toán:** Khoá luận tập trung vào **bài toán One-shot Generation**, trong đó mô hình chỉ được cung cấp một ký tự duy nhất làm ảnh tham chiếu phong cách (Style Reference) để sinh ra ký tự mục tiêu mang nội dung khác. Cụ thể, một ký tự Latin sẽ được dùng để định hình phong cách cho một Hán tự ở chiều xuôi, và một ký tự Hán sẽ được dùng để định hình phong cách cho một chữ cái Latin ở chiều ngược.

Mặc dù việc cung cấp nhiều mẫu tham chiếu (few-shot hoặc full-shot) về lý thuyết sẽ cung cấp cho mô hình nhiều thông tin hơn để tái tạo phong cách chính xác, nhưng cách tiếp cận One-shot (chỉ sử dụng một ảnh mẫu duy nhất) được lựa chọn nhằm tối đa hóa tính ứng dụng thực tiễn và giảm thiểu chi phí lao động cho nhà thiết kế. Trong các kịch bản thực tế, việc yêu cầu người dùng phải vẽ thủ công một lượng lớn ký tự mẫu để khởi chạy hệ thống sẽ đi ngược lại mục tiêu tự động hóa của bài toán sinh phông chữ; do đó, thách thức và đóng góp chính của khoá luận nằm ở việc xây dựng một mô hình đủ mạnh để trích xuất và tổng quát hóa phong cách phức tạp chỉ từ lượng thông tin đầu vào cực tiểu.

**Phạm vi về dữ liệu:** Sử dụng **các bộ dữ liệu phông chữ mã nguồn mở hỗ trợ đồng thời cả hai bảng mã**, ví dụ như các bộ font thuộc dự án Google Noto CJK hoặc các font nghệ thuật song ngữ. Việc lựa chọn các nguồn dữ liệu song ngữ này không chỉ đảm bảo tính đa dạng về phong cách mà còn đóng vai trò then chốt trong việc thiết lập cơ chế giám sát thực nghiệm: chúng cung cấp các cặp dữ liệu đối chứng (Paired Ground Truth) chính xác — tức cùng một bộ font luôn tồn tại song song cả ký tự Latin và Hán tự tương ứng. Điều này cho phép mô hình tính toán được các hàm mất mát tái tạo (Reconstruction Loss) một cách tin cậy, khắc phục hạn chế của các phương pháp học không giám sát trên dữ liệu không ghép cặp (unpaired data).

Hơn nữa, việc lựa chọn miền dữ liệu này tuân thủ nghiêm ngặt các thiết lập thực nghiệm chuẩn (Standardized Experimental Settings) đã được xác lập bởi các công trình tiên phong như FontDiffuser[4], DG-Font[5], và FTransGAN[16]. Trong nghiên cứu so sánh, việc đồng bộ hóa dữ liệu đầu vào với các phương pháp SOTA (State-of-the-art) là điều kiện tiên quyết để đảm bảo tính công bằng và trực diện (fair and direct comparison) khi đánh giá hiệu năng. Việc kế thừa hệ quy chiếu dữ liệu này giúp loại bỏ các yếu tố nhiễu do sự khác biệt về phân phối dữ liệu, từ đó khẳng định rằng mọi sự cải thiện về chỉ số định lượng (như FID hay LPIPS) đều phản ánh chính xác ưu việt của thuật toán đề xuất thay vì lợi thế của dữ liệu.

## 1.5. Cấu trúc của khoá luận

Phần còn lại của khoá luận này được trình bày như sau:

**Chương 2 – Cơ sở lý thuyết.**

Trình bày các khái niệm nền tảng về bài toán sinh font chữ. Đồng thời, chương này tổng hợp và phân tích các phương pháp sinh font trước đây, bao gồm nhóm mô hình dựa trên GAN[1] (DG-Font[5], CF-Font[6], DFS[15], FTransGAN[16]) và nhóm mô hình khuếch tán[2] (FontDiffuser[4]), chỉ ra ưu nhược điểm và xu hướng phát triển.

### **Chương 3 – Phương pháp đề xuất.**

Trình bày chi tiết pipeline gốc của FontDiffuser[4], bao gồm hai giai đoạn huấn luyện (Giai đoạn 1 – Tái tạo cấu trúc, Giai đoạn 2 – Tinh chỉnh phong cách). Phân tích cơ chế hoạt động của các mô-đun chính như MCA (Multi-scale Content Aggregation), RSI (Reference-Structure Interaction) và SCR (Style Contrastive Refinement). Trên cơ sở đó, chương này giới thiệu ý tưởng cải tiến nhằm mở rộng khả năng chuyển phong cách đa ngôn ngữ (cross-lingual style transfer) thông qua việc thay thế và điều chỉnh mô-đun SCR.

### **Chương 4 – Thực nghiệm và Đánh giá kết quả.**

Chương này mô tả chi tiết quy trình thiết lập thực nghiệm, bao gồm việc xây dựng tập dữ liệu đa ngôn ngữ (Latin–Hán), cấu hình huấn luyện và các tiêu chí đánh giá được sử dụng (FID[17], SSIM[18], LPIPS[19], L1, User Study). Đồng thời, chương trình bày các kết quả định lượng, định tính và đánh giá của con người, so sánh mô hình đề xuất (FontDiffuser + CL-SCR) với các mô hình nền tảng (GAN-based và Diffusion-based). Phần phân tích chuyên sâu sẽ đánh giá hiệu quả của mô-đun CL-SCR, nghiên cứu Ablation về các thành phần cải tiến, và thảo luận về ưu điểm, hạn chế cũng như ảnh hưởng của các tham số then chốt (như số lượng mẫu âm, Guidance Scale) đối với khả năng chuyển phong cách đa ngôn ngữ.

### **Chương 5 – Kết luận và Hướng phát triển.**

Tóm tắt toàn bộ đóng góp chính của khoá luận, bao gồm việc tái hiện pipeline FontDiffuser và đề xuất CL-SCR cho cross-lingual font generation. Đề xuất các hướng nghiên cứu mở rộng, như mở rộng sang nhiều ngôn ngữ hơn (tiếng Việt, tiếng Nhật, tiếng Ả Rập), và áp dụng parameter-efficient fine-tuning (như LoRA[20] hoặc Adapter[21]) để tối ưu tài nguyên huấn luyện.

### **Phụ lục – Trình bày phụ lục của khoá luận.**

# Chương 2

## Cơ sở lý thuyết

Trong chương này, khoá luận trình bày hệ thống cơ sở lý thuyết nền tảng về các mô hình sinh (Generative Models) và tổng quan tình hình nghiên cứu trong lĩnh vực sinh phông chữ tự động. Cấu trúc chương đi từ các phương pháp truyền thống dựa trên GAN[1], đến sự trỗi dậy của Mô hình khuếch tán (Diffusion Models)[2]. Đồng thời, phần cuối chương sẽ tập trung phân tích sâu về các kỹ thuật biểu diễn phong cách (Style Representation) và những thách thức đặc thù trong bài toán chuyển đổi đa ngôn ngữ, nhằm làm rõ động lực nghiên cứu cho phương pháp đề xuất tại Chương 3.

### 2.1. Một số phương pháp tiếp cận cho bài toán Sinh phông chữ (Font Generation)

Lĩnh vực sinh phông chữ (Font Generation) đã trải qua một sự chuyển dịch mạnh mẽ về mặt công nghệ trong thập kỷ qua. Các phương pháp hiện nay có thể được chia thành hai nhóm chính dựa trên mô hình lõi: Mạng đối nghịch sinh (GANs)[1] và Mô hình khuếch tán (Diffusion Models)[2].

#### 2.1.1. Các phương pháp dựa trên GAN (Generative Adversarial Networks)

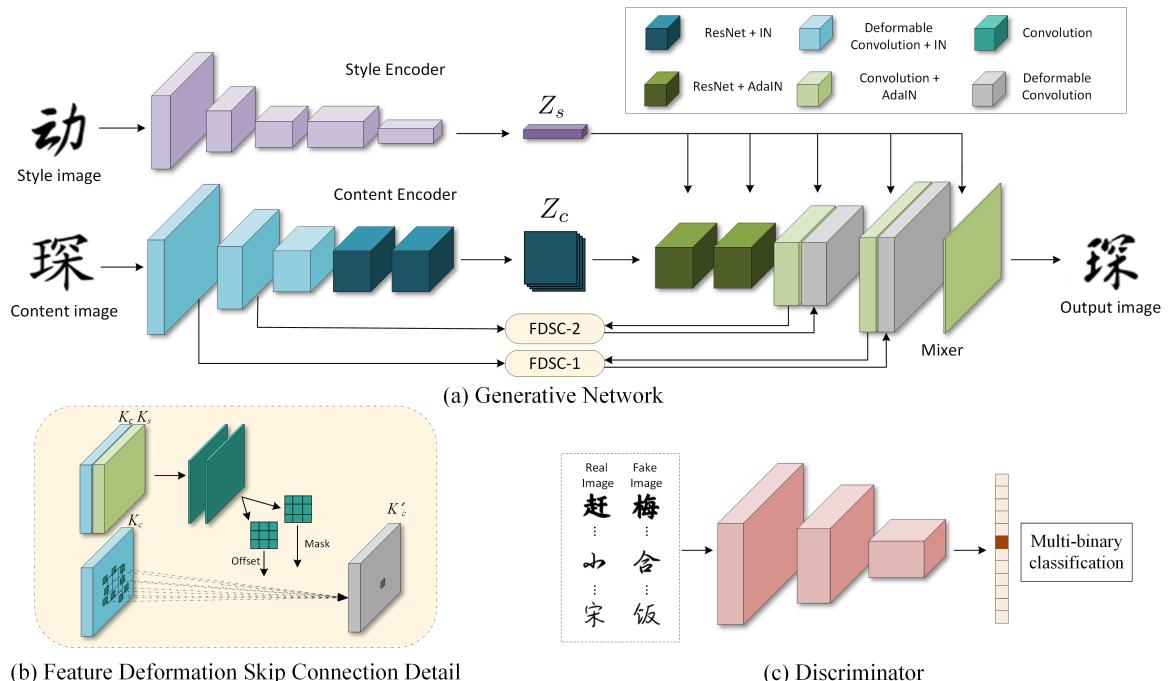
Trước sự bùng nổ của Diffusion Models[2] vào năm 2023, Generative Adversarial Networks (GAN)[1] là hướng tiếp cận chủ đạo (State-of-the-art) cho bài toán này. Các nghiên cứu GAN thường tập trung giải quyết vấn đề tách biệt nội dung (content) và phong cách (style).

##### 2.1.1.1. DG-Font (Deformable Generative Network, CVPR 2021)

DG-Font[5] tiếp cận bài toán sinh phông chữ theo hướng **không giám sát (unsupervised)**, tập trung giải quyết thách thức về sự sai lệch hình học lớn giữa phông chữ nguồn và phông chữ đích mà các phương pháp chuyển đổi phong cách dựa trên texture thông thường thường thất bại. Thay vì sử dụng dữ liệu cặp (paired data) vốn

kém, DG-Font đề xuất một kiến trúc mới cho phép học ánh xạ phong cách trực tiếp từ các tập dữ liệu không gán nhãn.

Đóng góp cốt lõi của mô hình là mô-đun **Feature Deformation Skip Connection (FDSC)**. Cơ chế này hoạt động bằng cách dự đoán các bản đồ dịch chuyển (displacement maps) từ đặc trưng nội dung và phong cách, sau đó áp dụng **tích chập biến dạng (deformable convolution)** lên các đặc trưng cấp thấp. Điều này cho phép mô hình “uốn nắn” cấu trúc không gian của ký tự nguồn sao cho khớp với dáng vẻ của ký tự đích trước khi đưa vào bộ trộn (Mixer) để sinh ảnh cuối cùng. Mặc dù đạt hiệu quả cao trong việc bảo toàn cấu trúc, DG-Font vẫn tồn tại nhược điểm có hưu của dòng GAN là sự mất ổn định khi huấn luyện; đối với các ký tự có sự biến đổi topo học quá lớn (ví dụ từ nét thanh sang nét đậm phá cách), ảnh sinh ra dễ bị hiện tượng đứt nét (broken strokes) hoặc mờ nhòe.



**Hình 2.1** — Kiến trúc mạng DG-Font. Mô-đun FDSC đóng vai trò nòng cốt trong việc học biến dạng hình học giữa các ký tự. (Trích từ [5])

### 2.1.1.2. CF-Font (Content Fusion, CVPR 2023)

CF-Font[6] tiếp cận bài toán sinh phông chữ few-shot theo hướng “**lai ghép**” **nội dung (content fusion)**, khắc phục hạn chế của các phương pháp truyền thống vốn chỉ dựa vào một font nguồn (source font) duy nhất. Nhận định rằng sự chênh lệch cấu

trúc (topology) giữa font nguồn và font đích là nguyên nhân chính gây ra các lỗi biến dạng, CF-Font đề xuất sử dụng một tập hợp các **font cơ sở (basis fonts)** tiêu chuẩn để làm “nguyên liệu” tham chiếu.

Đóng góp cốt lõi của nghiên cứu là mô-đun **Content Fusion Module (CFM)**. Cơ chế này hoạt động bằng cách dự đoán bộ trọng số nhiệt (fusion weights) để **tổ hợp tuyến tính** các đặc trưng nội dung từ các font cơ sở. Thay vì phải “uốn nắn” khó khăn từ một hình dạng cố định, mô hình có thể linh hoạt pha trộn các đặc điểm hình học từ nhiều nguồn khác nhau để tạo ra một “khung xương” nội dung tiệm cận nhất với font mục tiêu. Chiến lược này giúp giảm thiểu đáng kể việc mất mát thông tin cấu trúc, tuy nhiên cũng đánh đổi bằng chi phí tính toán cao hơn do phải xử lý đa luồng đầu vào. Ngoài ra, nếu tập font cơ sở không đủ bao quát không gian topo, ảnh sinh ra có thể xuất hiện các vết mờ hoặc bóng ma (ghosting artifacts) tại các vùng giao thoa nét.



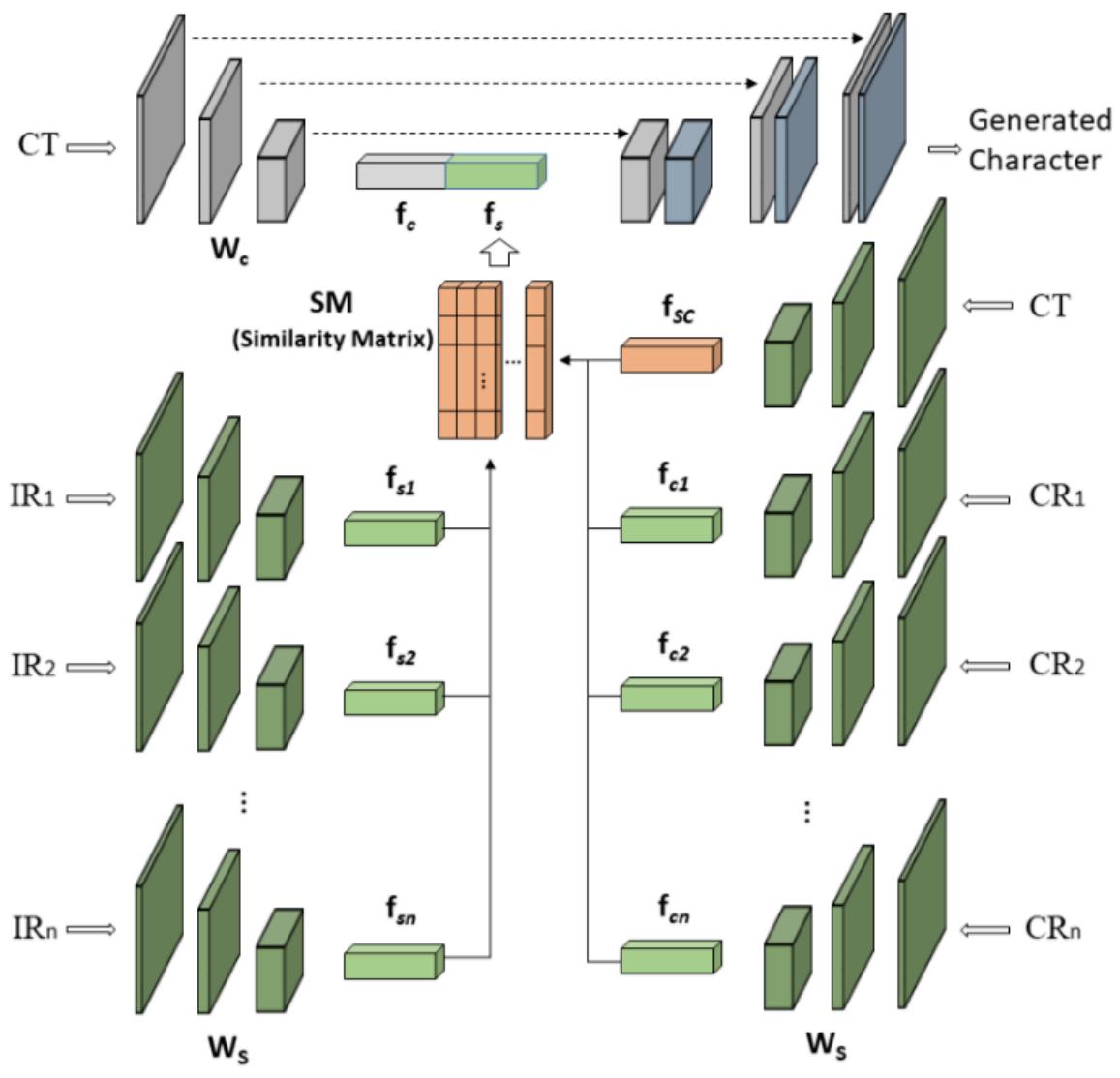
**Hình 2.2** — Minh họa cơ chế Content Fusion: Các đặc trưng từ tập font cơ sở (Source) được tổ hợp tuyến tính dựa trên bộ trọng số dự đoán (Weights) để xấp xỉ cấu trúc hình học của font mục tiêu. (Trích từ [6])

### 2.1.1.3. DFS (Few-Shot Text Style Transfer via Deep Feature Similarity, TIP 2020)

DFS[15] đề xuất một cách tiếp cận mới cho bài toán chuyển đổi phong cách few-shot bằng cách khai thác mối tương quan cấu trúc giữa các ký tự. Khác với các phương pháp trước đó thường nén toàn bộ thông tin phong cách vào một vector duy nhất, DFS trích xuất đặc trưng từ từng ảnh tham chiếu riêng biệt thông qua mạng CNN. Đóng góp quan trọng nhất của mô hình là cơ chế **Deep Feature Similarity**, trong đó một **Ma trận Tương đồng (Similarity Matrix - SM)** được tính toán dựa trên **độ tương quan (cross-correlation)** giữa đặc trưng nội dung của ký tự tham chiếu và ký tự mục tiêu. Các đặc trưng style đã được điều chỉnh sau đó được gộp lại và nối với đặc trưng

content, rồi đưa qua decoder đôi xứng dạng U-Net để tái tạo ký tự đích trong phong cách mong muốn. Mô hình được huấn luyện end-to-end với LSGAN[22] loss kết hợp loss tái tạo, cho phép sinh ảnh có độ chân thực cao hơn so với các phương pháp chỉ dùng CNN thuần tuý.

Cơ chế này hoạt động như một bộ lọc chú ý thông minh: nó cho phép mô hình tự động **gán trọng số lớn hơn cho các ký tự tham chiếu có cấu trúc hình học tương đồng** với ký tự cần sinh (ví dụ: sử dụng nét cong của chữ ‘O’ để hỗ trợ sinh chữ ‘Q’ hoặc ‘C’). Sau đó, các đặc trưng phong cách được trọng số hoá này sẽ được trộn (mix) với đặc trưng nội dung để giải mã thành ảnh kết quả. Mặc dù đạt được độ chính xác cao về chi tiết phong cách nhờ việc “chọn lọc” thông tin, DFS vẫn tồn tại nhược điểm là yêu cầu quá trình **tinh chỉnh (fine-tuning)** cho từng phong cách mới (leave-one-out strategy) để đạt kết quả tối ưu, làm hạn chế khả năng ứng dụng thời gian thực so với các mô hình suy diễn trực tiếp (feed-forward).



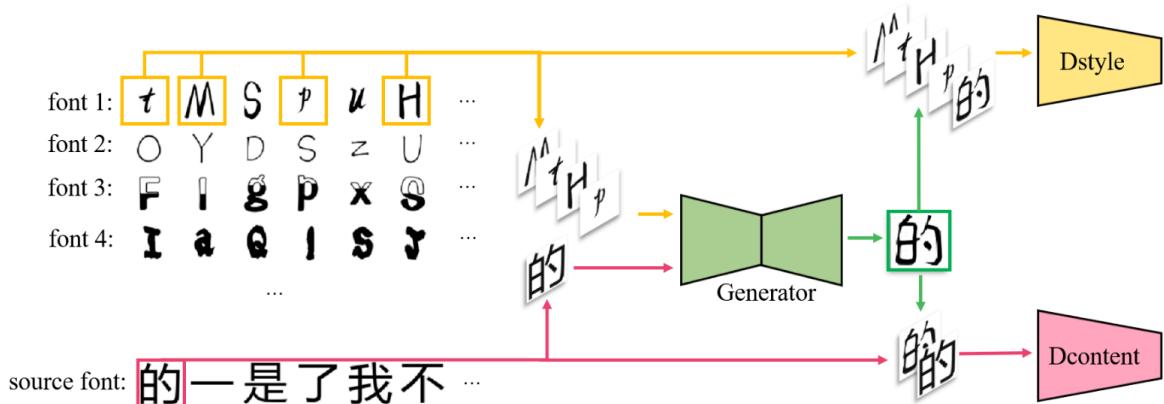
IR—referenced character	CR—content image of referenced character	CT—content image of target character
IR <sub>1</sub>	CR <sub>1</sub>	CT
IR <sub>2</sub>	CR <sub>2</sub>	Generated Character
IR <sub>3</sub>	CR <sub>3</sub>	
IR <sub>4</sub>	CR <sub>4</sub>	

**Hình 2.3** — Kiến trúc mạng DFS với thành phần cốt lõi là Ma trận Tương đồng (SM) giúp điều hướng dòng chảy thông tin phong cách. (Trích từ [15])

#### 2.1.1.4. FTransGAN (Few-shot Font Style Transfer between Different Languages, WACV 2021)

FTransGAN[16] là một trong những mô hình tiên phong giải quyết bài toán **chuyển đổi phong cách phông chữ đa ngôn ngữ (cross-lingual)** theo hướng few-shot learning. Khác với các phương pháp trước đó thường chỉ tập trung vào chuyển đổi trong cùng một ngôn ngữ, FTransGAN đề xuất một kiến trúc end-to-end cho phép trích xuất thông tin phong cách từ một ngôn ngữ (ví dụ: tiếng Anh) và áp dụng lên nội dung của ngôn ngữ khác (ví dụ: tiếng Trung).

Để giải quyết sự chênh lệch lớn về cấu trúc giữa các hệ chữ viết, FTransGAN thiết kế bộ mã hoá phong cách (Style Encoder) đặc biệt với **cơ chế chú ý đa tầng (multi-level attention)**. Kiến trúc này bao gồm hai mô-đun chính: **Context-aware Attention Network** giúp nắm bắt các đặc trưng cục bộ (như nét bút, hoạ tiết trang trí) và **Layer Attention Network** giúp tổng hợp các đặc trưng toàn cục để quyết định mức độ ưu tiên giữa các tầng đặc trưng khác nhau. Nhờ đó, mô hình có khả năng tạo ra các phông chữ chất lượng cao mà **không cần quá trình tinh chỉnh (fine-tuning)** phức tạp cho từng style mới. Tuy nhiên, FTransGAN vẫn còn hạn chế khi xử lý các phông chữ có tính nghệ thuật quá cao hoặc cấu trúc biến dạng mạnh, đồng thời yêu cầu số lượng ảnh phong cách đầu vào cố định trong quá trình huấn luyện.



Hình 2.4 — Tổng quan kiến trúc FTransGAN. (Trích từ [16])

#### 2.1.2. Mô hình khuếch tán (Diffusion Models)

Gần đây, Mô hình khuếch tán[2] (Diffusion Models) đã tạo nên một cuộc cách mạng trong lĩnh vực thị giác máy tính. Khác với GAN[1] – vốn dựa trên việc lừa mô hình

phân biệt, Diffusion Model mô phỏng quá trình nhiệt động lực học để biến đổi dần dần từ nhiễu sang dữ liệu có ý nghĩa. Trong phạm vi khoá luận này, khoá luận tập trung vào Mô hình Khuếch tán Khử nhiễu Xác suất (Denoising Diffusion Probabilistic Models - DDPM)[3], biến thể phổ biến nhất và là nền tảng của phương pháp FontDiffuser.

Nguyên lý cơ bản gồm hai giai đoạn:

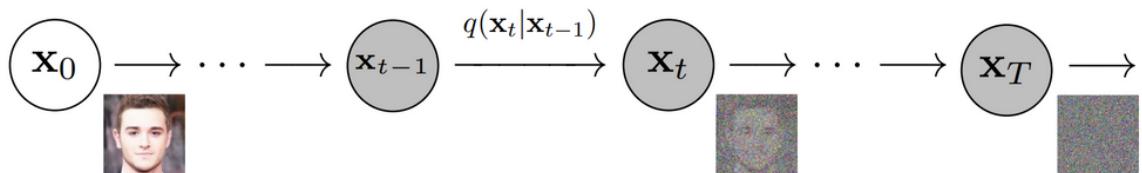
**Quá trình Khuếch tán xuôi:** phá huỷ dữ liệu một cách có kiểm soát bằng cách thêm nhiễu Gaussian nhiều bước.

**Quá trình Khuếch tán ngược:** học cách loại bỏ nhiễu từng bước để tái tạo lại dữ liệu gốc.

Điều này tương tự như việc ta học cách “tô dần” một bức tranh từ nền trắng nhiễu cho đến khi ra ảnh rõ nét.

### 2.1.2.1. Quá trình Khuếch tán xuôi (Forward Diffusion Process)

Trong quá trình này, nhiễu được thêm dần vào dữ liệu qua một loạt các bước. Điều này tương tự như chuỗi Markov, trong đó mỗi bước **phá huỷ dần cấu trúc dữ liệu** bằng cách thêm nhiễu Gauss:



**Hình 2.5** — Quá trình Khuếch tán xuôi. (Trích từ [23])

Về mặt toán học, xác suất chuyển trạng thái được biểu diễn như sau:

$$q(x_t|x_{t-1}) = N(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t \mathbf{I}) \quad (2.2)$$

Trong đó:

$x_0$ : ảnh gốc (clean image).

$x_t$ : ảnh ở bước  $t$  sau khi thêm nhiễu.

$\beta_t$ : hệ số nhiễu nhỏ (thường  $\beta_t \in [10^{-4}, 0.02]$ ).

$\mathbf{I}$ : ma trận đơn vị, đảm bảo nhiễu độc lập và đồng hướng.

Do tính chất cộng tính của phân phối Gaussian, ta có thể lấy mẫu trực tiếp  $x_t$  từ  $x_0$ :

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, \quad \epsilon \sim N(0, \mathbf{I}) \quad (2.3)$$

Trong đó:

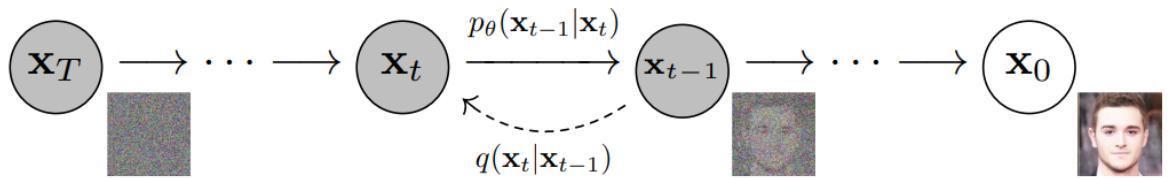
$$\alpha_t = 1 - \beta_t \quad (2.4)$$

$$\bar{\alpha}_t = \prod_{s=1}^t \alpha_s \quad (2.5)$$

Tính chất này rất quan trọng vì nó cho phép huấn luyện song song tại bất kỳ bước thời gian  $t$  nào mà không cần sinh tuần tự từng bước.

### 2.1.2.2. Quá trình Khuếch tán ngược (Reverse Diffusion Process)

Quá trình này nhằm mục đích tái tạo lại dữ liệu gốc bằng cách khử nhiễu thông qua một loạt các bước đảo ngược.



**Hình 2.6** — Quá trình Khuếch tán ngược. (Trích từ [23])

Về mặt toán học, xác suất chuyển trạng thái ngược được xấp xỉ bởi một phân phối Gauss với tham số được học:

$$p_\theta(x_{t-1}|x_t) = N(x_{t-1}; \mu_\theta(x_t, t), \sigma_t^2 \mathbf{I}) \quad (2.6)$$

Trong DDPM, phương sai  $\sigma_t^2$  thường được cố định là hằng số ( $\beta_t$  hoặc  $\tilde{\beta}_t$ ). Giá trị trung bình  $\mu_\theta$  được tham số hóa bởi mạng nơ-ron như sau:

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}} \left( x_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right) \quad (2.7)$$

Quá trình lấy mẫu thực tế (Sampling) để thu được  $x_{t-1}$  sẽ bao gồm thêm một lượng nhiễu ngẫu nhiên  $z$  (ngoại trừ bước cuối cùng  $t = 1$ ):

$$x_{t-1} = \mu_\theta(x_t, t) + \sigma_t z, \quad z \sim N(0, \mathbf{I}) \quad (2.8)$$

Trong huấn luyện, mô hình được tối ưu để giảm sai số giữa nhiều dự đoán  $\epsilon_\theta(x_t, t)$  và nhiều thực tế  $\epsilon$  đã thêm vào ở quá trình xuôi.

### 2.1.2.3. Hàm mất mát (Loss function)

Hàm mất mát được sử dụng phổ biến nhất là **Mean Squared Error (MSE)** trên không gian nhiễu:

$$L_{\text{simple}} = \mathbb{E}_{t, x_0, \epsilon} [\| \epsilon - \epsilon_\theta(x_t, t) \|^2] \quad (2.9)$$

Điều này tương đương với việc tối đa hóa cận dưới biến phân (variational lower bound) của khả năng sinh dữ liệu. Mặc dù các nghiên cứu gần đây đề xuất dự đoán  $v_t$  hoặc  $x_0$ , nhưng việc dự đoán nhiễu ( $\epsilon$ -prediction) kết hợp với hàm loss MSE đơn giản vẫn là chuẩn mực hiệu quả được sử dụng trong FontDiffuser.

### 2.1.2.4. FontDiffuser (AAAI 2024)

FontDiffuser[4] là công trình tiên phong áp dụng thành công Diffusion Model vào bài toán One-shot Font Generation. Pipeline của mô hình giải quyết ba vấn đề cốt lõi:

**Bảo toàn cấu trúc:** Sử dụng khối **MCA (Multi-Scale Content Aggregation)** để tổng hợp thông tin cấu trúc từ toàn cục đến chi tiết.

**Xử lý biến dạng:** Sử dụng khối **RSI (Reference-Structure Interaction)** thay thế cho các phương pháp biến dạng cũ, giúp tương thích tốt hơn giữa cấu trúc ảnh nguồn và phong cách ảnh đích.

**Học phong cách:** Sử dụng mô-đun **SCR (Style Contrastive Refinement)** để tinh chỉnh biểu diễn phong cách.

Đây chính là mô hình cơ sở (baseline) mà khoá luận này lựa chọn để kế thừa và phát triển.

## 2.2. Một số phương pháp tiếp cận cho bài toán Biểu diễn Phong cách (Style Representation)

Trong bài toán sinh phông chữ One-shot, đặc biệt là trong bối cảnh chuyển đổi đa ngôn ngữ (Cross-Lingual), việc trích xuất và biểu diễn chính xác “phong cách” (style) là yếu tố quyết định sự thành bại của mô hình.

Khái niệm “phong cách” (style) được định nghĩa là tập hợp các đặc trưng thị giác phi cấu trúc và bất biến về mặt ngữ nghĩa, bao gồm các thuộc tính hình học cục bộ như độ dày nét (stroke weight), kiểu dáng chân chữ (serif/sans-serif), độ tương phản nét (stroke contrast), cũng như các đặc tính bề mặt như độ xước hay kết cấu mực in. Khác với “nội dung” (content) quy định cấu trúc tô-pô học để nhận diện ký tự, phong cách đóng vai trò là lớp vỏ thẩm mỹ nhất quán xuyên suốt toàn bộ bộ chữ, đòi hỏi mô hình phải có khả năng tách biệt và tái áp dụng các đặc trưng này lên các khung xương cấu trúc hoàn toàn khác biệt mà không làm thay đổi định danh của ký tự đó.

Để chuyển hóa các khái niệm phong cách thị giác mang tính định tính này thành tín hiệu số học mà máy tính có thể xử lý, bài toán thường tiếp cận theo cơ chế ánh xạ đa tầng thông qua các hàm mất mát. Cụ thể, các thuộc tính vật lý cấp thấp như độ dày nét hay cấu trúc hình học được định lượng gián tiếp thông qua sai số điểm ảnh (L1 Loss) và độ tương đồng cấu trúc (SSIM); ví dụ, sự chênh lệch mật độ điểm ảnh giữa nét dày và nét mảnh sẽ tạo ra giá trị sai số lớn, buộc mô hình phải điều chỉnh trọng số để khớp độ dày. Trong khi đó, các thuộc tính trừu tượng cấp cao như kết cấu bề mặt (texture) hay độ xước bút pháp được định lượng thông qua khoảng cách trong không gian đặc trưng sâu (Deep Feature Space) của các mạng nơ-ron tích chập (như VGG). Tại đây, các bộ lọc tích chập đóng vai trò như các “cảm biến” phát hiện hoạ tiết, và độ tương đồng Cô-sin giữa các vector đặc trưng chính là “thước đo” mức độ khớp về phong cách nghệ thuật.

### 2.2.1. Neural Style Transfer truyền thống

Các phương pháp sơ khai (như Gatys et al.[24]) thường sử dụng Ma trận Gram (Gram Matrix) tính toán trên các bản đồ đặc trưng (feature maps) của mạng VGG pre-trained để định nghĩa phong cách. Tuy nhiên, phương pháp này chủ yếu nắm bắt các đặc trưng về chất liệu (texture) và hoạ tiết cục bộ. Đôi với ký tự, “phong cách” không chỉ là vân bề mặt mà còn bao gồm các yếu tố hình học cấp cao như: độ gãy khúc, kiểu chân chữ (serif/sans-serif), và cách kết thúc nét (stroke ending). Gram Matrix thường thất bại trong việc hướng dẫn mô hình áp dụng các đặc trưng này lên các cấu trúc hình học mới, dẫn đến kết quả bị biến dạng hoặc chỉ đơn thuần là phủ texture lên ảnh nội dung.

### 2.2.2. Học tương phản (Contrastive Learning)

Để khắc phục hạn chế trên, các nghiên cứu hiện đại (trong đó có FontDiffuser[4]) chuyển sang hướng **Học biểu diễn tương phản (Contrastive Representation)**

**Learning).** Tư tưởng cốt lõi là học một không gian embedding phong cách (style latent space) sao cho **các mẫu có cùng phong cách** (Positive samples) được *kéo lại gần nhau* và **các mẫu khác phong cách** (Negative samples) bị *đẩy ra xa nhau*.

Hàm mát mẻ InfoNCE[25] thường được sử dụng để tối ưu hoá không gian này:

$$L_{\text{NCE}} = -\log \left( \frac{\exp(\text{sim}(z, z^+)/\tau)}{\exp(\text{sim}(z, z^+)/\tau) + \sum_k \exp(\text{sim}(z, z_k^-)/\tau)} \right) \quad (2.10)$$

Trong đó:

$z$ : Vector đặc trưng (feature representation) hoặc biểu diễn tiềm ẩn của mẫu dữ liệu đang xét (thường được gọi là mẫu neo - anchor).

$z^+$ : Biểu diễn đặc trưng của mẫu dương (positive sample) – đây là mẫu tương đồng hoặc thuộc cùng một lớp với  $z$  mà mô hình cần học để tối đa hóa độ tương đồng.

$z_k^-$ : Biểu diễn đặc trưng của mẫu âm (negative sample) thứ  $k$  trong tập dữ liệu – đây là các mẫu khác lớp hoặc không tương đồng với  $z$  mà mô hình cần phân biệt và đẩy xa trong không gian đặc trưng.

$\text{sim}(\cdot, \cdot)$ : Hàm đo độ tương đồng giữa hai vector (similarity function), trong ngữ cảnh này thường là độ tương đồng Cô-sin (Cosine Similarity), tính toán góc giữa hai vector trong không gian đặc trưng.

$\tau$ : Tham số nhiệt độ (temperature parameter), là một siêu tham số (hyperparameter) dương giúp điều chỉnh độ nhạy của hàm mát mẻ. Giá trị  $\tau$  nhỏ giúp mô hình tập trung hơn vào các mẫu âm khó phân biệt (hard negatives), trong khi  $\tau$  lớn giúp quá trình huấn luyện mượt mà hơn.

$\sum_k$ : Phép tổng thực hiện trên tất cả các mẫu âm (negative samples) có trong batch hoặc hàng đợi (queue) hiện tại.

Trong FontDiffuser, mô-đun SCR áp dụng tư tưởng này để giám sát bộ mã hoá phong cách. Tuy nhiên, mô-đun này ban đầu được thiết kế cho cùng một ngôn ngữ (Hán → Hán). Khi mở rộng sang bài toán Cross-Lingual (**chuyển đổi qua lại giữa Latin và Hán**), sự khác biệt quá lớn về cấu trúc giữa hai hệ chữ tạo ra một khoảng cách miền (domain gap) sâu sắc, khiến các chiến lược chọn mẫu âm (negative selection) thông thường trở nên kém hiệu quả.

## 2.3. Thách thức trong bài toán Cross-Lingual: Chuyển đổi Hai chiều giữa Latin và Hán tự

Khác với các hướng tiếp cận truyền thống thường chỉ tập trung vào một chiều chuyển đổi đơn lẻ, khoá luận giải quyết bài toán tổng quát và thách thức hơn là chuyển đổi phong cách hai chiều (Bidirectional Style Transfer) giữa hệ chữ Latin và Hán tự. Sự khác biệt nền tảng giữa hai hệ chữ này đặt ra những rào cản kỹ thuật đặc thù cho từng hướng chuyển đổi, chủ yếu xoay quanh sự bất đối xứng về thông tin và hình thái học.

### 2.3.1. Vấn đề Chênh lệch Mật độ Thông tin (Information Density Asymmetry)

Một thách thức cốt lõi trong bài toán chuyển giao phong cách giữa chữ Latin và Hán tự bắt nguồn từ *sự bất cân xứng nghiêm trọng về mật độ thông tin hình học*. Theo quan điểm thiết kế chữ Hán[26], mỗi Hán tự được tổ chức như một **khối vuông khép kín**, nơi các nét bút không chỉ mang ý nghĩa hình thức mà còn có vai trò *phân bố* và *lắp đầy không gian thị giác*. Ngược lại, chữ Latin được xây dựng trên tư duy tuyến tính, với cấu trúc mở và số lượng nét tối thiểu, chỉ đủ để gợi hình dạng ký tự.

Sự khác biệt này dẫn đến hai bài toán đối nghịch trong quá trình chuyển đổi song hướng. Ở chiều **Latin → Hán tự** (e2c), mô hình phải đổi mới với bài toán ngoại suy (extrapolation): từ các ký tự Latin có mật độ thông tin cực thấp (ví dụ như I hoặc 0), hệ thống phải suy diễn và mở rộng phong cách để phù hợp với các Hán tự có cấu trúc dày đặc và nhiều tầng nét bút (ví dụ 龍). Trong trường hợp thiếu cơ chế suy luận phong cách hiệu quả, mô hình không thể “tưởng tượng” cách phân bố phong cách vào không gian khối vuông, dẫn đến các lỗi phổ biến như nét bị dính bết, phân bố không đều hoặc mất chi tiết cấu trúc.

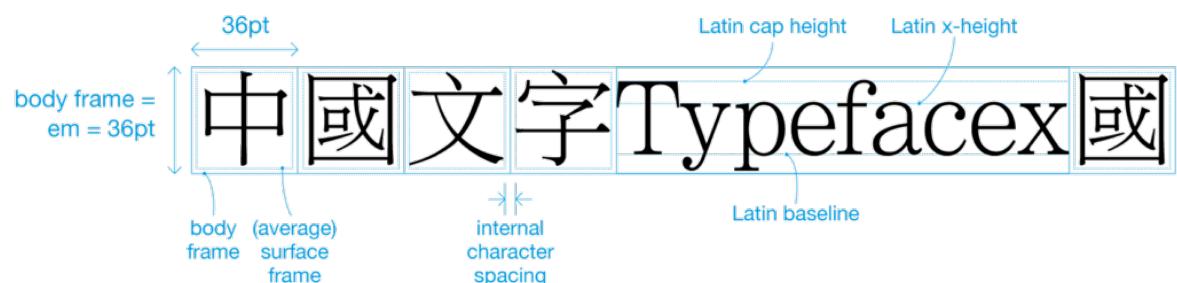
Ngược lại, ở chiều **Hán tự → Latin** (c2e), vấn đề chuyển thành bài toán *trùu tượng hoá* (abstraction). Ảnh nguồn Hán tự chứa lượng thông tin thị giác dư thừa so với khả năng biểu đạt của chữ Latin. Do đó, thách thức không nằm ở việc sinh thêm chi tiết, mà ở việc **loại bỏ có chọn lọc** các yếu tố cấu trúc gắn với nội dung (strokes, bộ cục khối) để chỉ giữ lại các đặc trưng phong cách cốt lõi như độ đậm nét, nhịp điệu bút pháp hay xu hướng hình học. Nếu quá trình này thất bại, hiện tượng *rò rỉ nội dung* (content leakage) sẽ xảy ra, khiến chữ Latin sinh ra mang hình dáng méo mó và vô tình tái hiện các yếu tố cấu trúc đặc thù của Hán tự.

### 2.3.2. Khoảng cách Hình thái học (Morphological Gap)

Bên cạnh sự chênh lệch về mật độ thông tin, *khoảng cách hình thái học* giữa hai hệ chữ còn tạo ra một rào cản cản bản đối với việc chuyển giao phong cách trực tiếp. Chữ Latin được thiết kế dựa trên dòng chảy ngang, với độ rộng ký tự biến thiên và cấu trúc mở, trong khi Hán tự tuân theo nguyên tắc **khối vuông cố định**, nơi mọi nét bút đều phải tương tác và cân bằng trong một không gian giới hạn.

Quan trọng hơn, các đơn vị hình thái cơ bản của hai hệ chữ không có sự tương ứng trực tiếp. Những đặc trưng cục bộ trong chữ Latin như chân chữ (serif), điểm kết thúc (terminal) hay độ cong của nét không thể ánh xạ một-một sang các khái niệm như nét bút hay bộ thủ trong Hán tự, vốn mang tính cấu trúc và ngữ nghĩa cao. Như đã được nhấn mạnh trong thiết kế chữ Hán, hình thái của mỗi nét không chỉ mang phong cách mà còn gắn chặt với vai trò của nó trong tổng thể khối chữ.

Chính vì khoảng cách hình thái sâu sắc này, việc áp dụng trực tiếp các mô-đun học phong cách truyền thống—vốn giả định sự tương đồng hình học giữa nguồn và đích—thường tỏ ra kém hiệu quả trong bối cảnh đa hệ chữ. Điều này tạo động lực cho khoa luận đề xuất **Cross-Lingual Style Contrastive Refinement (CL-SCR)**, một cơ chế học phong cách dựa trên việc tách rời và khai thác các đặc trưng *bất biến theo hình thái* (morphology-invariant features), cho phép chuyển giao phong cách một cách linh hoạt và ổn định giữa hai miền chữ viết có bản chất cấu trúc đối lập.



**Hình 2.7** — So sánh cấu trúc hình thái giữa chữ Latin và Hán tự. Chữ Latin được tổ chức theo bố cục tuyến tính dựa trên baseline, với các tham số hình học như x-height và cap height, đồng thời có độ rộng ký tự biến thiên. Ngược lại, Hán tự tuân theo cấu trúc khối vuông cố định (body frame), trong đó các nét bút được phân bố để cân bằng và lắp đầy không gian nội khói. Sự khác biệt căn bản về hệ quy chiếu hình học này tạo nên khoảng cách hình thái học giữa hai hệ chữ. (Trích từ [26])

# Chương 3

## Phương pháp đề xuất

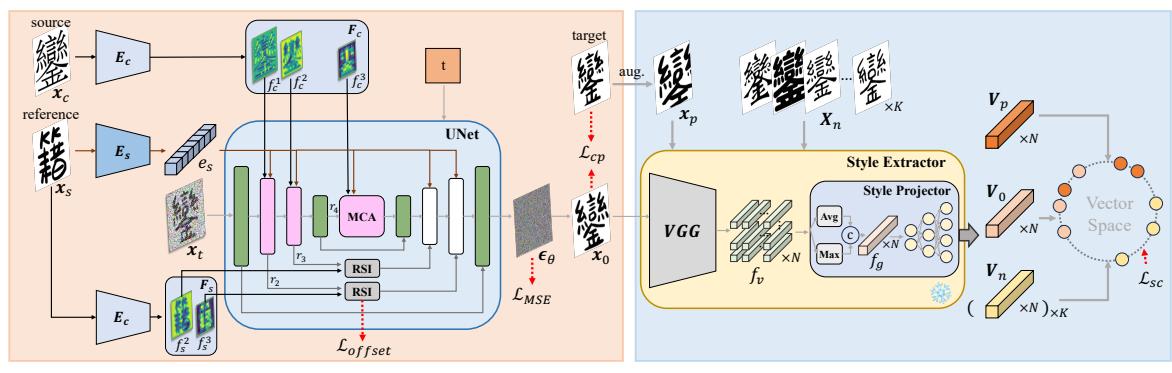
Trong chương trước, khoá luận đã phân tích các hạn chế của phương pháp GAN[1] và tiềm năng của Mô hình khuếch tán (Diffusion Models)[2] trong bài toán sinh phông chữ. Dựa trên cơ sở đó, chương này trình bày chi tiết phương pháp nghiên cứu được đề xuất.

Cụ thể, khoá luận kế thừa kiến trúc tiên tiến **FontDiffuser[4]** làm mô hình cơ sở (baseline) và đề xuất một cải tiến quan trọng tại giai đoạn tinh chỉnh phong cách (Phase 2) mang tên **Cross-Lingual Style Contrastive Refinement (CL-SCR)**. Mục tiêu của cải tiến này là giải quyết vấn đề về sự không nhất quán phong cách khi chuyển đổi giữa các hệ ngôn ngữ có cấu trúc khác biệt (như từ chữ Latin sang Hán tự).

Cấu trúc chương bao gồm: trình bày kiến trúc tổng thể của FontDiffuser[4], phân tích cơ chế hoạt động của mô-đun SCR gốc, và cuối cùng là chi tiết về giải pháp CL-SCR được đề xuất cho bài toán đa ngôn ngữ.

### 3.1. Kiến trúc nền tảng FontDiffuser

FontDiffuser được thiết kế dưới dạng một mô hình khuếch tán có điều kiện (Conditional Diffusion Model - CDM), mô hình hoá bài toán sinh phông chữ dưới dạng quy trình “khử nhiễu” (noise-to-denoise).

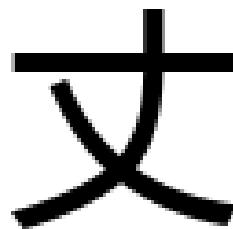


(b) Style Contrastive Refinement

**Hình 3.1** — Mô hình tổng thể của FontDiffuser gồm 2 giai đoạn:  
Tái tạo cấu trúc (Trái) và Tinh chỉnh phong cách (Phải). (Trích từ [4])

Mô hình nhận hai đầu vào chính:

*Ảnh nội dung (Source Image)*  $x_c$ : Cung cấp thông tin về cấu trúc nét, bố cục của ký tự gốc (ví dụ: một chữ cái Arial cơ bản).



**Hình 3.2** — Ví dụ về ảnh nội dung.

*Ảnh phong cách (Reference Image)*  $x_s$ : Cung cấp thông tin về kiểu dáng, độ đậm nhạt, serif, và các đặc trưng thẩm mỹ (ví dụ: một chữ cái thư pháp).



**Hình 3.3** — Ví dụ về ảnh phong cách.

Đầu ra của mô hình là ảnh  $x_0$  – một ký tự mới mang nội dung của  $x_c$  nhưng khoác lên mình phong cách của  $x_s$ .



**Hình 3.4** — Ví dụ về ảnh đầu ra.

Quy trình huấn luyện được chia thành hai giai đoạn (phases) tuần tự nhằm đảm bảo chất lượng sinh ảnh tối ưu:

### 3.1.1. Giai đoạn 1 - Tái tạo cấu trúc (Reconstruction Phase)

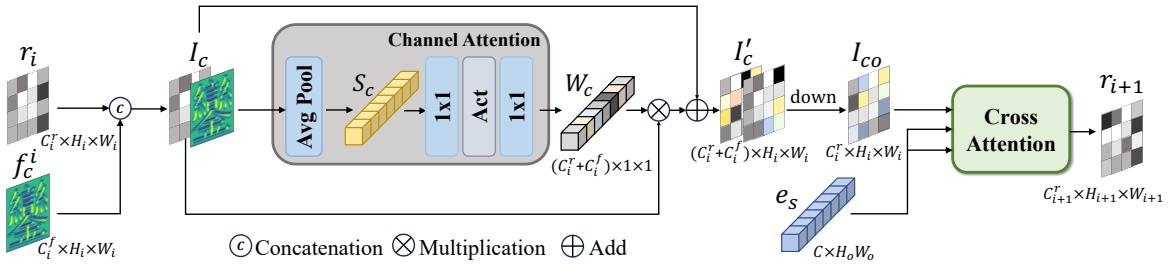
Mục tiêu của giai đoạn này là huấn luyện mô hình khuếch tán học cách khôi phục lại hình ảnh ký tự mục tiêu từ nhiều, dựa trên điều kiện  $x_c$  và  $x_s$ . Các thành phần cốt lõi bao gồm **Bộ mã hoá nội dung ( $E_c$ ) và phong cách ( $E_s$ )** - dùng để trích xuất đặc trưng ngữ nghĩa.

**Multi-scale Content Aggregation (MCA):** Đây là cơ chế tổng hợp đặc trưng đa tỉ lệ được thiết kế để giải quyết hạn chế của các phương pháp chỉ dựa vào một mức đặc trưng duy nhất. Khi sinh các ký tự phức tạp, một tầng đặc trưng đơn lẻ thường không thể đồng thời nắm bắt được cả bộ cục tổng thể lẫn những chi tiết tinh vi như nét mảnh, bộ phận nhỏ hoặc các dấu thanh. MCA khắc phục điều này bằng cách trích xuất nhiều mức đặc trưng nội dung từ các tầng khác nhau của bộ mã hoá, sau đó đưa chúng vào các khối UNet tương ứng.

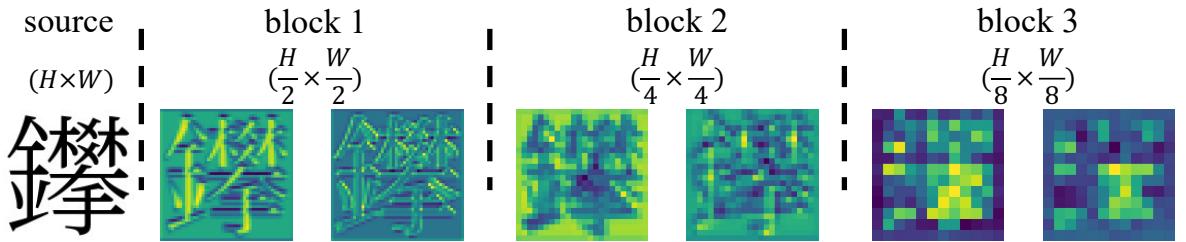
Cụ thể, quy trình hoạt động như sau:

1. Ảnh tham chiếu  $x_c$  trước hết được nhúng bởi bộ mã hoá nội dung  $E_c$  để thu được các đặc trưng đa tỷ lệ  $F_c = \{f_c^1, f_c^2, f_c^3\}$  từ các tầng khác nhau.
2. Mỗi đặc trưng nội dung  $f_c^i$  được đưa vào UNet thông qua ba khối MCA tương ứng. Tại đây,  $f_c^i$  được ghép nối (concatenated) với đặc trưng của khối UNet trước đó là  $r_i$ , tạo ra đặc trưng giàu thông tin  $I_c$ .
3. Để tăng cường khả năng chọn lọc kênh thích ứng, áp dụng cơ chế chú ý kênh (channel attention) lên  $I_c$ . Cơ chế này sử dụng một lớp gộp trung bình (average pooling), hai lớp tích chập  $1 \times 1$  và một hàm kích hoạt để tạo ra vector nhận biết kênh toàn cục  $W_c$ .
4. Vector  $W_c$  sau đó được dùng để trọng số hoá  $I_c$  thông qua phép nhân theo kênh (channel-wise multiplication).
5. Sau khi đi qua một kết nối phần dư (residual connection), một lớp tích chập  $1 \times 1$  được sử dụng để giảm số lượng kênh, thu được đầu ra  $I_{co}$ .
6. Cuối cùng, một mô-đun cross-attention được áp dụng để chèn style embedding  $e_s$ , trong đó  $e_s$  đóng vai trò là Key và Value, còn  $I_{co}$  đóng vai trò là Query.

Nhờ MCA, mô hình có thể tái hiện chính xác cả những thành phần nhỏ và các nét đặc trưng tinh tế—một yếu tố đặc biệt quan trọng đối với những hệ chữ có độ phức tạp cao, bao gồm các ký tự chứa nhiều bộ thủ hoặc các dấu thanh đòi hỏi độ chính xác cao.



**Hình 3.5 — Khối MCA (Multi-scale Content Aggregation).** (Trích từ [4])



**Hình 3.6 — ĐẶC TRUNG Content Ở CÁC KHỐI KHÁC NHAU.** (Trích từ [4])

**Reference-Structure Interaction (RSI):** Giữa ảnh nguồn và ảnh đích thường tồn tại những khác biệt đáng kể về mặt cấu trúc (ví dụ: kích thước phông chữ) cũng như sự lệch lạc về vị trí không gian (spatial misalignment) giữa đặc trưng của UNet và đặc trưng tham chiếu. Để giải quyết vấn đề này, nhóm tác giả đã đề xuất khối Tương tác Cấu trúc - Tham chiếu (RSI). Khối này sử dụng mạng tích chập biến hình (Deformable Convolutional Networks - DCN) để thực hiện biến đổi cấu trúc ngay trên kết nối tắt (skip connection) của UNet.

Điểm khác biệt so với các phương pháp trước đây là thay vì sử dụng CNN truyền thống để tính toán độ lệch (offset)  $\delta_{\text{offset}}$  — vốn hạn chế trong việc nắm bắt thông tin toàn cục — nhóm tác giả đã tích hợp cơ chế Cross-Attention để kích hoạt các tương tác tầm xa (long-distance interactions).

Quy trình cụ thể diễn ra như sau:

1. Ảnh tham chiếu  $x_s$  trước hết được nhúng bởi bộ mã hoá nội dung  $E_c$  để thu các bản đồ cấu trúc (structure maps)  $F_s = \{f_s^1, f_s^2\}$ .
2. Tại mỗi tầng, RSI tiếp nhận các đặc trưng từ UNet ( $r_i$ ) và bản đồ cấu trúc tương ứng ( $f_s^i$ ). Cả hai được làm phẳng (flatten) thành chuỗi vector  $S_r$  và  $S_s$ .

3. Cơ chế Cross-Attention được áp dụng để tính toán vùng quan tâm (region of interest) thông qua phép chiếu tuyến tính  $\phi$ :

**Query (Q):** Được tạo ra từ đặc trưng tham chiếu  $S_s(\phi_q(S_s))$ .

**Key (K) và Value (V):** Được tạo ra từ đặc trưng UNet  $S_r(\phi_k(S_r), \phi_v(S_r))$ .

4. Đặc trưng chú ý  $F_{\text{attn}}$  được tính toán thông qua hàm Softmax, sau đó được đưa qua mạng truyền thẳng (Feed-Forward Network - FFN) để sinh ra độ lệch cấu trúc  $\delta_{\text{offset}}$ .
5. Cuối cùng, DCN sử dụng độ lệch này để “uốn nắn” đặc trưng UNet, tạo ra đầu ra  $I_R$  đã được chỉnh.

$$I_R = \text{DCN}(r_i, \delta_{\text{offset}}) \quad (3.11)$$

Thông qua cơ chế này, RSI có khả năng trích xuất trực tiếp thông tin cấu trúc từ ảnh tham chiếu và điều chỉnh linh hoạt đặc trưng của ảnh nguồn, đảm bảo sự tương thích về phong cách mà không làm gãy vỡ các nét chi tiết.

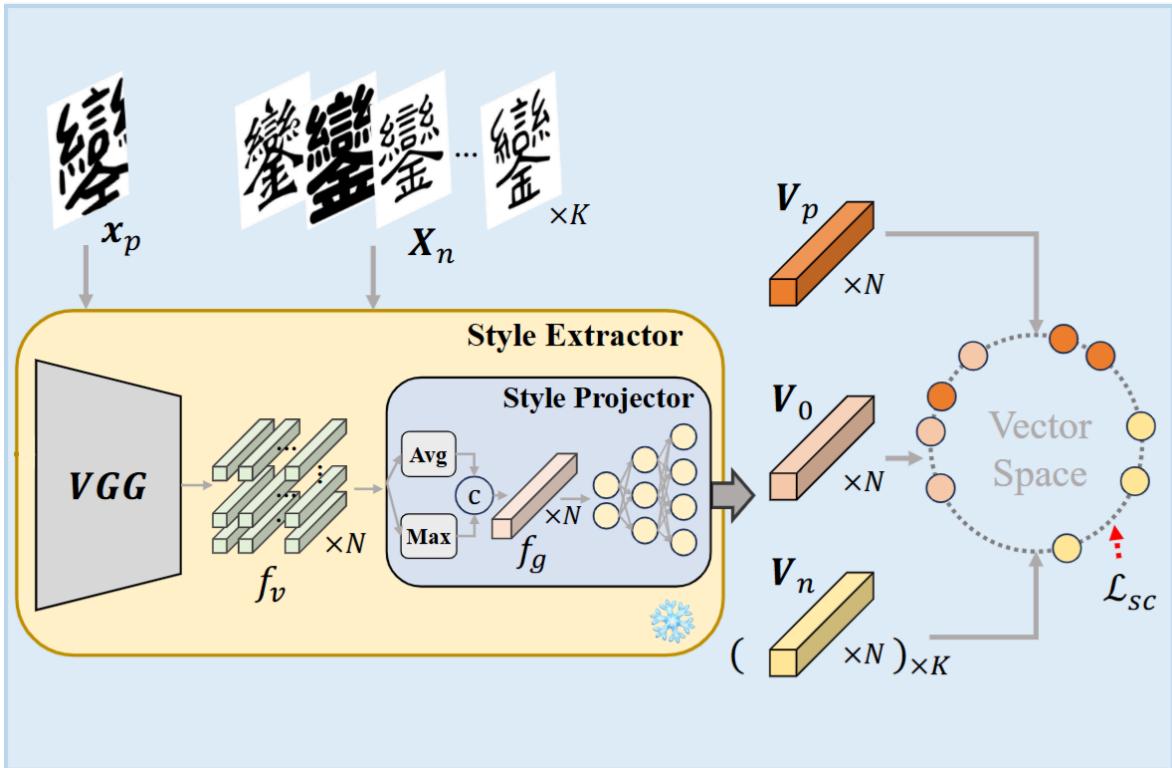
### 3.1.2. Giai đoạn 2 - Tinh chỉnh phong cách (Style Refinement Phase)

Mặc dù Giai đoạn 1 có thể tạo ra ký tự rõ nét, nhưng sự vuông víu (disentanglement) giữa đặc trưng phong cách và nội dung thường chưa hoàn hảo, dẫn đến kết quả phong cách không nhất quán. Giai đoạn 2 cố định các trọng số của UNet và tập trung huấn luyện mô-đun **Style Contrastive Refinement (SCR)**.

SCR hoạt động như một bộ giám sát đặc trưng (feature supervisor), sử dụng cơ chế học tương phản (Contrastive Learning) để cung cấp tín hiệu điều hướng, đảm bảo phong cách của ảnh sinh ra ( $x_0$ ) phải nhất quán với ảnh đích ( $x_p$ ) ở cả cấp độ toàn cục và cục bộ.

#### 3.1.2.1. Kiến trúc Khai thác Phong cách

Kiến trúc của SCR, như được minh họa trong [Hình 3.7](#), bao gồm hai thành phần chính:



**Hình 3.7** — Kiến trúc của mô-đun SCR. (Trích từ [4])

### 1. *Bộ trích xuất Đặc trưng (Style Extractor):*

Sử dụng một mạng **VGG** (lấy cảm hứng từ Zhang et al. 2022[27]) để nhúng ảnh phông chữ, khai thác các đặc tính phong cách và cấu trúc.

Để bao phủ đầy đủ cả phong cách cục bộ (như nét bút, serifs) và toàn cục (như độ đậm, độ nghiêng), bộ trích xuất chọn ra  $N$  tầng feature maps, ký hiệu là  $F_v = \{f_v^0, f_v^1, \dots, f_v^N\}$ .

### 2. *Bộ chiếu Đặc trưng (Style Projector):*

Các feature maps  $F_v$  được đưa vào bộ chiếu. Tại đây, áp dụng đồng thời **average pooling** và **maximum pooling** để trích xuất các đặc trưng kênh toàn cục khác nhau.

Kết quả từ hai phép pooling được nối (concatenate) theo chiều kênh, tạo thành đặc trưng tổng hợp  $F_g$ .

Cuối cùng,  $F_g$  được đưa qua các phép chiếu tuyến tính (linear projections) để thu được các **vector phong cách**  $V = \{v^0, v^1, \dots, v^N\}$ . Các vector này đóng vai trò là đầu vào cho hàm mất mát tương phản.

### 3.1.2.2. Cơ chế Học Tương phản và Hàm Mất mát

SCR sử dụng chiến lược học tương phản (Contrastive Learning), vận dụng hàm mất mát  $L_{sc}$  để điều hướng mô hình khuếch tán.

**Chiến lược Thiết lập Mẫu:** Để đảm bảo tính liên quan về nội dung nhưng phân biệt rõ ràng về phong cách, SCR lựa chọn mẫu cẩn thận:

**Mẫu sinh ra (Generated Sample -  $x_0$ ):** Ảnh được tạo ra bởi mô hình khuếch tán.

**Mẫu dương (Positive Sample -  $x_p$ ):** Là ảnh đích (target image) mang phong cách mong muốn. Để tăng cường **tính bền vững (robustness)** của quá trình bắt chước phong cách, một chiến lược tăng cường dữ liệu (augmentation strategy) được áp dụng trên  $x_p$ , bao gồm **cắt ngẫu nhiên (random cropping)** và **thay đổi kích thước ngẫu nhiên (random resizing)**.

**Mẫu âm (Negative Samples -  $x_n$ ):** Là  $K$  mẫu ảnh có **cùng nội dung** ký tự với  $x_p$  và  $x_0$  nhưng mang **phong cách khác biệt**.

**Định nghĩa hàm mất mát:** Hàm mất mát  $L_{sc}$  (còn được gọi là  $L_{SCR}$  trong công thức tổng thể) là một dạng của hàm **InfoNCE[25]** được tính tổng trên  $N$  tầng đặc trưng:

$$L_{sc} = - \sum_{l=0}^{N-1} \log \frac{\exp(v_0^l \cdot v_p^l / \tau)}{(\exp(v_0^l \cdot v_p^l / \tau) + \sum_{i=1}^K \exp(v_0^l \cdot v_{n_i}^l / \tau))} \quad (3.12)$$

Trong đó:

$N$ : Tổng số tầng đặc trưng được sử dụng để trích xuất và so sánh.

$l$ : Chỉ số đại diện cho tầng đặc trưng đang xét (từ 0 đến  $N - 1$ ).

$v_0^l$ : Vector đặc trưng lớp  $l$  của ảnh sinh (ảnh kết quả cần tối ưu).

$v_p^l$ : Vector đặc trưng lớp  $l$  của ảnh dương/ảnh mẫu (ảnh chứa phong cách mục tiêu).

$v_{n_i}^l$ : Vector đặc trưng lớp  $l$  của ảnh âm thứ  $i$  (các ảnh khác phong cách cần loại bỏ).

$K$ : Số lượng mẫu ảnh âm được sử dụng để so sánh trong công thức.

$v \cdot v'$ : Phép nhân vô hướng, biểu thị độ tương đồng Cô-sin giữa hai vector (đo mức độ giống nhau về phong cách).

$\tau$ : Tham số nhiệt độ (được thiết lập là 0.07), dùng để điều chỉnh độ nhạy của hàm mất mát.

Qua việc tối thiểu hóa hàm mất mát này, mô hình được định hướng để kéo vector phong cách của ảnh sinh lại gần vector của ảnh đích, đồng thời đẩy xa khỏi các vector của các phong cách không mong muốn.

### 3.1.3. Kết hợp vào Mục tiêu Huấn luyện

Để đạt được sự cân bằng giữa việc tái tạo nội dung chính xác và bắt chước phong cách tinh tế, quy trình huấn luyện của FontDiffuser áp dụng chiến lược **hai giai đoạn: từ thô đến tinh (coarse-to-fine two-phase strategy)**.

#### 1. Giai đoạn 1 - Tái tạo Cấu trúc (Phase 1 - Coarse Stage):

Trong giai đoạn đầu, mục tiêu là tối ưu hóa FontDiffuser để mô hình đạt được năng lực nền tảng trong việc tái tạo cấu trúc phông chữ (font reconstruction). Tại bước này, mô-đun SCR **chưa được kích hoạt**. Hàm mất mát tổng thể cho giai đoạn 1 ( $L_{\text{total}}^1$ ) là sự kết hợp của ba thành phần:

$$L_{\text{total}}^1 = L_{\text{MSE}} + \lambda_{\text{cp}}^1 L_{\text{cp}} + \lambda_{\text{off}}^1 L_{\text{offset}} \quad (3.13)$$

Chi tiết các thành phần:

**Hàm mất mát Khuếch tán Tiêu chuẩn ( $L_{\text{MSE}}$ ):** Đây là hàm mất mát cơ bản của mô hình khuếch tán, chịu trách nhiệm tính toán sai số giữa nhiều dự đoán  $\varepsilon_\theta$  và nhiều thực tế  $\varepsilon$  tại bước thời gian  $t$ , với điều kiện đầu vào là ảnh nội dung  $x_c$  và ảnh phong cách  $x_s$ :

$$L_{\text{MSE}} = \|\varepsilon - \varepsilon_{\theta(x_t, t, x_c, x_s)}\|^2 \quad (3.14)$$

**Hàm mất mát Nhận thức Nội dung ( $L_{\text{cp}}$  - Content Perceptual Loss):** Thành phần này được sử dụng để trừu phạt sự lệch lạc về nội dung (content misalignment) giữa ảnh sinh ra  $x_0$  và ảnh đích  $x_{\text{target}}$ . Khoa luận sử dụng các đặc trưng được mã hoá bởi mạng VGG ( $\text{VGG}_l(\cdot)$ ) trên  $L$  tầng được chọn:

$$L_{\text{cp}} = \sum_{l=1}^L \|\text{VGG}_l(x_0) - \text{VGG}_l(x_{\text{target}})\| \quad (3.15)$$

**Hàm mất mát Độ lệch ( $L_{\text{offset}}$  - Offset Loss):** Được thiết kế riêng cho mô-đun RSI (Reference-Structure Interaction), hàm này ràng buộc độ lớn của các vector dịch chuyển  $\delta_{\text{offset}}$  nhằm ngăn chặn các biến dạng cấu trúc quá mức, trong đó mean là phép tính trung bình:

$$L_{\text{offset}} = \text{mean}(\|\delta_{\text{offset}}\|) \quad (3.16)$$

Các siêu tham số trọng số cho giai đoạn 1 được thiết lập là:  $\lambda_{\text{cp}}^1 = 0.01$  và  $\lambda_{\text{off}}^1 = 0.5$ .

## 2. Giai đoạn 2 - Tinh chỉnh Phong cách (Phase 2 - Fine Stage):

Sau khi mô hình đã nắm bắt được cấu trúc, giai đoạn 2 sẽ kích hoạt mô-đun **SCR (Style Contrastive Refinement)**. Mục đích là tích hợp hàm mất mát tương phản phong cách ( $L_{\text{sc}}$ ) để cung cấp tín hiệu hướng dẫn (guidance), giúp mô hình khuếch tán tinh chỉnh các chi tiết phong cách ở cả cấp độ toàn cục và cục bộ.

Hàm mất mát tổng thể cho giai đoạn 2 ( $L_{\text{total}}^2$ ) được mở rộng như sau:

$$L_{\text{total}}^2 = L_{\text{MSE}} + \lambda_{\text{cp}}^2 L_{\text{cp}} + \lambda_{\text{off}}^2 L_{\text{offset}} + \lambda_{\text{sc}}^2 L_{\text{sc}} \quad (3.17)$$

Trong giai đoạn này, các trọng số được giữ nguyên cho các thành phần trước và bổ sung trọng số cho thành phần mới:

$\lambda_{\text{cp}}^2 = 0.01$  (trọng số nội dung).

$\lambda_{\text{off}}^2 = 0.5$  (trọng số độ lệch RSI).

$\lambda_{\text{sc}}^2 = 0.01$  (trọng số tương phản phong cách).

Việc bổ sung  $L_{\text{sc}}$  (như đã định nghĩa ở Phương trình (3.12) trong phần phân tích SCR ([Chương 3.1.2](#))) đóng vai trò then chốt trong việc đảm bảo ảnh đầu ra không chỉ đúng về cấu trúc (nhờ  $L_{\text{cp}}, L_{\text{offset}}$ ) mà còn đạt độ chân thực cao về phong cách nghệ thuật.

## 3.2. Cải tiến đề xuất: Cross-Lingual Style Contrastive Refinement (CL-SCR)

### 3.2.1. Hạn chế của SCR trong bối cảnh đa ngôn ngữ

Mô-đun SCR tiêu chuẩn (Standard SCR) hoạt động dựa trên giả định rằng ảnh nguồn và ảnh tham chiếu chia sẻ cùng một không gian hình thái (cùng một ngôn ngữ). Tuy nhiên, khi mở rộng sang bài toán **Cross-Lingual Font Generation** (Huấn luyện trên

dữ liệu tiếng Latin đơn giản  $D_{\text{source}}$ , ứng dụng sang chữ cái Hán  $D_{\text{target}}$  phức tạp và ngược lại), SCR bộc lộ điểm yếu về **thiên kiến cấu trúc (structural bias)**.

Cụ thể, bộ trích xuất đặc trưng StyleExtractor (sử dụng các tầng VGG pre-trained) có xu hướng “học vẹt” các đặc điểm cấu trúc dày đặc của Hán tự thay vì trích xuất phong cách trừu tượng. Khi gấp các ký tự Latin với cấu trúc thưa, sự chênh lệch miền (domain gap) khiến vector phong cách  $v_{\text{gen}}$  và  $v_{\text{target}}$  không còn tương đồng trong không gian tiềm ẩn.

### 3.2.2. Thiết kế mô-đun CL-SCR

Để giải quyết vấn đề này, khoá luận đề xuất mô-đun **Cross-Lingual SCR (CL-SCR)**. Dựa trên mã nguồn đã xây dựng, CL-SCR không thay đổi kiến trúc cốt lõi của StyleExtractor hay Projector, mà thay đổi **chiến lược lấy mẫu** và **cơ chế tính hàm mất mát đa luồng**.

#### 3.2.2.1. Chiến lược lấy mẫu mở rộng

Thay vì chỉ sử dụng cặp mẫu dương/âm đơn thuần (Intra-lingual), CL-SCR thiết lập đầu vào cho hàm forward của mô hình bao gồm hai luồng dữ liệu song song:

*Luồng Nội miền (Intra-Lingual Flow):*

*Anchor ( $x_{\text{gen}}$ ):* Ảnh sinh ra từ mô hình Diffusion.

*Intra-Positive ( $x_{\text{pos,intra}}$ ):* Ảnh cùng nội dung ký tự, cùng phong cách (Ground Truth). Giúp mô hình giữ vững cấu trúc cơ bản.

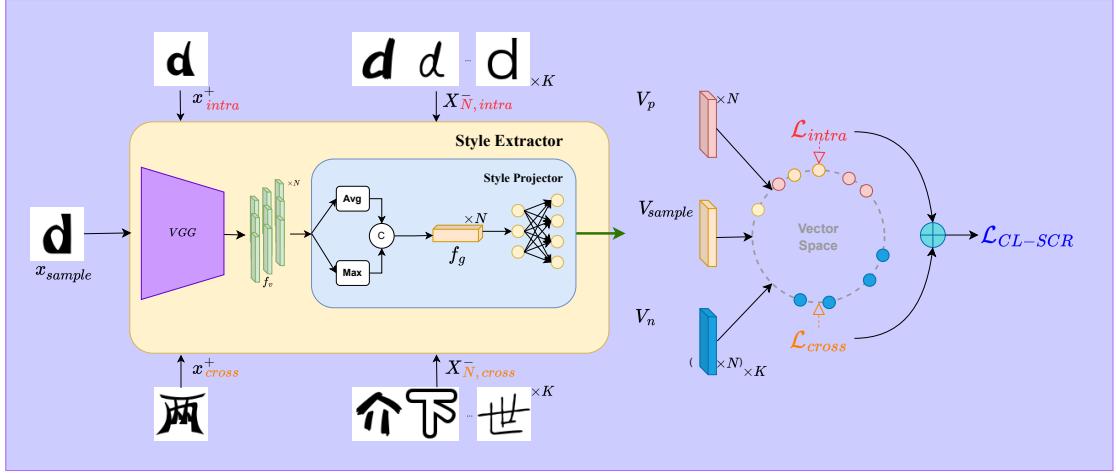
*Intra-Negative ( $x_{\text{neg,intra}}$ ):* Ảnh cùng nội dung, khác phong cách.

*Luồng Xuyên miền (Cross-Lingual Flow - Điểm cải tiến chính):*

*Cross-Positive ( $x_{\text{pos,cross}}$ ):* Các ảnh thuộc ngôn ngữ đích mang cùng Style ID với ảnh tham chiếu. Mục tiêu là ép buộc bộ Projector phải ánh xạ các đặc trưng từ hai ngôn ngữ khác nhau về cùng một cụm vector nếu chúng có cùng phong cách.

*Cross-Negative ( $x_{\text{neg,cross}}$ ):* Các ảnh thuộc ngôn ngữ đích có cấu trúc nét tương đồng nhưng khác phong cách.

Để cung cấp cái nhìn tổng quan về cách các luồng dữ liệu này tương tác trong không gian đặc trưng, kiến trúc chi tiết và cơ chế hoạt động của mô-đun CL-SCR được minh họa trực quan tại [Hình 3.8](#).



**Hình 3.8** — Kiến trúc của mô-đun Tinh chỉnh Tương phản Phong cách Đa ngôn ngữ (CL-SCR), minh họa với ví dụ cụ thể về việc sinh ký tự Latin ‘d’ theo phong cách đích ‘Z’. Với đầu vào là mẫu sinh  $x_{sample}$  đóng vai trò là anchor, mô-đun sử dụng bộ trích xuất phong cách (Style Extractor) chia sẻ trọng số để ánh xạ các đầu vào vào không gian vector tiềm ẩn. Các vector đặc trưng trích xuất được ký hiệu là  $V_{sample}$  (anchor),  $V_p$  (đại diện cho các mẫu dương  $x_{intra}^+$  và  $x_{cross}^+$ ), và  $V_n$  (đại diện cho các tập mẫu âm  $X_{N,intra}^-$  và  $X_{N,cross}^-$ ). Mục tiêu huấn luyện sử dụng cơ chế giám sát kép để đảm bảo phong cách ‘Z’ được nắm bắt chính xác: 1) Nhánh nội miền ( $L_{intra}$ ): Căn chỉnh  $V_{sample}$  với đặc trưng nhãn  $V_p$  (từ  $x_{intra}^+$ ) đồng thời đẩy xa các đặc trưng âm  $V_n$  (từ  $X_{N,intra}^-$ ); và 2) Nhánh xuyên miền ( $L_{cross}$ ): Thu hẹp khoảng cách phong cách bằng cách kéo  $V_{sample}$  lại gần đặc trưng dương xuyên ngôn ngữ  $V_p$  (từ ký tự Hán  $x_{cross}^+$ ). Mục tiêu cuối cùng  $L_{CL-SCR}$  kết hợp các lực này để đảm bảo sự chuyển giao thẩm mỹ nhất quán qua các hệ chữ viết.

### 3.2.2.2. Cơ chế tính toán Loss hỗn hợp

Hàm mất mát CL-SCR được định nghĩa là tổ hợp tuyến tính giữa mất mát nội miền và mất mát xuyên miền:

$$L_{CL-SCR} = \alpha_{intra} \cdot L_{intra} + \beta_{cross} \cdot L_{cross} \quad (3.18)$$

Trong đó, dựa trên thực nghiệm, các siêu tham số trọng số được thiết lập là  $\alpha_{intra} = 0.3$  và  $\beta_{cross} = 0.7$  nhằm ưu tiên khả năng chuyên giao phong cách sang ngôn ngữ đích trong khi vẫn giữ được sự ổn định từ dữ liệu cùng ngôn ngữ.

Cả  $L_{\text{intra}}$  và  $L_{\text{cross}}$  đều được tính toán dựa trên hàm mất mát InfoNCE[25], được **trung bình hóa** qua  $L$  tầng đặc trưng (từ các khối ReLU<sub>1</sub><sup>1</sup> đến ReLU<sub>1</sub><sup>5</sup> của mạng VGG-19). Công thức chi tiết cho thành phần Intra-Lingual Loss ( $L_{\text{intra}}$ ) và Cross-Lingual Loss ( $L_{\text{cross}}$ ) được tính theo trình tự như sau:

$$L_{\text{intra}} = -\frac{1}{L} \sum_{l=1}^L \log \frac{\exp(v_{\text{gen}}^l \cdot v_{\text{pos,intra}}^l / \tau)}{\exp(v_{\text{gen}}^l \cdot v_{\text{pos}_l,\text{intra}}^l / \tau) + \sum_{k=1}^K \exp(v_{\text{gen}}^l \cdot v_{\text{neg}_k, \text{intra}}^l / \tau)} \quad (3.19)$$

$$L_{\text{cross}} = -\frac{1}{L} \sum_{l=1}^L \log \frac{\exp(v_{\text{gen}}^l \cdot v_{\text{pos,cross}}^l / \tau)}{\exp(v_{\text{gen}}^l \cdot v_{\text{pos}_l,\text{cross}}^l / \tau) + \sum_{k=1}^K \exp(v_{\text{gen}}^l \cdot v_{\text{neg}_k, \text{cross}}^l / \tau)} \quad (3.20)$$

Trong đó:

**$L$ :** Tổng số tầng đặc trưng (layers) được sử dụng từ mạng trích xuất (Style Extractor) để tính toán loss.

**$l$ :** Chỉ số chạy đại diện cho tầng đặc trưng thứ  $l$  (từ 1 đến  $L$ ).

**$v_{\text{gen}}^l$ :** Vector đặc trưng phong cách tại lớp  $l$  của ảnh được sinh ra (Generated Image). Đây đóng vai trò là mẫu neo (anchor) trong phép so sánh.

**$v_{\text{pos,intra}}^l$ :** Vector đặc trưng của mẫu dương nội tại (Intra-positive). Đây là ảnh có cùng phong cách và cùng hệ chữ với ảnh sinh (ví dụ: ảnh sinh là chữ ‘A’ font thư pháp, thì mẫu dương là chữ ‘B’ font thư pháp).

**$v_{\text{pos,cross}}^l$ :** Vector đặc trưng của mẫu dương chéo (Cross-positive). Đây là ảnh có cùng phong cách nhưng thuộc hệ chữ khác (ví dụ: ảnh sinh là chữ ‘A’ Latin, mẫu dương là chữ Hán có cùng nét bút thư pháp đó).

**$v_{\text{neg}_k, \text{intra}}^l / v_{\text{neg}_k, \text{cross}}^l$ :** Các vector đặc trưng của mẫu âm (Negative samples) thứ  $k$ . Đây là các ảnh có phong cách khác biệt hoàn toàn, cần được đẩy xa khỏi ảnh sinh trong không gian đặc trưng.

**$K$ :** Tổng số lượng mẫu âm được sử dụng trong mỗi lần tính toán.

**$\tau$ :** Tham số nhiệt độ (Temperature), thường đặt là 0.07. Giúp điều chỉnh độ nhạy của hàm loss với các mẫu khó phân biệt.

**$\cdot$ :** Phép nhân vô hướng (Dot product), đại diện cho độ tương đồng Cô-sin (Cosine Similarity) giữa hai vector. Giá trị càng lớn nghĩa là hai phong cách càng giống nhau.

Với  $v = \text{Projector}(\text{Extractor}(x))$  là vector phong cách sau khi đi qua mạng chiếu.

### 3.2.2.3. Quy trình huấn luyện Pha 2 cải tiến

Trong giai đoạn tinh chỉnh (Phase 2), hàm mất mát tổng thể được cập nhật để tích hợp CL-SCR. Việc sử dụng song song cả intra và cross loss giúp mô hình vừa duy trì tính ổn định (nhờ intra) vừa học được tính bất biến của phong cách qua các ngôn ngữ (nhờ cross).

Hàm mục tiêu cuối cùng là:

$$L_{\text{Total}}^2 = L_{\text{MSE}} + \lambda_{\text{content}} L_{\text{content}} + \lambda_{\text{offset}} L_{\text{offset}} + \lambda_{\text{style}} L_{\text{CL-SCR}} \quad (3.21)$$

Trong đó:

$L_{\text{MSE}}$ : đảm bảo ảnh sinh ra không bị biến dạng quá nhiều so với ảnh gốc.

$L_{\text{content}}$  (**Content Perceptual Loss**): giữ gìn cấu trúc nét chữ.

$L_{\text{offset}}$ : kiểm soát độ dịch chuyên của mô-đun RSI.

$L_{\text{CL-SCR}}$ : đóng vai trò trọng tâm trong việc chuyển giao phong cách đa ngôn ngữ.

Việc tích hợp CL-SCR kỳ vọng sẽ giúp mô hình “bắt” được các đặc trưng phong cách trừu tượng (như độ xước cọ, độ thanh mảnh) tốt hơn và áp dụng chính xác lên các ký tự Hán phức tạp và ngược lại.

### 3.3. Đề xuất thuật toán tính CL-SCR

Dựa trên cơ chế lấy mẫu đa luồng và hàm mất mát InfoNCE[25], thuật toán tính toán giá trị loss cho mô-đun CL-SCR được trình bày chi tiết dưới đây.

#### Thuật toán 3.1 — Thuật toán tính hàm mất mát CL-SCR

Input	$S$	Vector đặc trưng của ảnh sinh (Sample/Anchor)
	$P_{\text{intra}}, N_{\text{intra}}$	Tập mẫu Dương/Âm thuộc luồng Nội miền
	$P_{\text{cross}}, N_{\text{cross}}$	Tập mẫu Dương/Âm thuộc luồng Xuyên miền
	$\alpha, \beta$	Trọng số cho luồng nội miền và xuyên miền
	$L$	Số lượng tầng đặc trưng (sử dụng $\text{ReLU}_1^x$ )
	mode	Chế độ huấn luyện {intra, cross, both}
Output	$L_{\text{total}}$	Giá trị loss cuối cùng
1	<b>procedure</b>	
1	CAL_CL_SCR_LOSS( $S, P_{\text{intra}}, N_{\text{intra}}, P_{\text{cross}}, N_{\text{cross}}, \alpha, \beta, \tau$ ):	
2	$L_{\text{total}} \leftarrow 0.0$	▷ Loss tổng
		▷ Biến đếm số nhánh tham gia tính
3	count $\leftarrow 0$	loss
4	<b>if</b> mode $\in \{\text{intra, both}\}$ <b>and</b>	
4	$P_{\text{intra}} \neq \emptyset$ <b>then</b>	
5	$L_{\text{intra}} \leftarrow 0$	▷ Duyệt qua các tầng $\text{ReLU}_1^1$ đến
6	<b>for</b> $l = 1 \rightarrow L$ <b>do</b>	$\text{ReLU}_1^5$
7	$L_{\text{intra}} \leftarrow L_{\text{intra}} +$	
7	$\text{InfoNCE}(S^l, P_{\text{intra}}^l, N_{\text{intra}}^l, \tau)$	
8	<b>end for</b>	
9	$L_{\text{intra}} \leftarrow L_{\text{intra}} / L$	▷ Trung bình cộng các tầng
10	<b>if</b> mode == both <b>then</b>	
11	$L_{\text{total}} \leftarrow L_{\text{total}} + \alpha \cdot$	
11	$L_{\text{intra}}$	▷ Áp dụng trọng số $\alpha$
12	<b>else</b>	
13	$L_{\text{total}} \leftarrow L_{\text{total}} + L_{\text{intra}}$	

```

14   |   end if
15   |   count  $\leftarrow$  count + 1
16   |   end if
17   |   if mode  $\in \{\text{cross, both}\}$  and
18   |    $P_{\text{cross}} \neq \emptyset$  then
19   |   |    $L_{\text{cross}} \leftarrow 0$ 
20   |   |   for  $l = 1 \rightarrow L$  do
21   |   |   |    $L_{\text{cross}} \leftarrow L_{\text{cross}} +$ 
22   |   |   |   InfoNCE( $S^l, P_{\text{cross}}^l, N_{\text{cross}}^l, \tau$ )
23   |   |   end for
24   |   |    $L_{\text{cross}} \leftarrow L_{\text{cross}} / L$ 
25   |   |   if mode == both then
26   |   |   |    $L_{\text{total}} \leftarrow L_{\text{total}} + \beta \cdot$ 
27   |   |   |    $L_{\text{cross}}$ 
28   |   |   else
29   |   |   |    $L_{\text{total}} \leftarrow L_{\text{total}} + L_{\text{cross}}$ 
30   |   |   end if
31   |   |   count  $\leftarrow$  count + 1
32   |   end if
33   |   return  $L_{\text{total}}$ 
34 end procedure

```

# Chương 4

## Thực nghiệm và Đánh giá kết quả

Chương này trình bày chi tiết **thiết lập thực nghiệm**, bao gồm mô tả bộ dữ liệu, các thước đo đánh giá và cấu hình huấn luyện chi tiết trên nền tảng phần cứng giới hạn. Tiếp theo, khoá luận sẽ đưa ra các **so sánh định lượng và định tính giữa phương pháp đề xuất (CL-SCR FontDiffuser)** với các **phương pháp tiên tiến hiện nay (State-of-the-Art)** nhằm chứng minh hiệu quả trong bài toán sinh phông chữ đa ngôn ngữ (Cross-Lingual Font Generation) theo cả hai chiều: **từ Hán tự sang Latin** và **từ Latin sang Hán tự**.

### 4.1. Bộ dữ liệu (Datasets)

#### 4.1.1. Cấu trúc

Để đảm bảo tính khách quan và khả năng so sánh công bằng với các nghiên cứu tiên tiến, khoá luận không tự xây dựng dữ liệu mới mà kế thừa **bộ dữ liệu chuẩn** từ công trình “Few-shot Font Style Transfer between Different Languages”[16]. Đây là tập dữ liệu chuyên biệt cho bài toán đa ngôn ngữ, bao gồm **818 bộ phông chữ song ngữ** với độ đa dạng phong cách cao, trải dài từ serif, sans-serif đến thư pháp và viết tay. Cấu trúc dữ liệu được tổ chức thành hai tập con tương tác chặt chẽ nhằm phục vụ bài toán chuyển đổi hai chiều: **tập ký tự Hán** chứa trung bình **800 ký tự** thông dụng (chuẩn GB2312) đóng vai trò miền đích phức tạp, và **tập ký tự Latin** bao gồm **52 ký tự cơ bản**. Đặc điểm cốt lõi của bộ dữ liệu này là **sự nhất quán tuyệt đối về phong cách** giữa hai hệ chữ trong cùng một bộ font, **cung cấp các cặp dữ liệu nhãn (Ground-truth)** tự nhiên giúp mô-đun CL-SCR học được sự tương quan phong cách xuyên ngôn ngữ.



**Hình 4.1** — Minh họa hai hệ chữ trong cùng một bộ font.

#### 4.1.2. Tiền xử lý và Chuẩn hoá

**Quy trình tiền xử lý:** Về quy trình tiền xử lý, dữ liệu thô trải qua các bước chuẩn hoá để tối ưu hoá quá trình huấn luyện. Cụ thể, toàn bộ ảnh ký tự được render dưới dạng **thang độ xám (grayscale)** nhằm loại bỏ nhiều màu sắc, giúp mô hình tập trung tối đa vào việc học các đặc trưng hình học và cấu trúc nét. Các ảnh đầu vào sau đó được chuẩn hoá đồng bộ về kích thước  $64 \times 64$  pixel, đồng thời áp dụng kỹ thuật **căn chỉnh tự động (auto-centering)** để đưa ký tự về tâm ảnh với tỷ lệ lề phù hợp.

Cuối cùng, một bước **lọc bỏ thủ công** được thực hiện để loại trừ các mẫu lỗi như ký tự bị đứt nét hoặc render thiếu, đảm bảo chất lượng đầu vào tốt nhất cho mô hình.

**Quy trình Chuẩn hóa và Lấy mẫu Động:** Tiếp nối các bước xử lý thô, để đảm bảo tính tương thích tối đa với kiến trúc mạng nơ-ron tích chập và cơ chế khuếch tán, khoá luận thiết lập một **đường ống xử lý dữ liệu** chuyên biệt được triển khai thời gian thực trong quá trình huấn luyện. Cụ thể, thông qua lớp `FontDataset`, mọi ảnh đầu vào (bao gồm ảnh nội dung, ảnh phong cách và các mẫu âm) đều được chuyển đổi đồng bộ sang không gian màu **RGB (3 kênh)** để khớp với đầu vào tiêu chuẩn của bộ mã hoá U-Net. Ké đến, kỹ thuật **nội suy song tuyến tính (Bilinear Interpolation)** được áp dụng để đưa ảnh về độ phân giải mục tiêu, giúp làm mượt các đường biên răng cưa và bảo toàn thông tin cấu trúc tốt hơn so với các phương pháp lấy mẫu lân cận. Về mặt số học, dữ liệu trải qua bước **chuẩn hoá giá trị (Value Normalization)**, chuyển đổi các điểm ảnh từ dải  $[0, 255]$  sang dạng Tensor với dải giá trị tiêu chuẩn  $[-1, 1]$ , tạo điều kiện hội tụ ổn định cho quá trình khử nhiễu Gaussian. Đặc biệt, để phục vụ mô-đun CL-SCR, khoá luận áp dụng chiến lược **Lấy mẫu âm động (Dynamic Negative Sampling)**: thay vì cố định các cặp mẫu, hệ thống tự động truy xuất và lựa chọn ngẫu nhiên  $K$  mẫu âm từ kho dữ liệu dựa trên chế độ huấn luyện (nội miền `intra` hoặc xuyên miền `cross`) ngay tại mỗi bước lặp, giúp mô hình liên tục được tiếp xúc với các biến thể phong cách đa dạng và tránh hiện tượng học vẹt.

## 4.2. Thiết lập Thực nghiệm

### 4.2.1. Cấu hình Huấn luyện (Implementation Details)

Các thí nghiệm được thực hiện trên môi trường tính toán đám mây Kaggle với **GPU NVIDIA Tesla P100 (16GB VRAM)**. Mã nguồn được triển khai trên nền tảng **PyTorch** và thư viện **Diffusers**.

Quá trình huấn luyện tuân theo chiến lược **hai giai đoạn (Two-stage training)** với các siêu tham số được thiết lập cụ thể như sau dựa trên tài nguyên phần cứng giới hạn:

#### 1. **Giai đoạn Tái tạo (Phase 1 - Reconstruction):**

Trong giai đoạn khởi đầu này, mục tiêu chính của mô hình là học các đặc trưng cấu trúc nội dung và phong cách cơ bản. Quá trình huấn luyện được thực hiện xuyên suốt **400,000 bước lặp** với kích thước batch được cố định là **4**. Về chiến lược tối ưu hoá,

khoá luận sử dụng bộ giải thuật **AdamW** với tốc độ học khởi tạo là  $1 \times 10^{-4}$ , kết hợp cùng lịch trình điều chỉnh Linear bao gồm **10,000 bước khởi động** (warmup steps) để đảm bảo mô hình hội tụ ổn định. Hàm mất mát tổng hợp được cấu hình với các trọng số thành phần cụ thể là  $\lambda_{\text{percep}} = 0.01$  cho Content Perceptual Loss và  $\lambda_{\text{offset}} = 0.5$  cho Offset Loss nhằm hỗ trợ mô-đun RSI học biến dạng cấu trúc.

## **2. Tiền huấn luyện mô-đun CL-SCR:**

Trước khi được tích hợp vào luồng sinh ảnh chính, mô-đun CL-SCR (Cross-Lingual Style Contrastive Refinement) trải qua một quá trình huấn luyện độc lập nhằm xây dựng không gian biểu diễn phong cách tối ưu. Quá trình này được thực hiện trong tổng số **200,000 bước lặp** với kích thước batch là **16**. Khoá luận sử dụng bộ tối ưu hoá Adam để cập nhật tham số cho cả bộ trích xuất đặc trưng (Style Feat Extractor) và bộ chiếu đặc trưng (Style Projector) với tốc độ học cố định là  $1 \times 10^{-4}$ .

Để tăng cường tính bền vững của biểu diễn phong cách đối với các biến thể hình học, khoá luận áp dụng chiến lược tăng cường dữ liệu (Data Augmentation) thông qua kỹ thuật **Random Resized Crop**. Cụ thể, ảnh đầu vào được **cắt ngẫu nhiên với tỷ lệ diện tích từ 80% đến 100% (scale 0.8 - 1.0)** và **tỷ lệ khung hình dao động nhẹ trong khoảng 0.8 đến 1.2**, sau đó được đưa về kích thước chuẩn thông qua nội suy song tuyến tính (bilinear interpolation).

## **3. Giai đoạn Tinh chỉnh Phong cách bằng mô-đun CL-SCR (Phase 2 - Style Refinement with CL-SCR):**

Bước sang giai đoạn hai, mô-đun CL-SCR được kích hoạt để tinh chỉnh sâu các đặc trưng phong cách Latin, trong khi tốc độ học của các thành phần khác được giảm xuống để tránh phá vỡ cấu trúc đã học. Quá trình này diễn ra trong **30,000 bước** với **kích thước batch 4** nhằm dành tài nguyên VRAM cho các tính toán của mô-đun tương phản. Tốc độ học được thiết lập ở mức thấp hơn là  $1 \times 10^{-5}$ , áp dụng chiến lược Constant (hằng số) sau **1,000 bước khởi động**. Đối với cấu hình CL-SCR, khoá luận lựa chọn chế độ huấn luyện kết hợp cả nội miền và xuyên miền (`scr_mode="both"`) với tỷ trọng  $\alpha_{\text{intra}} = 0.3$  và ưu tiên  $\beta_{\text{cross}} = 0.7$ , đồng thời sử dụng **4 mẫu âm** (negative samples) cho mỗi lần tính toán loss. Hàm mục tiêu tổng thể lúc này là sự kết hợp của các thành phần theo công thức:

$$L_{\text{total}} = L_{\text{MSE}} + 0.01 \cdot L_{\text{percep}} + 0.5 \cdot L_{\text{offset}} + 0.01 \cdot L_{\text{CL-SCR}} \quad (4.22)$$

## **4. Quy trình Inference:**

Trong quá trình lấy mẫu (Inference), mô hình FontDiffuser[4] được đóng gói thành một Pipeline dựa trên DPM-Solver để tối ưu hóa tốc độ.

**Cấu hình Lấy mẫu:** Khoá luận sử dụng bộ giải **DPM-Solver++** với số bước suy diễn được cố định là **20** (`num_inference_steps=20`), đây là một sự cân bằng giữa tốc độ tính toán và chất lượng ảnh sinh. Chiến lược hướng dẫn vô điều kiện (Classifier-Free Guidance[28]) được áp dụng với tham số hướng dẫn ( $s$ ) được xác định trong file cấu hình (`guidance_scale`). Để lấy mẫu, các ảnh đầu vào được tiền xử lý và chuẩn hoá về kích thước (`content_image_size`, `style_image_size`) rồi đưa về Tensor với dải giá trị  $[-1, 1]$ .

**Lấy mẫu Hàng loạt (Batch Sampling):** Do khoá luận thực hiện đánh giá định lượng trên một lượng lớn mẫu, quy trình lấy mẫu được tự động hoá thông qua hàm `batch_sampling`, bao phủ cả hai hướng nghiên cứu.

#### 4.2.2. Kịch bản Đánh giá (Evaluation Scenarios)

Để đánh giá toàn diện khả năng của mô hình, khoá luận thiết lập **hai kịch bản kiểm thử** với **độ khó tăng dần** (theo chuẩn của FontDiffuser[4] và DG-Font[5]):

**SFUC (Seen Font, Unseen Character):** Font đã xuất hiện trong tập huấn luyện, nhưng ký tự sinh ra chưa từng thấy. Kịch bản này đánh giá khả năng **nội suy phong cách**.

**UFSC (Unseen Font, Seen Character):** Font mới hoàn toàn (chưa từng xuất hiện trong quá trình huấn luyện). Đây là kịch bản quan trọng nhất để đánh giá khả năng **One-shot Generalization** của mô hình đối với phong cách lạ.

### 4.3. Các thước đo đánh giá (Evaluation Metrics)

Để đảm bảo **tính khách quan** và **toàn diện** trong việc kiểm định chất lượng mô hình, khoá luận áp dụng hệ thống đánh giá đa chiều bao gồm cả các **chỉ số định lượng tiêu chuẩn (Quantitative Metrics)** và **đánh giá định tính dựa trên cảm nhận người dùng (Subjective User Study)**.

#### 4.3.1. Chỉ số Định lượng (Quantitative Metrics)

Khoá luận sử dụng bộ 4 chỉ số tiêu chuẩn trong bài toán sinh ảnh để đánh giá chất lượng ảnh sinh ( $x$ ) so với ảnh thật ( $y$ ):

#### 4.3.1.1. L1 (Mean Absolute Error)

Độ đo **L1** tính trung bình giá trị tuyệt đối của sai khác giữa các điểm ảnh (pixel-wise), phản ánh độ chính xác về cường độ điểm ảnh:

$$L1 = \frac{1}{N} \sum_{i=1}^N |x_i - y_i| \quad (4.23)$$

Trong đó:

$N$ : Tổng số lượng điểm ảnh (pixels) trong hình ảnh.

$x_i$ : Giá trị cường độ điểm ảnh tại vị trí  $i$  của ảnh sinh ra.

$y_i$ : Giá trị cường độ điểm ảnh tại vị trí  $i$  của ảnh mầu (Ground Truth).

$|\cdot|$ : Phép tính giá trị tuyệt đối.

**Ý nghĩa:** Giá trị L1 càng nhỏ thể hiện **sai số tái tạo càng thấp**, tức ảnh sinh càng sát với ảnh gốc về mặt tín hiệu. Tuy nhiên, L1 thường **không phản ánh tốt cảm nhận thị giác của mắt người** (ví dụ: ảnh mờ vẫn có thể có L1 thấp).

#### 4.3.1.2. SSIM (Structural Similarity Index)

Độ đo **SSIM**[18] đánh giá mức độ tương đồng về **cấu trúc, độ sáng và độ tương phản**. Khác với L1, SSIM mô phỏng cách mắt người cảm nhận sự thay đổi cấu trúc cục bộ:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (4.24)$$

Trong đó:

$\mu_x, \mu_y$ : Giá trị trung bình cục bộ của ảnh  $x$  và ảnh  $y$  (đại diện cho **độ sáng**).

$\sigma_x^2, \sigma_y^2$ : Phương sai cục bộ của ảnh  $x$  và ảnh  $y$  (đại diện cho **độ tương phản**).

$\sigma_{xy}$ : Hiệp phương sai giữa  $x$  và  $y$  (đại diện cho **sự tương đồng cấu trúc**).

$C_1, C_2$ : Các hằng số nhỏ để đảm bảo tính ổn định khi mẫu số tiến tới 0 (thường được tính theo  $C_1 = (k_1 L)^2$ ,  $C_2 = (k_2 L)^2$  với  $L$  là dải giá trị động của pixel).

**Ý nghĩa:** Giá trị SSIM nằm trong khoảng  $[0, 1]$ , **giá trị càng cao** thể hiện **chất lượng ảnh càng tốt**.

#### 4.3.1.3. LPIPS (Learned Perceptual Image Patch Similarity)

Độ đo **LPIPS[19]** đánh giá **khoảng cách cảm nhận** dựa trên các đặc trưng trích xuất từ mạng nơ-ron sâu (thường là VGG[29] hoặc AlexNet[30]). Chỉ số này khắc phục nhược điểm của L1/SSIM khi xử lý các ảnh bị mờ nhẹ nhưng vẫn giống về ngữ nghĩa:

$$\text{LPIPS}(x, y) = \sum_l \frac{1}{H_l W_l} \sum_{h,w} \|w_l \cdot (f_l^x(h, w) - f_l^y(h, w))\|_2^2 \quad (4.25)$$

Trong đó:

$f_l^x, f_l^y$ : Bản đồ đặc trưng (feature map) tại lớp thứ  $l$  của mạng nơ-ron trích xuất từ ảnh  $x$  và  $y$ .

$H_l, W_l$ : Chiều cao và chiều rộng của bản đồ đặc trưng tại lớp  $l$ .

$w_l$ : Vector trọng số dùng để chuẩn hoá kênh (channel scaling factors).

$\cdot$ : Phép nhân từng phần tử (element-wise product).

$\|\cdot\|_2^2$ : Bình phương khoảng cách Euclid.

**Ý nghĩa:** LPIPS khắc phục nhược điểm của L1/SSIM khi xử lý các ảnh bị mờ nhẹ nhưng vẫn đúng về ngữ nghĩa. **Giá trị LPIPS càng thấp** chứng tỏ ảnh sinh **càng giống ảnh thật về mặt thị giác tự nhiên theo cảm nhận của mắt người**.

#### 4.3.1.4. FID (Fréchet Inception Distance)

Độ đo **FID[17]** đánh giá chất lượng tổng thể và độ đa dạng của tập ảnh sinh dựa trên khoảng cách thống kê giữa hai phân bố đặc trưng (thường được trích xuất từ lớp **Pool3** của mạng InceptionV3):

$$\text{FID}(r, g) = \|\mu_r - \mu_g\|_2^2 + \text{Tr} \left( \sum_r + \sum_g - 2 \left( \sum_r \sum_g \right)^{\frac{1}{2}} \right) \quad (4.26)$$

Trong đó:

$\mu_r, \mu_g$ : Vector trung bình đặc trưng (mean feature vector) của tập ảnh thật ( $r$ ) và tập ảnh sinh ( $g$ ).

$\sum_r, \sum_g$ : Ma trận hiệp phương sai (covariance matrix) của tập ảnh thật và tập ảnh sinh.

$\|\cdot\|_2^2$ : Bình phương khoảng cách Euclid giữa hai vector trung bình.

Tr: Phép tính vết của ma trận (Trace - tổng các phần tử trên đường chéo chính).

**Ý nghĩa:** FID đo khoảng cách Fréchet giữa hai phân bố Gaussian đa biến. **Giá trị FID càng thấp** cho thấy **phân bố của ảnh sinh càng tiệm cận với phân bố ảnh thật**, đồng nghĩa với việc mô hình **sinh ra ảnh vừa chân thực (realism) vừa đa dạng (diversity)**.

#### 4.3.1.5. Phân tích mối tương quan và Vai trò của bộ độ đo

Việc sử dụng đơn lẻ một độ đo không thể phản ánh toàn diện hiệu năng của mô hình sinh phông chữ, do đó khoá luận kết hợp bốn độ đo trên theo **chiến lược đánh giá đa tầng**. Đầu tiên, ở tầng **đánh giá độ chính xác điểm ảnh (Pixel-level Accuracy)**, các chỉ số **L1** và **SSIM** đảm bảo rằng ảnh sinh ra không bị lệch lạc quá nhiều về vị trí không gian so với ảnh mẫu (Ground Truth). Tuy nhiên, đối với các mô hình sinh (Generative Models), việc tối ưu hóa quá mức L1 thường dẫn đến hiện tượng ảnh bị **làm mờ (blurring effect)** để giảm thiểu sai số trung bình. Để khắc phục, tầng thứ hai tập trung vào **đánh giá chất lượng cảm nhận (Perceptual Quality)** thông qua **LPIPS** và **FID**. Trong khi LPIPS đo lường sự tương đồng trong **không gian đặc trưng (Feature Space)** giúp mô hình được “tha thứ” cho những sai lệch nhỏ về pixel miễn là đặc điểm nhận dạng bảo toàn, thì FID đóng vai trò trọng tâm trong việc đánh giá mức độ “**thật**” (**realism**) và **tính đa dạng (diversity)**.

Sự kết hợp giữa SSIM (cấu trúc) và LPIPS (cảm nhận) là đặc biệt quan trọng trong bài toán Cross-Lingual, nơi việc giữ cấu trúc chữ quan trọng ngang hàng với việc bắt chước phong cách.

#### 4.3.2. Đánh giá Định tính (Qualitative Evaluation)

Các chỉ số định lượng (Quantitative Metrics) như FID hay LPIPS, mặc dù khách quan, nhưng không thể mô phỏng hoàn toàn gu thẩm mỹ và khả năng đọc hiểu của con người. Do đó, để kiểm chứng tính thực tiễn của phương pháp đề xuất, Khoa luận thực hiện **đánh giá định tính** trên hai khía cạnh: **phân tích thị giác dựa trên chuyên môn (Visual Analysis)** và **khảo sát cảm nhận người dùng (User Study)**.

##### 4.3.2.1. Quy trình Phân tích Trực quan (Visual Analysis Protocol)

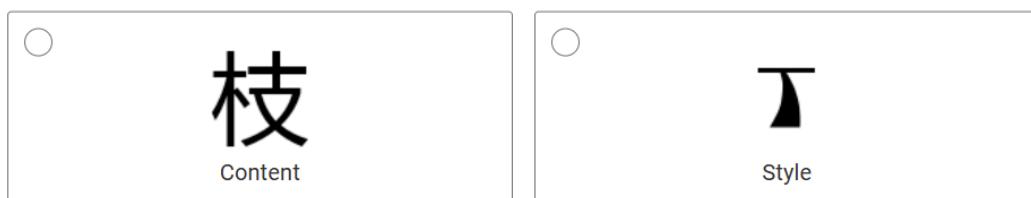
Để kiểm chứng chất lượng thực tế, khoá luận thực hiện **so sánh song song** giữa ảnh sinh ra từ mô hình đề xuất và các mô hình khác nhằm soi xét các **lỗi thị giác cụ thể**

bằng mắt thường, tập trung vào việc quan sát xem các nét chữ — đặc biệt là những **nét mảnh** hoặc **các vùng giao nhau phức tạp** — có giữ được **độ liền mạch và dứt khoát** hay bị **dứt gãy**, đồng thời kiểm tra xem ảnh có gặp phải các lỗi “khó coi” như bị **mờ nhoè, lem luốc** hoặc xuất hiện các **vết mực thừa** khiến cấu trúc chữ bị biến dạng hay không.

#### 4.3.2.2. Thiết kế Khảo sát Người dùng (User Study Design)

Để đánh giá chất lượng thị giác và tính nhất quán phong cách một cách khách quan nhất theo cảm nhận của con người, khoá luận thiết kế một bảng **khảo sát mù (blind test)** với sự tham gia của tổng cộng **20 tình nguyện viên**. Nhóm khảo sát bao gồm 5 người bạn học chuyên ngành thiết kế đồ họa có kiến thức về typography và 15 người dùng phổ thông, đảm bảo tính đại diện cho cả đánh giá kỹ thuật và thẩm mỹ công chúng.

Bộ dữ liệu khảo sát được xây dựng từ **20 bộ mẫu ngẫu nhiên** trích xuất từ tập kiểm thử (Test Set), **bao gồm các mẫu đại diện cho cả hai kịch bản chuyển đổi phong cách: từ Hán tự sang Latin và từ Latin sang Hán tự**. Trong mỗi câu hỏi, tình nguyện viên được yêu cầu so sánh kết quả sinh ảnh giữa các mô hình khác nhau. Cụ thể, trong mỗi câu hỏi, tình nguyện viên được cung cấp hai dữ liệu đầu vào gồm: một **ảnh nội dung (Content Image)** để xác định cấu trúc ký tự và một **ảnh tham chiếu (Reference Style)** để xác định phong cách mục tiêu.



**Hình 4.2** — Ví dụ về ảnh nội dung và ảnh tham chiếu.

Dựa trên hai dữ liệu này, người tham gia được yêu cầu quan sát các **ảnh kết quả** được sinh ra bởi 5 mô hình khác nhau (bao gồm DG-Font, CF-Font, DFS, FTransGAN và Phương pháp đề xuất Ours). Vị trí hiển thị của các ảnh kết quả này được **xáo trộn ngẫu nhiên** nhằm loại bỏ thiên kiến vị trí. Nhiệm vụ của tình nguyện viên là chọn ra bức ảnh duy nhất mà họ đánh giá là tối ưu nhất dựa trên hai tiêu chí: **độ nhất quán**

**phong cách** so với ảnh tham chiếu và **chất lượng hình ảnh tổng thể** (độ sắc nét và tính toàn vẹn cấu trúc).



**Hình 4.3** — Ví dụ về các kết quả mà người tham gia khảo sát có thể chọn.

**Tiêu chí đánh giá:** Thay vì chấm điểm phức tạp, người tham gia được yêu cầu thực hiện đánh giá dựa trên **lựa chọn ưu tiên**. Cụ thể, với mỗi bộ mẫu, tình nguyện viên cần chọn ra một bức ảnh duy nhất mà họ cho là tốt nhất dựa trên sự cân bằng giữa hai tiêu chí cốt lõi. Thứ nhất là **Tính nhất quán phong cách**, tức ảnh sinh ra phải kế thừa chính xác các đặc trưng thị giác của ảnh phong cách (như độ đậm nhạt, kết cấu nét cọ, hoặc kiểu chân chữ serif/sans-serif). Thứ hai là **Tính toàn vẹn nội dung**, tức ký tự sinh ra phải duy trì đúng cấu trúc hình học của ảnh nội dung, đảm bảo tính dễ đọc và không bị biến dạng kỳ quái (ví dụ: chữ 丘 trong kịch bản e2c phải giữ nguyên các nét ngang dọc đặc trưng, không bị lai tạp thành ký tự Latin). Kết quả cuối cùng được định lượng thông qua **Tỷ lệ Ưu tiên**, tính bằng phần trăm số phiếu bầu chọn mà mỗi mô hình nhận được trên tổng số lượt đánh giá.

#### 4.4. Kết quả Thực nghiệm và Thảo luận

Trong chương này, khoá luận trình bày toàn bộ kết quả thực nghiệm của mô hình đề xuất. Nội dung bao gồm đánh giá định lượng và định tính chi tiết, nghiên cứu bóc tách (ablation study) về các thành phần kiến trúc, khảo sát người dùng, và phân tích các trường hợp thất bại. Các kết quả này được đối chiếu trực tiếp với nhiều mô hình sinh

font hiện đại, bao gồm các mô hình **GAN-based** (DG-Font[5], CF-Font[6], DFS[15], FTransGAN[16]), mô hình **diffusion-based** (FontDiffuser[4]), và các phiên bản mô hình của khoá luận.

Để đánh giá toàn diện khả năng chuyển đổi đa ngôn ngữ, khoá luận thực hiện thực nghiệm trên hai hướng chính với các mục tiêu nghiên cứu và cấu hình mô hình cụ thể, khẳng định giá trị nghiên cứu ngang nhau của bài toán Cross-Lingual Font Generation:

### 1. Hướng Latin → Hán tự:

Đây là kịch bản kiểm tra khả năng **chuyển giao phong cách Latin** tinh tế lên **cấu trúc Hán tự** phức tạp. Trong kịch bản này, mô hình cần học các đặc trưng nét (như serif, độ dày nét, góc bo) của hệ chữ Latin và áp dụng chúng lên các ký tự Hán. Mục tiêu là kiểm tra hiệu quả của mô-đun **CL-SCR** trong việc tách biệt phong cách Latin khỏi nội dung Latin, đảm bảo sự nhất quán phong cách khi áp dụng lên hệ chữ có hình thái học khác biệt (Hán tự).

Khoa luận sử dụng hai cấu hình mô hình cho hướng này: Ours<sub>A</sub> (sử dụng ký tự A làm ảnh phong cách tham chiếu) và Ours<sub>AZ</sub> (sử dụng ký tự ngẫu nhiên trong khoảng A đến Z làm ảnh phong cách tham chiếu).

### 2. Hướng Hán tự → Latin:

Đây là kịch bản kiểm tra khả năng **khái quát hóa phong cách Hán tự** phức tạp lên **cấu trúc Latin** đơn giản. Trong kịch bản này, mô hình phải học các đặc trưng phong cách đa dạng (ví dụ: nét bút lông, độ dày-mỏng bất đối xứng) từ Hán tự và áp dụng chúng lên cấu trúc Latin. Sự thành công trong hướng này chứng tỏ mô hình có thể trích xuất các đặc trưng phong cách bậc cao của Hán tự để áp dụng hợp lý lên các ký tự Latin có cấu trúc tuyển tính hơn.

Đối với hướng Hán tự → Latin, khoá luận tiến hành phân loại và đánh giá các kịch bản dựa trên **độ phức tạp của ký tự Hán tự** (số nét  $M$ ) được sử dụng làm ảnh tham chiếu phong cách, nhằm phân tích độ nhạy của mô hình đối với sự đa dạng của nét:

**Bảng 4.1** — Bảng phân loại các kịch bản dựa trên độ phức tạp của ký tự.

Cấp độ	Định nghĩa (Số nét $M$ )	Cấu hình Mô hình	Mục tiêu Phân tích
All	Đánh giá tổng thể trên các ký tự Hán tự có số nét ngẫu nhiên.	Ours <sub>All</sub>	Đánh giá hiệu năng trung bình của mô hình

Cấp độ	Định nghĩa (Số nét $M$ )	Cấu hình Mô hình	Mục tiêu Phân tích
			trên toàn bộ miền dữ liệu Hán tự.
Easy	Ảnh phong cách là Hán tự có số nét $6 \leq M \leq 10$ .	Ours <sub>Easy</sub>	Kiểm tra khả năng học các đặc trưng phong cách từ cấu trúc đơn giản.
Medium	Ảnh phong cách là Hán tự có số nét $11 \leq M \leq 20$ .	Ours <sub>Medium</sub>	Kiểm tra hiệu quả của các mô-đun bảo toàn nét (MCA) khi đối mặt với cấu trúc trung bình.
Hard	Ảnh phong cách là Hán tự có số nét $M \geq 21$ .	Ours <sub>Hard</sub>	Đánh giá khả năng trích xuất phong cách từ cấu trúc phức tạp và dày đặc nhất mà không làm mất thông tin nét.

有更串 噇雕臚 龜巖爨  
 花识芷 磔壇愒 櫺鷀叢  
 蛛郇跂 褙𠀧漱 麻龍龐

**Hình 4.4** — Ví dụ ba loại độ phức tạp. (Trích từ [4])

Việc phân loại theo độ phức tạp này giúp khoá luận xác định mô-đun **CL-SCR** hoặc các kiến trúc lõi khác (**MCA**, **RSI**) hoạt động hiệu quả nhất ở mức độ phức tạp cấu trúc nào của phong cách Hán tự, từ đó cung cấp những cái nhìn sâu sắc hơn về khả năng học đặc trưng của mô hình khuếch tán.

#### 4.4.1. So sánh Định lượng

Các bảng dưới đây trình bày kết quả so sánh giữa phương pháp đề xuất (Ours) với các baseline mạnh nhất hiện nay gồm DG-Font[5], CF-Font[6], DFS[15], FTransGAN[16] và trên 2 kịch bản UFSC và SFUC cho tác vụ chuyển đổi phong cách từ chữ Latin sang ảnh nguồn Hán và ngược lại.

#### 4.4.1.1. Tác vụ chuyển đổi phong cách từ chữ Latin sang ảnh nguồn Hán (e2c)

**Bảng 4.2** — Kết quả Định lượng cho Latin → Hán tự (e2c) trên SFUC.  
Mũi tên chỉ hướng tốt hơn (thấp hơn hoặc cao hơn).

Phương pháp	L1 ↓	SSIM ↑	LPIPS ↓	FID ↓
DG-Font[5]	0.2773	0.2702	0.4023	106.3833
CF-Font[6]	0.2659	0.2740	0.3979	91.2134
DFS[15]	0.2131	0.3558	0.3812	45.4212
FTransGAN[16]	<b>0.1844</b>	<u>0.3900</u>	0.3548	40.4561
FontDiffuser (Baseline)[4]	0.1976	0.3775	0.2968	14.6871
Ours <sub>A</sub> (w/ CL-SCR)	<u>0.1927</u>	<b>0.3912</b>	<b>0.2868</b>	<u>12.3964</u>
Ours <sub>AZ</sub> (w/ CL-SCR)	0.1939	0.3890	<u>0.2911</u>	<b>11.7691</b>

**Bảng 4.3** — Kết quả Định lượng cho Latin → Hán tự (e2c) trên UFSC.  
Mũi tên chỉ hướng tốt hơn (thấp hơn hoặc cao hơn).

Phương pháp	L1 ↓	SSIM ↑	LPIPS ↓	FID ↓
DG-Font[5]	0.2797	0.2654	0.3649	54.0974
CF-Font[6]	0.2638	0.2716	0.3615	51.3925
DFS[15]	<b>0.2008</b>	0.3048	0.3876	62.7206
FTransGAN[16]	<u>0.2089</u>	0.3109	0.3329	42.1053
FontDiffuser (Baseline)[4]	0.2283	0.2946	0.3184	29.0999
Ours <sub>A</sub> (w/ CL-SCR)	0.2218	<u>0.3144</u>	<b>0.2892</b>	<u>17.8373</u>
Ours <sub>AZ</sub> (w/ CL-SCR)	0.2214	<b>0.3197</b>	<u>0.2954</u>	<b>13.5508</b>

Dựa trên số liệu từ [Bảng 4.2](#) và [Bảng 4.3](#), có thể rút ra những đánh giá quan trọng về hiệu năng của phương pháp đề xuất so với các mô hình State-of-the-Art (SOTA).

**Thứ nhất, về chất lượng thị giác và độ tự nhiên của ảnh sinh:** Kết quả thực nghiệm cho thấy sự cải thiện mang tính đột phá được phản ánh qua chỉ số FID. Trong kịch bản SFUC (Seen Font), biến thể tốt nhất Ours<sub>AZ</sub> đạt FID là **11.769**, giảm khoảng **20%** so với baseline mạnh nhất là FontDiffuser[4] (14.687) và bỏ xa các phương pháp GAN truyền thống. Sức mạnh thực sự của mô hình được thể hiện rõ nét hơn ở kịch bản khó UFSC (Unseen Font), nơi mô hình phải sinh ảnh từ các font chưa từng thấy. Tại đây, Ours<sub>AZ</sub> đạt FID **13.551**, thấp hơn tới **53%** so với FontDiffuser (29.100). Điều này chứng minh mô-đun CL-SCR đã giải quyết hiệu quả vấn đề “domain gap” (khoảng cách miền dữ liệu) giữa chữ Hán và chữ Latin, giúp ảnh sinh ra có phân bố sát với ảnh thật thay vì bị nhiễu hoặc méo mó.

**Thứ hai, về khả năng bảo toàn cấu trúc và nghịch lý L1:** Phương pháp đề xuất dẫn đầu về chỉ số tương đồng cấu trúc SSIM ở cả hai kịch bản (đạt **0.391** ở SFUC và **0.320** ở UFSC), cho thấy các nét chữ được tái tạo sắc nét và đúng cấu trúc. Một điểm đáng lưu ý là mô hình FTransGAN[16] đạt kết quả tốt nhất về sai số điểm ảnh L1 (0.1844 ở SFUC), nhưng chỉ số FID của nó lại rất cao (40.456). Đây là minh chứng điển hình cho “nghịch lý L1”: các mô hình hồi quy (như FTransGAN hay DFS) thường tối ưu hoá bằng cách sinh ra các ảnh “trung bình cộng” bị mờ để giảm thiểu sai số pixel tuyệt đối. Ngược lại, phương pháp đề xuất chấp nhận chỉ số L1 cao hơn một chút để tái tạo các chi tiết tần số cao, mang lại độ sắc nét và tính chân thực vượt trội cho thị giác con người.

**Thứ ba, hiệu quả của chiến lược tham chiếu ngẫu nhiên (A vs. AZ):** Sự so sánh nội bộ giữa hai biến thể (Ours<sub>A</sub> và Ours<sub>AZ</sub>) khẳng định tầm quan trọng của việc đa dạng hoá dữ liệu tham chiếu đầu vào. Ours<sub>AZ</sub> đạt hiệu suất vượt trội hơn hẳn so với Ours<sub>A</sub>, đặc biệt là sự chênh lệch lớn về FID ở kịch bản UFSC (**13.55** so với **17.84**). Điều này chứng minh rằng việc sử dụng linh hoạt các ký tự ngẫu nhiên (A-Z) làm ảnh mẫu giúp quá trình trích xuất đặc trưng tách biệt được phong cách khỏi nội dung hiệu quả hơn, thay vì bị chi phối (bias) bởi cấu trúc hình học cố định của ký tự A. Nhờ đó, mô hình nắm bắt được bản chất của phong cách (như độ đậm nhạt, serif, texture) để áp dụng nhất quán cho các font chữ lạ, tránh tình trạng sao chép máy móc các đặc điểm cục bộ của một ký tự tham chiếu duy nhất.

	Ảnh nội dung	泡玉瓜瓦申
	Ảnh phong cách	B I N U V
SFUC	DG-Font	泡玉瓜瓦申
	CF-Font	泡玉瓜瓦申
	DFS	泡玉瓜瓦申
	FTransGAN	泡玉瓜瓦申
	FontDiffuser (Baseline)	泡玉瓜瓦申
	Ours <sub>AZ</sub>	泡玉瓜瓦申
	Target	泡玉瓜瓦申

**Hình 4.5** — So sánh ảnh sinh trên tập SFUC cho kịch bản Latin → Hán tự (e2c) giữa các phương pháp và ground truth.

	Ảnh nội dung	毛毫民气水
	Ảnh phong cách	Z D W B J
DG-Font		𠀤 〇 𣎵 气水
CF-Font		毛毫民气水
UFSC	DFS	毛毫民气水
	FTransGAN	毛毫民气水
	FontDiffuser (Baseline)	毛毫民气水
Ours <sub>AZ</sub>		毛毫民气水
Target		毛毫民气水

**Hình 4.6** — So sánh ảnh sinh trên tập UFSC cho kịch bản Latin → Hán tự (e2c) giữa các phương pháp và ground truth.

#### 4.4.1.2. Tác vụ chuyển đổi phong cách từ chữ Hán sang ảnh nguồn Latin (c2e)

**Bảng 4.4** — Kết quả Định lượng cho Hán tự → Latin (c2e) trên SFUC.  
Mũi tên chỉ hướng tốt hơn (thấp hơn hoặc cao hơn).

Phương pháp	L1 ↓	SSIM ↑	LPIPS ↓	FID ↓
DG-Font[5]	0.1462	0.5542	0.2821	74.1655
CF-Font[6]	0.1402	0.5621	0.2790	67.1241
DFS[15]	0.1083	0.6140	0.2585	40.4042
FTransGAN[16]	0.1381	0.5291	0.2851	55.5859
FontDiffuser (Baseline)[4]	0.1223	0.6107	0.2270	21.2234
Ours <sub>All</sub> (w/ CL-SCR)	0.1083	0.6406	0.2019	14.7298
Ours <sub>Easy</sub> (w/ CL-SCR)	<b>0.1079</b>	<b>0.6413</b>	<b>0.2018</b>	<b>14.6558</b>
Ours <sub>Medium</sub> (w/ CL-SCR)	<u>0.1082</u>	<u>0.6406</u>	<u>0.2024</u>	<u>14.8556</u>
Ours <sub>Hard</sub> (w/ CL-SCR)	0.1114	0.6318	0.2084	15.7662

**Bảng 4.5** — Kết quả Định lượng cho Hán tự → Latin (c2e) trên UFSC.  
Mũi tên chỉ hướng tốt hơn (thấp hơn hoặc cao hơn).

Phương pháp	L1 ↓	SSIM ↑	LPIPS ↓	FID ↓
DG-Font[5]	0.1397	0.5624	0.2751	89.8197
CF-Font[6]	0.1317	0.5756	0.2726	84.3787
DFS[15]	0.1139	0.5819	0.2907	75.2760
FTransGAN[16]	0.1456	0.4949	0.3023	88.4450
FontDiffuser (Baseline)[4]	0.1370	0.5731	0.2476	59.5788
Ours <sub>All</sub> (w/ CL-SCR)	0.1090	0.6377	0.1985	<b>41.1152</b>
Ours <sub>Easy</sub> (w/ CL-SCR)	<u>0.1050</u>	<u>0.6439</u>	<u>0.1945</u>	<u>41.7273</u>

Phương pháp	L1 ↓	SSIM ↑	LPIPS ↓	FID ↓
Ours <sub>Medium</sub> (w/ CL-SCR)	<b>0.1029</b>	<b>0.6466</b>	<b>0.1929</b>	43.6918
Ours <sub>Hard</sub> (w/ CL-SCR)	0.1050	0.6444	0.1982	45.5486

Dựa trên số liệu từ [Bảng 4.4](#) và [Bảng 4.5](#), kết quả thực nghiệm cho thấy **phương pháp đề xuất (Ours) đạt được sự cải thiện toàn diện so với các mô hình SOTA**, đồng thời hé lộ mối tương quan thú vị giữa độ phức tạp của Hán tự nguồn và hiệu quả chuyển đổi phong cách lên chữ Latin.

**Thứ nhất, về hiệu năng tổng thể và khả năng tổng quát hoá:** Mô hình đề xuất vượt trội hoàn toàn so với Baseline FontDiffuser[4] ở cả hai kịch bản. Trên tập dữ liệu quen thuộc SFUC, cấu hình Ours<sub>Easy</sub> đạt mức FID thấp kỷ lục **14.656**, giảm khoảng 31% so với Baseline (21.223). Sự chênh lệch càng trở nên rõ rệt hơn ở kịch bản khó UFSC (Unseen Font), nơi Ours<sub>All</sub> đạt FID **41.115**, thấp hơn đáng kể so với mức **59.579** của Baseline. Khi so sánh với các phương pháp GAN (như DG-Font, CF-Font, FTransGAN), vốn có chỉ số FID rất cao (trên 80 tại UFSC), phương pháp đề xuất chứng minh ưu thế tuyệt đối về độ tự nhiên và tính thẩm mỹ. Điều này khẳng định mô-đun CL-SCR không chỉ hiệu quả trong việc tinh chỉnh phong cách nội tại mà còn giúp mô hình tổng quát hoá tốt hơn khi phải áp dụng các phong cách Hán tự lạ lẫm, phức tạp lên cấu trúc Latin đơn giản.

**Thứ hai, phân tích “Điểm ngọt” về độ phức tạp nét:** Việc phân tách ảnh tham chiếu (reference images) thành các nhóm Easy, Medium và Hard mang lại những góc nhìn giá trị về hiệu quả của việc chuyển đổi phong cách. Tại [Bảng 4.5](#), cấu hình Ours<sub>Medium</sub> đạt kết quả tốt nhất về các chỉ số cấu trúc và điểm ảnh (**L1 thấp nhất 0.1029, SSIM cao nhất 0.6466**). Điều này gợi ý rằng các Hán tự có số nét trung bình (11-20 nét) là “**điểm ngọt**” để làm ảnh mẫu trích xuất phong cách: chúng cung cấp đủ thông tin về bút pháp và kết cấu (tốt hơn Easy) nhưng không gây ra quá nhiều nhiễu cấu trúc cho quá trình suy luận như các ký tự Hard (trên 21 nét). Vì chữ Latin có cấu trúc hình học rất đơn giản, việc sử dụng các ký tự nguồn quá phức tạp (Hard) khiến mô hình gặp khó khăn trong việc lọc bỏ các chi tiết thừa khi mapping sang đích, dẫn đến hiệu suất tái tạo cấu trúc (SSIM) thấp hơn.

**Thứ ba, sự đánh đổi giữa độ chính xác và độ tự nhiên:** Một điểm đáng lưu ý là mặc dù việc sử dụng ảnh tham chiếu nhóm Medium (Ours<sub>Medium</sub>) giúp tối ưu hoá các

chỉ số kỹ thuật (L1/SSIM), nhưng cấu hình sử dụng toàn bộ không gian tham chiếu (Ours<sub>All</sub>) lại đạt chỉ số **FID tốt nhất** trên tập UFSC (**41.115**). Điều này cho thấy việc đa dạng hoá độ phức tạp của ảnh đầu vào (input reference) giúp mô hình tiếp cận được không gian biểu diễn phong cách phong phú và liên tục hơn. Nhờ đó, ảnh sinh ra có độ tự nhiên cao nhất về mặt cảm nhận thị giác (visual perception), ngay cả khi độ khớp chính xác từng điểm ảnh thua kém nhẹ so với việc chỉ sử dụng nhóm ảnh mẫu Medium.

SFUC	Ảnh nội dung	k   m n o
	Ảnh phong cách	李 线 她 坦 与
DG-Font		𠂇 角 m 部 安
CF-Font		k l m n o
DFS		k   m n o
FTransGAN		k   m n o
FontDiffuser (Baseline)		k   m n o
Ours <sub>All</sub>		k   m n o

Target

k l m n o

**Hình 4.7** — So sánh ảnh sinh trên tập SFUC cho kịch bản Hán tự → Latin (c2e) giữa các phương pháp và ground truth.

Ảnh nội dung

c d e f g

Ảnh phong cách

衣 牛 土 生 至

DG-Font

𠂇 𢃁 𠂇 𠂇 𠂇

CF-Font

c d e f g

DFS

𠂇 牛 土 生 至

FTransGAN

ⓘ d e ⓘ g

FontDiffuser (Baseline)

ⓘ d e ⓘ g

Ours<sub>All</sub>

ⓘ d e ⓘ g



**Hình 4.8** — So sánh ảnh sinh trên tập UFSC cho kịch bản Hán tự → Latin (c2e) giữa các phương pháp và ground truth.

#### 4.4.2. So sánh Định tính

Bên cạnh các chỉ số đo lường, việc phân tích trực quan là bước không thể thiếu để kiểm chứng khả năng xử lý các trường hợp khó của mô hình, đặc biệt là các lỗi cấu trúc mà các chỉ số thông kê như FID đôi khi không phản ánh hết. Khoá luận thực hiện phân tích dựa trên hình ảnh sinh ra từ hai chiều chuyển đổi đối lập.

##### 4.4.2.1. Phân tích Trực quan

Để kiểm chứng các chỉ số định lượng, phân tích trực quan tại các hình từ [Hình 4.5](#) đến [Hình 4.8](#) minh chứng sự ưu việt của phương pháp đề xuất (Ours) so với các phương pháp tiên tiến và **mô hình cơ sở (FontDiffuser baseline)** về *độ sắc nét* và *khả năng bảo toàn nội dung* xuyên ngôn ngữ.

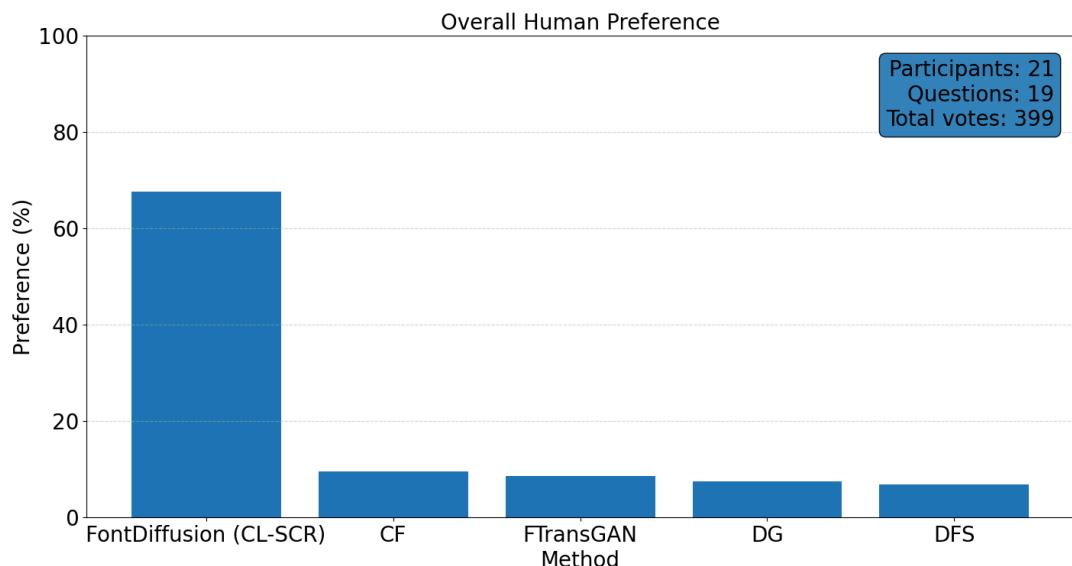
Cụ thể, đối với tác vụ **Latin sang Hán tự**, trong khi *DFS* sinh ra các nét mảnh thiêu sức sống tại [Hình 4.5](#) và *FTransGAN* gặp hiện tượng “bóng ma” mờ nhòe do hạn chế của hàm mất mát L1 tại [Hình 4.6](#), thì **FontDiffuser (baseline)** tuy đã cải thiện được độ rõ nét nhưng vẫn gặp khó khăn trong việc điều chỉnh *trọng lượng nét bút (stroke weight)*, dẫn đến các ký tự đôi khi bị quá đậm hoặc quá nhạt so với ảnh mẫu. Ngược lại, mô hình Ours<sub>AZ</sub> lại tái tạo chính xác *độ đậm* và *cấu trúc* tinh tế của nét cọ, đảm bảo sự đồng nhất cao với phong cách mục tiêu.

Đối với chiều ngược lại từ **Hán tự sang Latin**, phương pháp đề xuất khắc phục hoàn toàn lỗi *rò rỉ nội dung (content leak)* nghiêm trọng của *DG-Font* tại [Hình 4.7](#) (nơi các chữ cái Latin bị biến dạng thành giả Hán tự). Đặc biệt, khả năng *học kết cấu tinh vi* được thể hiện rõ rệt tại [Hình 4.8](#): trong khi **FontDiffuser (baseline)** có xu hướng *làm trơn hoà (oversmoothing)* các chi tiết gai góc hoặc sinh ra các nét liền mạch sai lệch, thì Ours<sub>All</sub> là mô hình duy nhất tái hiện thành công hiệu ứng “in kim” (*dot-matrix*) tách biệt, thay vì sinh ra các nét viền rỗng như *FTransGAN* hay hình ảnh vỡ

nát như DFS. Điều này khăng định **giá trị thực tiễn và khả năng tổng quát hoá** vượt trội của mô-đun CL-SCR so với kiến trúc nền tảng ban đầu.

#### 4.4.2.2. Dánh giá Cảm nhận Người dùng

Dựa trên quy trình khảo sát mù (blind test) đã được thiết lập chi tiết tại [Chương 4.3.2.2](#), khoá luận tổng hợp kết quả bình chọn từ 20 tình nguyện viên trên tập dữ liệu kiểm thử ngẫu nhiên.



**Hình 4.9** — Biểu đồ so sánh tỷ lệ ưu tiên của người dùng giữa phương pháp đề xuất (Ours) và các phương pháp SOTA khác. Kết quả cho thấy sự vượt trội về độ hài lòng thị giác của mô hình tích hợp CL-SCR.

#### Phân tích và Thảo luận:

Kết quả định lượng cho thấy sự vượt trội của phương pháp đề xuất (Ours) với tỷ lệ được ưu tiên lựa chọn trung bình đạt **khoảng 70%**, bỏ xa các phương pháp đối chứng (cao nhất là CF-Font chỉ đạt khoảng 10%). Sự chênh lệch áp đảo này phản ánh sự tương đồng giữa cảm nhận chủ quan của mắt người và các chỉ số máy học (FID/LPIPS) đã phân tích trước đó.

Xu hướng lựa chọn của người dùng có thể được lý giải thông qua sự so sánh trực quan, trong đó **tính dễ đọc (Legibility)** đóng vai trò là yếu tố tiên quyết. Thực tế cho thấy, người dùng thường có phản xạ loại bỏ ngay lập tức các mẫu bị **biến dạng cấu trúc nặng nề** - một nhược điểm có thể khiến các mô hình thuộc họ GAN (như DG-

Font, CF-Font) nhận được tỷ lệ bình chọn rất thấp (< 10%). Trong bối cảnh đó, mô hình đề xuất đã chứng minh được ưu thế nhờ khả năng bảo toàn khung xương ký tự vững chắc thông qua cơ chế MCA và RSI, giúp các kết quả sinh ra vượt qua được rào cản nhận thức đầu tiên về mặt cấu trúc để tiến tới các đánh giá chi tiết hơn về phong cách.

Tóm lại, tỷ lệ ưu tiên cao trong khảo sát người dùng là minh chứng thực tiễn khẳng định phương pháp đề xuất đã đạt được điểm cân bằng tốt nhất giữa hai yếu cầu cốt lõi: giữ đúng chữ (Content) và thể hiện đúng kiểu (Style).

## 4.5. Nghiên cứu Bóc tách (Ablation Study)

Trong phần này, khoá luận thực hiện các phân tích chuyên sâu nhằm định lượng đóng góp cụ thể của từng thành phần kỹ thuật trong phương pháp đề xuất. Để đảm bảo tính tập trung và sức thuyết phục của các kết luận, thay vì dàn trải thí nghiệm trên mọi biến thể, khoá luận cố định và lựa chọn hai cấu hình đại diện tiêu biểu nhất làm cơ sở so sánh:

**Đối với hướng Latin → Hán tự ( e2c )**: Khoá luận sử dụng cấu hình Ours<sub>AZ</sub>. Đây là cấu hình chịu áp lực tổng quát hoá lớn nhất (do phải xử lý style ngẫu nhiên) và cũng là cấu hình đạt hiệu năng cao nhất trong các thực nghiệm trước đó. Việc chứng minh hiệu quả trên cấu hình “khó” nhất này sẽ khẳng định tính đúng đắn và mạnh mẽ (robustness) của các cải tiến đề xuất.

**Đối với hướng Hán tự → Latin ( c2e )**: Khoá luận sử dụng cấu hình Ours<sub>All</sub>. Do đặc thù độ phức tạp nét đa dạng của Hán tự, cấu hình này bao quát toàn bộ phổ dữ liệu huấn luyện, cung cấp cái nhìn toàn diện về độ ổn định của mô hình thay vì chỉ tập trung vào một tập con cụ thể (như Easy hay Hard).

Các thí nghiệm dưới đây sẽ lần lượt đánh giá tác động của sáu yếu tố then chốt: các mô-đun kiến trúc, kỹ thuật tăng cường dữ liệu, chế độ hàm măt măt, số lượng mẫu âm, alpha và beta, và trọng số hướng dẫn.

### 4.5.1. Ảnh hưởng của các mô-đun trong FontDiffuser

Để xác định đóng góp cụ thể của từng thành phần trong kiến trúc tổng thể, đặc biệt là hiệu quả của mô-đun đề xuất so với bản gốc, khoá luận tiến hành **thực nghiệm bóc**

tách (**Ablation Study**) bằng cách **thay thế và bỏ** sung dần các mô-đun vào mạng nền tảng. Bốn mô-đun được khảo sát bao gồm:

**M: Multi-scale Content Aggregation (MCA)** - Tổng hợp nội dung đa quy mô.

**R: Reference-Structure Interaction (RSI)** - Tương tác cấu trúc tham chiếu.

**S: Style Contrastive Refinement (SCR)** - Tinh chỉnh tương phản phong cách đơn ngôn ngữ (Của FontDiffuser gốc).

**CL: Cross-Lingual Style Contrastive Refinement (CL-SCR)** - Tinh chỉnh tương phản phong cách đa ngôn ngữ (Đè xuất cải tiến).

Kết quả thực nghiệm trên hai hướng chuyển đổi được trình bày chi tiết tại [Bảng 4.6](#) và [Bảng 4.7](#).

**Bảng 4.6** — Ảnh hưởng của các thành phần M, R, S và CL đến hiệu năng mô hình trên tác vụ Latin → Hán tự (e2c).

	Mô-đun				L1 ↓	SSIM ↑	LPIPS ↓	FID ↓
	M	R	S	CL				
SFUC	x	x	x	x	0.2441	0.2983	0.4434	70.3650
	✓	✓	✓	x	<u>0.1976</u>	<u>0.3775</u>	<u>0.2968</u>	<u>14.6871</u>
	✓	✓	x	✓	<b>0.1939</b>	<b>0.3890</b>	<b>0.2911</b>	<b>11.7691</b>
UFSC	x	x	x	x	0.2815	0.1965	0.4854	75.7399
	✓	✓	✓	x	<u>0.2283</u>	<u>0.2946</u>	<u>0.3184</u>	<u>29.0999</u>
	✓	✓	x	✓	<b>0.2214</b>	<b>0.3197</b>	<b>0.2954</b>	<b>13.5508</b>

**Bảng 4.7** — Ảnh hưởng của các thành phần M, R, S và CL đến hiệu năng mô hình trên tác vụ Hán tự → Latin (c2e).

	Mô-đun				L1 ↓	SSIM ↑	LPIPS ↓	FID ↓
	M	R	S	CL				
SFUC	x	x	x	x	0.2763	0.2491	0.4792	84.7434
	✓	✓	✓	x	<u>0.1223</u>	<u>0.6107</u>	<u>0.2270</u>	<u>21.2234</u>

	Mô-đun				L1 ↓	SSIM ↑	LPIPS ↓	FID ↓
	M	R	S	CL				
UFSC	✓	✓	✗	✓	<b>0.1083</b>	<b>0.6406</b>	<b>0.2019</b>	<b>14.7298</b>
	✗	✗	✗	✗	0.3017	0.1793	0.5102	119.9425
	✓	✓	✓	✗	<u>0.1370</u>	<u>0.5731</u>	<u>0.2476</u>	<u>59.5788</u>
	✓	✓	✗	✓	<b>0.1090</b>	<b>0.6377</b>	<b>0.1985</b>	<b>41.1152</b>

### Nhận xét và Thảo luận:

Quan sát từ dữ liệu thực nghiệm cho thấy vai trò nền tảng không thể thay thế của các mô-đun **M** và **R**. Khi tích hợp hai mô-đun này vào mạng Baseline, hiệu năng mô hình có sự chuyển biến mang tính bước ngoặt, thể hiện qua việc **chỉ số FID giảm sâu** ở cả hai hướng nghiên cứu. Đơn cử như trong kịch bản e2c (UFSC), việc có M và R giúp FID giảm từ **70.36** xuống **29.10** (tương ứng với cấu hình FontDiffuser Gốc). Điều này khẳng định rằng mạng Diffusion thuận tuý gấp rất nhiều khó khăn trong việc định hình cấu trúc ký tự phức tạp nếu chỉ dựa vào đặc trưng cấp cao; M và R chính là “bộ khung xương” cung cấp các đặc trưng nội dung chi tiết đa tầng và tinh chỉnh độ khớp không gian, giúp mô hình dựng hình chính xác các nét và bộ thủ.

Tuy nhiên, điểm nhấn quan trọng nhất nằm ở sự so sánh giữa mô-đun **S (SCR gốc)** và **CL (CL-SCR đề xuất)**. Kết quả thực nghiệm cho thấy **CL-SCR** vượt trội hơn hẳn so với SCR gốc, đặc biệt là trong các kịch bản khó (**Unseen Font**). Cụ thể, trong hướng **e2c** (UFSC), việc thay thế S bằng CL giúp FID giảm mạnh từ **29.10** xuống **13.55**. Tương tự ở hướng **c2e** (UFSC), FID giảm từ **59.58** xuống **41.11**.

**Lý giải:** SCR gốc vốn được thiết kế cho bài toán đơn ngôn ngữ, nơi khoảng cách giữa các phong cách nhỏ hơn. Khi áp dụng cho bài toán đa ngôn ngữ (**Cross-Lingual**), SCR gốc gặp khó khăn trong việc tách biệt triệt để phong cách khỏi nội dung do sự khác biệt lớn về hình thái học. Ngược lại, CL-SCR với **cơ chế tương phản đa miền và chiến lược lấy mẫu âm cải tiến** đã giúp mô hình “hiểu” và trích xuất được bản chất phong cách (như kết cấu, bút pháp) một cách trừu tượng hơn, qua đó đảm bảo chất lượng sinh ảnh ổn định và tự nhiên ngay cả với các font chữ mới lạ.

**Bảng 4.8** — So sánh kết quả sinh ảnh giữa các mô-đun khác nhau trên tập dữ liệu chưa từng thấy cho hai hướng tác vụ (e2c và c2e).

	Mô-đun M R S CL	Example 1	Example 2
UFSC (e2c)	x x x x	二 口 首 音	𠂇 C O
	x x ✓ x	默 首 音	默 首 音
	x x x ✓	默 首 音	默 首 音
	Target	默 首 音	默 首 音
UFSC (c2e)	x x x x	圭 留 田	𠂇 美 𠂇
	x x ✓ x	t d k	t d k
	x x x ✓	t d k	t d k
	Target	t d k	t d k

**Kết luận:** Tổng hợp lại, kết quả nghiên cứu bóc tách đã làm sáng tỏ vai trò riêng biệt và bổ trợ lẫn nhau của các thành phần kiến trúc. Trong khi **MCA** và **RSI** đóng vai trò là nền tảng cấu trúc không thể thiếu để ngăn chặn sự sụp đổ hình dáng ký tự, thì **CL-SCR** chính là nhân tố quyết định nâng tầm chất lượng thị giác và khả năng tổng quát hoá. Việc CL-SCR giúp giảm sâu chỉ số **FID** trên các **tập dữ liệu lạ (UFSC)** so với SCR gốc khẳng định rằng cơ chế tương phản đa ngôn ngữ là chìa khoá để mô hình vượt qua rào cản hình thái học, cho phép chuyển giao phong cách Latin sang Hán tự một cách tự nhiên và linh hoạt hơn.

#### 4.5.2. Ảnh hưởng của Tăng cường dữ liệu (Data Augmentation)

Mục tiêu của nghiên cứu này là đánh giá vai trò của **chiến lược tăng cường dữ liệu**, cụ thể là kỹ thuật **Random Resized Crop** (cắt và thay đổi tỷ lệ ngẫu nhiên) được áp dụng trong quá trình huấn luyện mô-đun **CL-SCR**. Về mặt lý thuyết, việc tăng cường dữ liệu giúp mô hình học được **các đặc trưng phong cách bất biến theo tỷ lệ** và tránh hiện tượng **quá khớp (overfitting)**. Để kiểm chứng điều này, khoa luận **so sánh hiệu năng** của mô hình tiêu biểu ( $Ours_{AZ}$  cho hướng e2c và  $Ours_{All}$  cho hướng c2e) trong hai cấu hình: **có và không có Augmentation**.

Kết quả thực nghiệm được trình bày chi tiết tại [Bảng 4.9](#) và [Bảng 4.10](#).

**Bảng 4.9** — Ảnh hưởng của tăng cường dữ liệu đối với hiệu năng mô hình trên tác vụ Latin → Hán tự (e2c).

	Phương pháp	L1 ↓	SSIM ↑	LPIPS ↓	FID ↓
SFUC	w/o Augment	0.1974	0.3831	0.2967	14.1295
	w/ Augment	<b>0.1939</b>	<b>0.3890</b>	<b>0.2911</b>	<b>11.7691</b>
UFSC	w/o Augment	0.2295	0.3066	0.3060	15.7706
	w/ Augment	<b>0.2214</b>	<b>0.3197</b>	<b>0.2954</b>	<b>13.5508</b>

**Bảng 4.10** — Ảnh hưởng của tăng cường dữ liệu đối với hiệu năng mô hình trên tác vụ Hán tự → Latin (c2e).

	Phương pháp	L1 ↓	SSIM ↑	LPIPS ↓	FID ↓
SFUC	w/o Augment	<b>0.1076</b>	<b>0.6504</b>	<b>0.1978</b>	<b>12.3668</b>
	w/ Augment	0.1083	0.6406	0.2019	14.7298
UFSC	w/o Augment	0.1126	0.6364	0.2015	43.0665
	w/ Augment	<b>0.1090</b>	<b>0.6377</b>	<b>0.1985</b>	<b>41.1152</b>

**Nhận xét và Thảo luận:**

Đối với hướng chuyển đổi từ Latin sang Hán tự ( e2c ), quan sát tại [Bảng 4.9](#) cho thấy việc áp dụng Augmentation mang lại sự cải thiện **toàn diện và nhất quán** trên mọi chỉ

số ở cả hai kịch bản SFUC và UFSC. **Đáng chú ý nhất là chỉ số FID trên tập UFSC giảm mạnh từ 15.77 xuống 13.55**, tương ứng với mức cải thiện **khoảng 14%**. Điều này có thể được lý giải bởi **đặc thù cấu trúc đơn giản** của ký tự Latin đóng vai trò là ảnh phong cách. Nếu thiếu đi sự đa dạng hoá dữ liệu thông qua Augmentation, mô hình dễ bị phụ thuộc vào các đặc trưng vị trí không gian cố định. Kỹ thuật **Random Resized Crop** buộc mô-đun CL-SCR phải tập trung học các **đặc trưng bản chất** như độ dày nét, serif hay độ tương phản bất kể biến đổi về kích thước hay vị trí, từ đó giúp quá trình áp dụng phong cách lên cấu trúc phức tạp của Hán tự trở nên linh hoạt và tự nhiên hơn.

Trong khi đó, hướng chuyển đổi ngược lại từ Hán tự sang Latin (c2e) tại **Bảng 4.10** lại hé lộ một sự đánh đổi thú vị giữa khả năng **ghi nhớ và khái quát hóa**. Trên tập dữ liệu đã biết (SFUC), cấu hình không có Augmentation đạt kết quả tốt hơn với FID 12.36 so với 14.72. Tuy nhiên, ưu thế **đảo chiều hoàn toàn** trên tập dữ liệu chưa biết (UFSC), nơi cấu hình có Augmentation giành lại vị thế dẫn đầu với FID giảm từ **43.06** xuống **41.11** và sai số L1 cũng được cải thiện. Hiện tượng này minh chứng rõ ràng cho **vai trò điều hòa (Regularization)** của tăng cường dữ liệu. Ở kịch bản SFUC, việc thiếu nhiều cho phép mô hình **tối ưu hóa cục bộ (overfit)** trên các mẫu đã thấy, dẫn đến chỉ số cao nhưng kém bền vững. Ngược lại, khi đối mặt với dữ liệu lạ trong UFSC, khả năng ghi nhớ trở nên vô hiệu, và lúc này các **đặc trưng phong cách cốt lõi** mang tính khái quát cao mà mô hình học được nhờ **Augmentation** mới thực sự phát huy tác dụng. Vì vậy, kết quả vượt trội trên UFSC khẳng định rằng tăng cường dữ liệu là thành phần thiết yếu để đảm bảo **khả năng tổng quát hóa** của mô hình trong các ứng dụng thực tế.

**Bảng 4.11** — So sánh kết quả sinh ảnh giữa mô hình có và không áp dụng tăng cường dữ liệu trên tập dữ liệu chưa từng thấy cho hai hướng tác vụ (e2c và c2e).

	Phương pháp	Example 1	Example 2
UFSC (e2c)	w/ Augment	默 首 音	默 首 音
	w/o Augment	默 首 音	默 首 音
	Target	默 首 音	默 首 音
UFSC (c2e)	w/ Augment	t d k	t d k
	w/o Augment	t d k	t d k
	Target	t d k	t d k

**Kết luận:** Dựa trên phân tích trên, khoá luận khẳng định **chiến lược Tăng cường dữ liệu** là thành phần không thể thiếu, đặc biệt quan trọng để nâng cao hiệu suất trên các **dữ liệu chưa từng biết (Unseen Domains)**, mặc dù có thể đánh đổi một lượng nhỏ hiệu năng trên các dữ liệu đã biết.

#### 4.5.3. Ảnh hưởng của Chế độ hàm loss (Loss Mode)

Trong kiến trúc CL-SCR, hàm mất mát **InfoNCE**[25] đóng vai trò điều hướng không gian biểu diễn phong cách. Khoá luận khảo sát **ba biến thể chiến lược huấn luyện** được định nghĩa trong tham số `loss_mode` : `scr_intra` : Chỉ sử dụng mẫu âm nội miền (Intra-domain). Ví dụ: so sánh Style Latin với các Style Latin khác. `scr_cross` : Chỉ sử dụng mẫu âm xuyên miền (Cross-domain). Ví dụ: so sánh Style Latin với Style Hán tự. `scr_both` : Kết hợp cả hai với trọng số  $\alpha_{\text{intra}} = 0.3$  và  $\beta_{\text{cross}} = 0.7$ .

Kết quả thực nghiệm được trình bày tại [Bảng 4.12](#) và [Bảng 4.13](#).

**Bảng 4.12** — Ảnh hưởng của các chế độ loss đối với hiệu năng mô hình trên tác vụ Latin → Hán tự (e2c).

	Chế độ mất mát	L1 ↓	SSIM ↑	LPIPS ↓	FID ↓
SFUC	scr_intra	<u>0.1969</u>	<u>0.3812</u>	<u>0.2958</u>	11.9552
	scr_cross	0.1993	0.3770	0.2982	<u>11.8645</u>
	scr_both	<b>0.1939</b>	<b>0.3890</b>	<b>0.2911</b>	<b>11.7691</b>
UFSC	scr_intra	<u>0.2290</u>	<u>0.3008</u>	<u>0.3085</u>	<u>15.7197</u>
	scr_cross	0.2326	0.2911	0.3128	16.2615
	scr_both	<b>0.2214</b>	<b>0.3197</b>	<b>0.2954</b>	<b>13.5508</b>

**Bảng 4.13** — Ảnh hưởng của các chế độ loss đối với hiệu năng mô hình trên tác vụ Hán tự → Latin (c2e).

	Chế độ mất mát	L1 ↓	SSIM ↑	LPIPS ↓	FID ↓
SFUC	scr_intra	<b>0.0993</b>	<b>0.6614</b>	<b>0.1903</b>	<b>13.6449</b>
	scr_cross	0.1091	<u>0.6436</u>	<u>0.2017</u>	<u>14.0159</u>
	scr_both	<u>0.1083</u>	0.6406	0.2019	14.7298
UFSC	scr_intra	<b>0.0971</b>	<b>0.6601</b>	<b>0.1845</b>	<u>41.3399</u>
	scr_cross	0.1175	0.6209	0.2095	44.7758
	scr_both	<u>0.1090</u>	<u>0.6377</u>	<u>0.1985</u>	<b>41.1152</b>

#### Nhận xét và Thảo luận:

Đối với hướng chuyên đổi từ Latin sang Hán tự ( e2c ), số liệu tại [Bảng 4.12](#) phản ánh **sự thống trị rõ rệt của chiến lược hỗn hợp** `scr_both` trên hầu hết các chỉ số, đặc biệt là sự cải thiện vượt bậc về chỉ số FID trong kịch bản khó UFSC (đạt **13.55** so với 15.72 của `scr_intra` và 16.26 của `scr_cross` ). Kết quả này có thể được lý giải bởi **đặc thù thông tin “thưa” (sparse)** của phong cách Latin. Nếu chỉ sử dụng so sánh nội miền `scr_intra`, mô hình khó học được cách các đặc trưng Latin đơn giản

tương tác với cấu trúc Hán tự phức tạp; ngược lại, nếu chỉ dùng `scr_cross`, khoảng cách miền quá lớn lại gây ra sự bất ổn định trong quá trình hội tụ. Do đó, sự kết hợp trong `scr_both` đóng vai trò như **cầu nối**, giúp mô hình vừa nắm bắt vững chắc đặc trưng nội tại của Latin, vừa học được mối tương quan ngữ nghĩa với Hán tự để tạo ra kết quả tối ưu.

Bức tranh trở nên phức tạp và thú vị hơn khi xét đến chiều ngược lại từ Hán tự sang Latin ( `c2e` ) tại [Bảng 4.13](#), nơi xuất hiện một **nghịch lý về độ giàu thông tin**. Khác với hướng `e2c`, chiến lược `scr_intra` **lại thể hiện sự vượt trội về các chỉ số cấu trúc và điểm ảnh**(L1 thấp nhất 0.097, SSIM cao nhất) trên cả hai tập dữ liệu. Nguyên nhân sâu xa nằm ở bản chất “**dense**” (**dense**) **và giàu thông tin** của phong cách Hán tự (nét bút, độ dày, kết cấu). Chỉ cần **so sánh nội bộ giữa các Hán tự** là đã đủ để mô hình trích xuất được một vector phong cách mạnh mẽ. Trong bối cảnh này, việc ép buộc so sánh xuyên miền với Latin (thông qua thành phần cross trong `scr_both` ) vô tình tạo ra nhiều do sự khác biệt quá lớn về cấu trúc, làm giảm nhẹ độ chính xác tái tạo. Tuy nhiên, `scr_both` **vẫn giữ được ưu thế về độ tự nhiên tổng thể** (FID 41.11 so với 41.34) trên tập lạ UFSC, đóng vai trò như một cơ chế điều hòa cần thiết để đảm bảo tính thẩm mỹ khi đối mặt với các font hoàn toàn mới.

**Bảng 4.14** — So sánh kết quả sinh ảnh giữa các chế độ mát mát khác nhau trên tập dữ liệu chưa từng thấy cho hai hướng tác vụ (e2c và c2e).

	Chế độ mát mát	Example 1	Example 2
UFSC (e2c)	scr_intra	默 首 音	默 首 音
	scr_cross	默 首 音	默 首 音
	scr_both	默 首 音	默 首 音
	Target	默 首 音	默 首 音
UFSC (c2e)	scr_intra	t d k	t d k
	scr_cross	t d k	t d k
	scr_both	t d k	t d k
	Target	t d k	t d k

**Kết luận:** Tổng kết lại, đối với bài toán tổng quát, **chiến lược scr\_both là lựa chọn an toàn và ổn định nhất** để cân bằng giữa độ chính xác và tính tự nhiên. Tuy nhiên, thực nghiệm cũng mở ra một góc nhìn quan trọng: khi miền nguồn có lượng thông tin phong phú như Hán tự, **chiến lược học nội miền (scr\_intra) cũng mang lại hiệu quả rất ấn tượng**, gợi ý tiềm năng tối ưu hóa chi phí huấn luyện cho các ứng dụng cụ thể mà không nhất thiết phải phụ thuộc vào dữ liệu cặp đôi xuyên ngôn ngữ.

#### 4.5.4. Ảnh hưởng của Số lượng mẫu âm (Negative Sample Numbers)

Trong khuôn khổ của **phương pháp học tương phản** (Contrastive Learning), số lượng mẫu âm ( $K$ ) đóng vai trò quan trọng trong việc định hình không gian biểu diễn đặc trưng. Theo lý thuyết thông thường, việc tăng số lượng mẫu âm thường giúp mô hình phân biệt tốt hơn giữa các đặc trưng phong cách, từ đó học được các biểu diễn phong phú hơn. Để kiểm chứng giả thuyết này trong bối cảnh sinh phông chữ đa ngôn ngữ, khoá luận tiến hành thực nghiệm với các giá trị  $K$  lần lượt là 4, 8 và 16 trên cả hai hướng chuyển đổi. Kết quả chi tiết được tổng hợp tại [Bảng 4.15](#) và [Bảng 4.16](#).

**Bảng 4.15** — Ảnh hưởng của số lượng mẫu âm đối với hiệu năng mô hình trên tác vụ Latin → Hán tự (e2c).

	Số lượng mẫu âm	L1 ↓	SSIM ↑	LPIPS ↓	FID ↓
SFUC	4	<b>0.1939</b>	<b>0.3890</b>	<b>0.2911</b>	<u>11.7691</u>
	8	0.1972	<u>0.3835</u>	<u>0.2952</u>	12.3750
	16	<u>0.1967</u>	0.3833	0.2956	<b>10.6901</b>
UFSC	4	<b>0.2214</b>	<b>0.3197</b>	<b>0.2954</b>	<b>13.5508</b>
	8	0.2285	0.3048	0.3061	<u>15.0245</u>
	16	<u>0.2273</u>	<u>0.3064</u>	<u>0.3048</u>	16.7855

**Bảng 4.16** — Ảnh hưởng của số lượng mẫu âm đối với hiệu năng mô hình trên tác vụ Hán tự → Latin (c2e).

	Số lượng mẫu âm	L1 ↓	SSIM ↑	LPIPS ↓	FID ↓
SFUC	4	0.1083	0.6406	0.2019	<b>14.7298</b>
	8	<u>0.1080</u>	<u>0.6464</u>	<u>0.1999</u>	<u>14.8365</u>
	16	<b>0.1059</b>	<b>0.6468</b>	<b>0.1992</b>	15.7326
UFSC	4	<u>0.1090</u>	<u>0.6377</u>	<b>0.1985</b>	<b>41.1152</b>
	8	<b>0.1087</b>	<b>0.6398</b>	<b>0.1985</b>	43.8077

Số lượng mẫu âm	L1 ↓	SSIM ↑	LPIPS ↓	FID ↓
16	0.1111	0.6311	<u>0.2008</u>	<u>43.5042</u>

### Nhận xét và Thảo luận:

Phân tích số liệu từ thực nghiệm cho thấy một kết quả **trái ngược với trực giác phổ biến trong học tương phản** trên các tác vụ thị giác máy tính khác. Cụ thể, trong hướng chuyển đổi từ Latin sang Hán tự ([Bảng 4.15](#)), cấu hình sử dụng số lượng mẫu âm nhỏ nhất ( $K = 4$ ) lại thể hiện sự vượt trội về **độ ổn định và khả năng tổng quát hóa**. Trên tập kiểm thử khó UFSC, cấu hình này đạt chỉ số FID tốt nhất là **13.55**, thấp hơn đáng kể so với mức 16.78 khi sử dụng 16 mẫu âm. Đồng thời, các chỉ số về cấu trúc như SSIM và sai số L1 cũng đạt giá trị tối ưu tại  $K = 4$ . Điều này gợi ý rằng đối với hệ chữ Latin vốn có cấu trúc nét tương đối đơn giản và “thưa”, việc sử dụng quá nhiều mẫu âm có thể vô tình đưa vào các **tín hiệu nhiễu** hoặc các mẫu có phong cách quá tương đồng (**false negatives**), khiến mô hình bị rối loạn trong việc định vị biên giới phong cách, dẫn đến suy giảm hiệu năng trên dữ liệu chưa từng thấy.

Xu hướng tương tự cũng được quan sát thấy ở chiều ngược lại từ Hán tự sang Latin ([Bảng 4.16](#)), mặc dù có sự phân hoá nhẹ giữa khả năng ghi nhớ và khái quát hóa. Khi đánh giá trên tập font đã biết (SFUC), việc tăng số lượng mẫu âm lên 16 giúp cải thiện nhẹ các chỉ số điểm ảnh như L1 và SSIM, do mô hình tận dụng được nhiều dữ liệu so sánh hơn để khớp chi tiết các nét phức tạp của Hán tự. Tuy nhiên, lợi thế này **không duy trì được khi chuyển sang tập font lạ (UFSC)**. Tại đây, cấu hình  $K = 4$  một lần nữa khẳng định tính hiệu quả với chỉ số FID thấp nhất (**41.11**), vượt qua cả cấu hình  $K = 8$  và  $K = 16$ . Kết quả này cũng cố nhận định rằng trong bài toán chuyển đổi đa ngôn ngữ với sự chênh lệch lớn về miền dữ liệu, một tập hợp mẫu âm **nhỏ nhưng tinh gọn** sẽ hiệu quả hơn việc cố gắng phân biệt với một lượng lớn mẫu âm có thể gây nhiễu. Do đó, việc lựa chọn  $K = 4$  không chỉ giúp **tối ưu hóa tài nguyên tính toán** mà còn đảm bảo chất lượng sinh ảnh tốt nhất về mặt thị giác.

**Bảng 4.17** — So sánh kết quả sinh ảnh giữa các số lượng mẫu âm khác nhau trên tập dữ liệu chưa từng thấy cho cả hai hướng tác vụ (e2c và c2e).

	Số lượng mẫu âm	Example 1	Example 2
UFSC (e2c)	4	默 首 音	默 首 音
	8	默 首 音	默 首 音
	16	默 首 音	默 首 音
	Target	默 首 音	默 首 音
UFSC (c2e)	4	t d k	t d k
	8	t $\square$ k	t d k
	16	t d $\mathbb{E}$	t d k
	Target	t d k	t d k

**Kết luận:** Tổng kết lại, thực nghiệm về số lượng mẫu âm đã làm sáng tỏ một đặc điểm thú vị trong bài toán chuyển đổi phong cách xuyên ngôn ngữ: **sự tối giản lại mang lại hiệu quả tối ưu**. Trái với kỳ vọng rằng nhiều mẫu âm sẽ giúp học biểu diễn phong cách tốt hơn, kết quả cho thấy việc **giới hạn  $K = 4$**  giúp mô hình xây dựng được **không gian biểu diễn phong cách cô đọng**, tránh được hiện tượng quá khớp (overfitting) hoặc nhiễu loạn thông tin từ các mẫu âm dư thừa. Đặc biệt trên các tập dữ liệu chưa từng thấy (UFSC), cấu hình  $K = 4$  luôn duy trì vị thế dẫn đầu về chỉ số FID ở cả hai hướng chuyển đổi, chứng minh đây là **thiết lập tối ưu** để cân bằng giữa

độ chính xác tái tạo và khả năng tổng quát hoá, đồng thời **giảm tải đáng kể chi phí huấn luyện**.

#### 4.5.5. Ảnh hưởng của Alpha và Beta

Trong kiến trúc CL-SCR được đề xuất, hàm mất mát tổng thể được thiết lập dưới dạng tổng trọng số của hai thành phần: mất mát nội tại (Intra-Lingual loss) và mất mát chéo (Cross-Lingual loss), tuân theo công thức:  $L_{CL-SCR} = \alpha L_{intra} + \beta L_{cross}$ . Trong đó,  $\alpha$  điều chỉnh mức độ tập trung vào việc bảo toàn tính nhất quán phong cách trong cùng một ngôn ngữ, còn  $\beta$  kiểm soát lực ràng buộc để kéo các biểu diễn phong cách của hai ngôn ngữ lại gần nhau trong không gian đặc trưng. Để xác định tỷ lệ tối ưu giữa hai cơ chế này, khoa luận tiến hành khảo sát thực nghiệm với ba cấu hình trọng số đổi ngẫu ( $\alpha, \beta$ ) lần lượt là  $(0.3, 0.7)$ ,  $(0.5, 0.5)$  và  $(0.7, 0.3)$ . Mục tiêu là phân tích sự đánh đổi giữa khả năng tái tạo chi tiết (do  $\alpha$  chi phối) và khả năng chuyển đổi phong cách liên ngôn ngữ (do  $\beta$  chi phối) trên cả hai chiều bài toán. Kết quả chi tiết được tổng hợp tại [Bảng 4.18](#) và [Bảng 4.19](#).

**Bảng 4.18** — Ảnh hưởng của alpha và beta đối với hiệu năng mô hình trên tác vụ Latin → Hán tự (e2c).

	Phương pháp	L1 ↓	SSIM ↑	LPIPS ↓	FID ↓
SFUC	$\alpha = 0.3, \beta = 0.7$	<b>0.1939</b>	<b>0.3890</b>	<u>0.2911</u>	11.7691
	$\alpha = 0.5, \beta = 0.5$	0.1964	<u>0.3855</u>	0.2934	<u>11.1352</u>
	$\alpha = 0.7, \beta = 0.3$	<u>0.1963</u>	0.3827	<b>0.2908</b>	<b>10.3742</b>
UFSC	$\alpha = 0.3, \beta = 0.7$	<b>0.2214</b>	<b>0.3197</b>	<b>0.2954</b>	<b>13.5508</b>
	$\alpha = 0.5, \beta = 0.5$	0.2277	0.3088	0.3026	15.1777
	$\alpha = 0.7, \beta = 0.3$	<u>0.2264</u>	<u>0.3095</u>	<u>0.2991</u>	<u>14.4760</u>

**Bảng 4.19** — Ảnh hưởng của alpha và beta đối với hiệu năng mô hình trên tác vụ Hán tự → Latin (c2e).

	Phương pháp	L1 ↓	SSIM ↑	LPIPS ↓	FID ↓
	$\alpha = 0.3, \beta = 0.7$	<u>0.1083</u>	<u>0.6406</u>	<u>0.2019</u>	<b>14.7298</b>

	<b>Phương pháp</b>	<b>L1 ↓</b>	<b>SSIM ↑</b>	<b>LPIPS ↓</b>	<b>FID ↓</b>
<b>SFUC</b>	$\alpha = 0.5, \beta = 0.5$	0.1099	0.6392	0.2051	<u>15.5683</u>
	$\alpha = 0.7, \beta = 0.3$	<b>0.1072</b>	<b>0.6432</b>	<b>0.2002</b>	16.3548
<b>UFSC</b>	$\alpha = 0.3, \beta = 0.7$	<u>0.1090</u>	<u>0.6377</u>	<u>0.1985</u>	<b>41.1152</b>
	$\alpha = 0.5, \beta = 0.5$	<b>0.1053</b>	<b>0.6434</b>	<b>0.1957</b>	<u>43.4240</u>
	$\alpha = 0.7, \beta = 0.3$	0.1115	0.6287	0.2014	45.2293

### **Nhận xét và Thảo luận:**

Kết quả thực nghiệm cho thấy vai trò đối trọng thú vị giữa **tính nhất quán nội tại (Intra-Lingual)** và **sự ràng buộc xuyên ngôn ngữ (Cross-Lingual)**. Đối với hướng chuyển đổi từ Latin sang Hán tự (Bảng 4.18), ta quan sát thấy sự đảo chiều về hiệu năng giữa kịch bản quen thuộc (SFUC) và kịch bản lạ (UFSC). Trên tập SFUC, cấu hình ưu tiên tính nội tại ( $\alpha = 0.7, \beta = 0.3$ ) đạt kết quả FID tốt nhất (**10.37**), cho thấy khi phong cách đã biết, việc tập trung tinh chỉnh cấu trúc nội bộ của Hán tự giúp ảnh sinh sắc nét hơn. Tuy nhiên, trên tập kiểm thử khó UFSC, cấu hình ưu tiên liên kết chéo ( $\alpha = 0.3, \beta = 0.7$ ) lại vượt trội với FID đạt **13.55** (so với **14.47** và **15.17**). Điều này gợi ý rằng để tổng quát hóa tốt trên các font chữ chưa từng thấy, mô hình cần dựa nhiều hơn vào “cấu nối” tương đồng giữa hai ngôn ngữ ( $\beta$ ) thay vì quá tập trung vào đặc trưng cục bộ của từng hệ chữ.

Xu hướng này trở nên nhất quán và rõ rệt hơn ở chiều ngược lại từ Hán tự sang Latin (Bảng 4.19). Trong cả hai kịch bản SFUC và UFSC, việc gán trọng số cao cho thành phần Cross-Lingual ( $\beta = 0.7$ ) đều mang lại hiệu suất FID tối ưu (**14.73** và **41.11**). Nguyên nhân có thể xuất phát từ **khoảng cách thông tin (information gap)**: Hán tự có cấu trúc phức tạp và giàu thông tin hơn nhiều so với Latin. Do đó, khi sinh chữ Latin từ nguồn Hán, mô hình cần một cơ chế ràng buộc xuyên ngôn ngữ mạnh mẽ ( $\beta$  lớn) để định hướng việc lọc bỏ các chi tiết thừa và ánh xạ chính xác phong cách, thay vì bị “sa lầy” vào việc học cấu trúc nội tại phức tạp của Hán tự ( $\alpha$ ). Kết quả này khẳng định rằng trong bài toán Cross-Lingual bất đối xứng, **tăng cường giám sát liên ngôn ngữ** là chìa khoá để cải thiện chất lượng sinh ảnh và độ tự nhiên thị giác.

**Bảng 4.20** — So sánh kết quả sinh ảnh giữa các alpha và beta khác nhau trên tập dữ liệu chưa từng thấy cho cả hai hướng tác vụ (e2c và c2e).

	Phương pháp	Example 1	Example 2
UFSC (e2c)	$\alpha = 0.3, \beta = 0.7$	默 首 音	默 首 音
	$\alpha = 0.5, \beta = 0.5$	默 首 音	默 首 音
	$\alpha = 0.7, \beta = 0.3$	默 首 音	默 首 音
	Target	默 首 音	默 首 音
UFSC (c2e)	$\alpha = 0.3, \beta = 0.7$	t d k	t d k
	$\alpha = 0.5, \beta = 0.5$	t d k	t d k
	$\alpha = 0.7, \beta = 0.3$	t d k	t d k
	Target	t d k	t d k

**Kết luận:** Tổng kết lại, thực nghiệm về trọng số  $\alpha$  và  $\beta$  đã chỉ ra sự bất đối xứng về nhu cầu giám sát của mô hình. Trong khi thành phần Intra-Lingual ( $\alpha$ ) chỉ thực sự phát huy tác dụng tối đa trong các kịch bản dữ liệu đã biết, thì thành phần **Cross-Lingual ( $\beta$ ) lại đóng vai trò chủ đạo** trong các tác vụ yêu cầu khả năng khai quật hoá cao hoặc chuyển đổi từ tập mẫu phức tạp sang đơn giản. Dựa trên kết quả này, khoá luận đề xuất cấu hình ưu tiên liên kết chéo ( $\alpha = 0.3, \beta = 0.7$ ) là thiết lập mặc định cho mô hình cuối cùng, nhằm tối ưu hoá hiệu suất cho các ứng dụng thực tế nơi dữ liệu đầu vào thường xuyên biến đổi và chưa biết trước.

#### 4.5.6. Ảnh hưởng của Trọng số hướng dẫn (Guidance Scale)

Trong cơ chế sinh ảnh của mô hình khuếch tán (Diffusion Models), **Trọng số hướng dẫn (Guidance Scale,  $s$ )** đóng vai trò như một “cần gạt” kiểm soát sự cân bằng giữa độ đa dạng của ảnh sinh và độ bám sát vào điều kiện đầu vào (content/style). Theo nguyên lý của phương pháp Classifier-free Guidance[28] được áp dụng trong FontDiffuser, công thức cập nhật mẫu là:

$$\epsilon_{\text{pred}} = \epsilon_{\text{uncond}} + s(\epsilon_{\text{cond}} - \epsilon_{\text{uncond}}) \quad (4.27)$$

Về mặt lý thuyết, việc tăng giá trị  $s$  sẽ ép buộc mô hình tuân thủ chặt chẽ hơn các đặc trưng phong cách mục tiêu, nhưng nếu  $s$  quá lớn sẽ dẫn đến hiện tượng bão hòa (saturation) và xuất hiện các chi tiết giả (artifacts). Để tìm ra “điểm ngọt” (sweet spot) cho tác vụ sinh font chữ, khoá luận thực hiện khảo sát với giải giá trị  $s$  chạy từ 2.5 đến 15.

**Bảng 4.21** — Ảnh hưởng của trọng số hướng dẫn đối với hiệu năng mô hình trên tác vụ Latin → Hán tự (e2c).

	Trọng số hướng dẫn	L1 ↓	SSIM ↑	LPIPS ↓	FID ↓
SFUC	2.5	0.1982	0.3812	0.2957	14.1162
	5	0.1955	0.3861	0.2922	12.8616
	7.5	0.1939	0.3890	<b>0.2911</b>	<u>11.7691</u>
	10	0.1932	<b>0.3894</b>	<u>0.2921</u>	<b>11.5753</b>
	12.5	<b>0.1927</b>	<u>0.3893</u>	0.2936	12.3513
	15	<u>0.1929</u>	0.3874	0.2971	14.1336
UFSC	2.5	0.2262	0.3096	0.2985	<b>13.2760</b>
	5	0.2229	0.3176	<u>0.2955</u>	<u>13.3922</u>
	7.5	0.2214	<b>0.3197</b>	<b>0.2954</b>	13.5508
	10	0.2209	<u>0.3194</u>	0.2970	13.7769

Trọng số hướng dẫn	L1 ↓	SSIM ↑	LPIPS ↓	FID ↓
12.5	<u>0.2207</u>	0.3187	0.2991	14.7846
15	<b>0.2204</b>	0.3185	0.3025	17.0116

**Bảng 4.22** — Ảnh hưởng của trọng số hướng dẫn đối với hiệu năng mô hình trên tác vụ Hán tự → Latin (c2e).

Trọng số hướng dẫn	L1 ↓	SSIM ↑	LPIPS ↓	FID ↓
SFUC	2.5	0.1108	0.6369	0.2027
	5	0.1093	<b>0.6395</b>	<b>0.2010</b>
	7.5	0.1083	<u>0.6406</u>	<u>0.2019</u>
	10	0.1075	<b>0.6408</b>	0.2038
	12.5	<u>0.1070</u>	0.6402	0.2077
	15	<b>0.1069</b>	0.6385	0.2129
UFSC	2.5	0.1096	0.6352	<u>0.2002</u>
	5	0.1060	<b>0.6418</b>	<b>0.1944</b>
	7.5	0.1090	0.6377	0.1985
	10	0.1070	0.6391	0.2014
	12.5	<b>0.1025</b>	<b>0.6477</b>	<u>0.1976</u>
	15	<u>0.1031</u>	<u>0.6426</u>	0.2045

#### Nhận xét và Thảo luận:

Kết quả thực nghiệm tại hai bảng trên cho thấy một xu hướng **nhạy cảm ngược chiều** so với các tác vụ sinh ảnh thông thường (nơi  $s$  thường được đặt quanh mức 7.5). Cụ thể, trong tác vụ Latin sang Hán tự (Bảng 4.21), các chỉ số chất lượng đạt đỉnh ở mức

guidance scale trung bình thấp. Trên tập dữ liệu đã biết (SFUC), giá trị FID tối ưu nằm tại ngưỡng  $s = 10$  (11.57), tuy nhiên sự chênh lệch so với  $s = 7.5$  là không đáng kể. Đáng chú ý, khi chuyển sang tập dữ liệu lạ (UFSC), việc giữ guidance scale ở mức thấp (2.5 – 7.5) giúp duy trì chỉ số FID ổn định nhất (quanh mức 13.5), trong khi việc tăng  $s$  lên 15 khiến chất lượng ảnh suy giảm rõ rệt (FID tăng vọt lên 17.01). Điều này gợi ý rằng đối với các cấu trúc phức tạp như Hán tự, việc cưỡng ép mô hình quá mức bằng guidance scale cao sẽ làm mất đi tính tự nhiên của nét bút.

Hiện tượng này càng trở nên cực đoan hơn ở chiều ngược lại từ Hán tự sang Latin ([Bảng 4.22](#)). Dữ liệu chỉ ra rằng mô hình đạt hiệu suất tốt nhất tại các mức guidance scale **rất thấp ( $s = 2.5$  hoặc  $s = 5$ )**. Trên tập UFSC, cấu hình  $s = 5$  đạt FID tốt nhất là **40.00**, trong khi việc tăng  $s$  lên 15 khiến chỉ số này tệ đi gần 30% (lên mức 52.75). Nguyên nhân cốt lõi nằm ở việc không gian phong cách của Hán tự dày đặc hơn rất nhiều so với Latin. Khi sử dụng guidance scale lớn để ép các đặc trưng phong cách phong phú của Hán tự vào khung xương đơn giản của chữ Latin, mô hình dễ sinh ra các nhiễu (artifacts) hoặc biến dạng cấu trúc không mong muốn. Các chỉ số về cấu trúc như SSIM và L1 cũng đồng thuận với nhận định này khi đạt giá trị tối ưu ở ngưỡng thấp ( $s \leq 7.5$ ).

**Bảng 4.23** — So sánh kết quả sinh ảnh giữa các trọng số hướng dẫn khác nhau trên tập dữ liệu chưa từng thấy cho cả hai hướng tác vụ (e2c và c2e).

	Trọng số hướng dẫn	Example 1	Example 2
UFSC (e2c)	2.5	默 首 音 音	默 首 音 音
	5	默 首 音 音	默 首 音 音
	7.5	默 首 音 音	默 首 音 音
	10	默 首 音 音	默 首 音 音
	12.5	默 首 音 音	默 首 音 音
	15	默 首 音 音	默 首 音 音
	Target	默 首 音 音	默 首 音 音
UFSC (c2e)	2.5	t d k	t d k
	5	t d k	t d k
	7.5	t d k	t d k
	10	t d k	t d k
	12.5	t d k	t d k
	15	t d k	t d k

Target	<b>t d k</b>	<i>t d k</i>
--------	--------------	--------------

**Kết luận:** Tổng kết lại, thực nghiệm khẳng định rằng **Trọng số hướng dẫn thấp đến trung bình** là lựa chọn tối ưu cho bài toán chuyển đổi font chữ đa ngôn ngữ. Khác với các mô hình tạo sinh nghệ thuật cần  $s$  cao để đảm bảo đúng prompt, FontDiffuser hoạt động hiệu quả nhất khi  $s$  nằm trong khoảng [2.5, 7.5]. Việc thiết lập giá trị này giúp mô hình cân bằng tốt nhất giữa việc chuyển tải phong cách và bảo toàn cấu trúc hình học, tránh được các biến dạng do quá khớp (over-exposure). Dựa trên sự ổn định qua các kịch bản thử nghiệm, khoá luận đề xuất thiết lập mặc định  $s = 7.5$  cho chiều Latin-Hán (để cân bằng độ nét) và  $s = 5.0$  cho chiều Hán-Latin (để đảm bảo độ tự nhiên).

# Chương 5

## Kết luận và Hướng phát triển

### 5.1. Kết quả đạt được

Khoá luận đã hoàn thành mục tiêu xây dựng một khung giải pháp toàn diện cho bài toán sinh phông chữ đa ngôn ngữ (Cross-lingual Font Generation), tập trung vào cặp ngôn ngữ có sự chênh lệch hình thái lớn là Latin và Hán tự. Đóng góp quan trọng nhất về mặt lý thuyết là việc đề xuất và tích hợp thành công mô-đun **Cross-Lingual Style Contrastive Refinement (CL-SCR)** vào kiến trúc khuếch tán nền tảng. Khác với các phương pháp tiếp cận trước đây thường gặp khó khăn trong việc tách biệt phong cách khỏi nội dung khi miền dữ liệu thay đổi, CL-SCR đã chứng minh khả năng học được các **biểu diễn phong cách bất biến (invariant style representations)**. Cơ chế này cho phép mô hình vượt qua rào cản về “**Bất cân xứng mật độ thông tin**”, giải quyết hiệu quả cả hai bài toán: ngoại suy phong cách từ cấu trúc Latin đơn giản sang Hán tự phức tạp (**e2c**) và trừu tượng hoá phong cách từ Hán tự dày đặc nét sang Latin (**c2e**) mà không làm vỡ cấu trúc ký tự.

Về mặt thực nghiệm, kết quả định lượng trên tập dữ liệu chuẩn đã khẳng định sự vượt trội của phương pháp đề xuất so với các mô hình State-of-the-Art thuộc dòng GAN (như **DG-Font**, **CF-Font**) và cả mô hình FontDiffuser nguyên bản. Cụ thể, chỉ số **FID** và **LPIPS** được cải thiện đáng kể trên các tập dữ liệu chưa từng thấy (UFSC), chứng tỏ khả năng tổng quát hoá mạnh mẽ của mô hình. Bên cạnh đó, kết quả khảo sát người dùng cũng cho thấy sản phẩm sinh ra từ phương pháp đề xuất đạt độ thẩm mỹ cao, với nét chữ sắc sảo và tự nhiên, khắc phục được các lỗi phổ biến như **mờ nhòe (blurring)** hay **biến dạng (artifacts)** thường thấy ở các mô hình đối chứng. Thành công của khoá luận không chỉ dừng lại ở việc cải thiện các chỉ số đo lường mà còn mở ra hướng đi mới cho việc ứng dụng Generative AI vào lĩnh vực thiết kế đồ họa và tự động hoá quy trình sáng tạo phông chữ đa ngôn ngữ.

## 5.2. Các định hướng phát triển

Mặc dù đã đạt được những kết quả khả quan, nghiên cứu vẫn tồn tại một số hạn chế nhất định, mở ra các hướng phát triển tiềm năng trong tương lai.

Thứ nhất, mở rộng phạm vi ngôn ngữ. Hiện tại mô hình mới chỉ tập trung vào cặp Latin-Hán. Hướng nghiên cứu tiếp theo sẽ thử nghiệm khả năng chuyển đổi trên các hệ chữ viết đa dạng hơn như **Tiếng Việt (Chữ Nôm/Quốc ngữ hoá)**, **Tiếng Nhật (Kanji/Kana)** hay **Tiếng Ả Rập**. Điều này đòi hỏi mô hình phải thích ứng với các đặc trưng hình thái học mới như dấu thanh điệu (diacritics) hoặc tính liên kết nét (cursiveness) đặc thù.

Thứ hai, tối ưu hoá tài nguyên huấn luyện. Mô hình Diffusion hiện tại đòi hỏi chi phí tính toán lớn. Để khắc phục, khoá luận đề xuất áp dụng các kỹ thuật **tinh chỉnh hiệu quả tham số (Parameter-Efficient Fine-Tuning - PEFT)** như **LoRA**[20] hoặc **Adapter**[21]. Việc này sẽ cho phép huấn luyện mô hình trên các GPU phổ thông mà không làm suy giảm đáng kể chất lượng sinh ảnh.

Cuối cùng, về khả năng xử lý phong cách cực đoan, mô hình đôi khi gặp khó khăn với các phông chữ thư pháp biến dạng cao. Giải pháp tiềm năng là tích hợp các cơ chế **chú ý biến dạng (Deformable Attention)** mạnh mẽ hơn hoặc kết hợp với **biểu diễn vector (Vector Graphics)** để nắm bắt tốt hơn các đường cong phức tạp thay vì chỉ dựa vào ảnh raster thuần tuý.

# Tài liệu tham khảo

- [1] I. J. Goodfellow *et al.*, “Generative Adversarial Nets,” in *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada, 2014*, pp. 2672–2680. [Online]. Available: <https://proceedings.neurips.cc/paper/2014/hash/5ca3e9b122f61f8f06494c97b1afccf3-Abstract.html>
- [2] J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, and S. Ganguli, “Deep Unsupervised Learning using Nonequilibrium Thermodynamics,” in *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, in JMLR Workshop and Conference Proceedings, vol. 37. JMLR.org, 2015, pp. 2256–2265. [Online]. Available: <http://proceedings.mlr.press/v37/sohl-dickstein15.html>
- [3] J. Ho, A. Jain, and P. Abbeel, “Denoising Diffusion Probabilistic Models,” in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020*. [Online]. Available: <https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html>
- [4] Z. Yang, D. Peng, Y. Kong, Y. Zhang, C. Yao, and L. Jin, “FontDiffuser: One-Shot Font Generation via Denoising Diffusion with Multi-Scale Content Aggregation and Style Contrastive Learning,” in *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2024, February 20-27, 2024, Vancouver, Canada*, AAAI Press, 2024, pp. 6603–6611. doi: [10.1609/AAAI.V38I7.28482](https://doi.org/10.1609/AAAI.V38I7.28482).
- [5] Y. Xie, X. Chen, L. Sun, and Y. Lu, “DG-Font: Deformable Generative Networks for Unsupervised Font Generation,” in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, Computer Vision Foundation / IEEE, 2021, pp. 5130–5140. doi: [10.1109/CVPR46437.2021.00509](https://doi.org/10.1109/CVPR46437.2021.00509).

- [6] C. Wang, M. Zhou, T. Ge, Y. Jiang, H. Bao, and W. Xu, “CF-Font: Content Fusion for Few-Shot Font Generation,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, IEEE, 2023, pp. 1858–1867. doi: [10.1109/CVPR52729.2023.00185](https://doi.org/10.1109/CVPR52729.2023.00185).
- [7] S. Park, S. Chun, J. Cha, B. Lee, and H. Shim, “Few-shot Font Generation with Localized Style Representations and Factorization,” in *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, AAAI Press, 2021, pp. 2393–2402. doi: [10.1609/AAAI.V35I3.16340](https://doi.org/10.1609/AAAI.V35I3.16340).
- [8] S. Park, S. Chun, J. Cha, B. Lee, and H. Shim, “Multiple Heads are Better than One: Few-shot Font Generation with Multiple Localized Experts,” in *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, IEEE, 2021, pp. 13880–13889. doi: [10.1109/ICCV48922.2021.01364](https://doi.org/10.1109/ICCV48922.2021.01364).
- [9] L. Tang *et al.*, “Few-Shot Font Generation by Learning Fine-Grained Local Styles,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, IEEE, 2022, pp. 7885–7894. doi: [10.1109/CVPR52688.2022.00774](https://doi.org/10.1109/CVPR52688.2022.00774).
- [10] Y. Kong *et al.*, “Look Closer to Supervise Better: One-Shot Font Generation via Component-Based Discriminator,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, IEEE, 2022, pp. 13472–13481. doi: [10.1109/CVPR52688.2022.01312](https://doi.org/10.1109/CVPR52688.2022.01312).
- [11] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-Image Translation with Conditional Adversarial Networks,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, IEEE Computer Society, 2017, pp. 5967–5976. doi: [10.1109/CVPR.2017.632](https://doi.org/10.1109/CVPR.2017.632).
- [12] M.-Y. Liu, T. M. Breuel, and J. Kautz, “Unsupervised Image-to-Image Translation Networks,” in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December*

4-9, 2017, Long Beach, CA, USA, 2017, pp. 700–708. [Online]. Available: <https://proceedings.neurips.cc/paper/2017/hash/dc6a6489640ca02b0d42dabeb8e46bb7-Abstract.html>

- [13] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks,” in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, IEEE Computer Society, 2017, pp. 2242–2251. doi: [10.1109/ICCV.2017.244](https://doi.org/10.1109/ICCV.2017.244).
- [14] M.-Y. Liu *et al.*, “Few-Shot Unsupervised Image-to-Image Translation,” in *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, IEEE, 2019, pp. 10550–10559. doi: [10.1109/ICCV.2019.01065](https://doi.org/10.1109/ICCV.2019.01065).
- [15] A. Zhu, X. Lu, X. Bai, S. Uchida, B. K. Iwana, and S. Xiong, “Few-Shot Text Style Transfer via Deep Feature Similarity,” *IEEE Trans. Image Process.*, vol. 29, pp. 6932–6946, 2020, doi: [10.1109/TIP.2020.2995062](https://doi.org/10.1109/TIP.2020.2995062).
- [16] C. Li, Y. Taniguchi, M. Lu, and S. Konomi, “Few-shot Font Style Transfer between Different Languages,” in *IEEE Winter Conference on Applications of Computer Vision, WACV 2021, Waikoloa, HI, USA, January 3-8, 2021*, IEEE, 2021, pp. 433–442. doi: [10.1109/WACV48630.2021.00048](https://doi.org/10.1109/WACV48630.2021.00048).
- [17] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium,” in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 2017, pp. 6626–6637. [Online]. Available: <https://proceedings.neurips.cc/paper/2017/hash/8a1d694707eb0fefe65871369074926d-Abstract.html>
- [18] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004, doi: [10.1109/TIP.2003.819861](https://doi.org/10.1109/TIP.2003.819861).
- [19] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, “The Unreasonable Effectiveness of Deep Features as a Perceptual Metric,” in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, July 18-22, 2018*, IEEE, 2018, pp. 5865–5874. doi: [10.1109/CVPR.2018.00620](https://doi.org/10.1109/CVPR.2018.00620).

*UT, USA, June 18-22, 2018*, Computer Vision Foundation / IEEE Computer Society, 2018, pp. 586–595. doi: [10.1109/CVPR.2018.00068](https://doi.org/10.1109/CVPR.2018.00068).

- [20] E. J. Hu *et al.*, “LoRA: Low-Rank Adaptation of Large Language Models,” in *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, OpenReview.net, 2022. [Online]. Available: <https://openreview.net/forum?id=nZeVKeeFYf9>
- [21] N. Houlsby *et al.*, “Parameter-Efficient Transfer Learning for NLP,” in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, in Proceedings of Machine Learning Research, vol. 97. PMLR, 2019, pp. 2790–2799. [Online]. Available: <http://proceedings.mlr.press/v97/houlsby19a.html>
- [22] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, “Least Squares Generative Adversarial Networks,” in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, IEEE Computer Society, 2017, pp. 2813–2821. doi: [10.1109/ICCV.2017.304](https://doi.org/10.1109/ICCV.2017.304).
- [23] GeeksforGeeks, “What are Diffusion Models?.” [Online]. Available: <https://www.geeksforgeeks.org/artificial-intelligence/what-are-diffusion-models/>
- [24] L. A. Gatys, A. S. Ecker, and M. Bethge, “A Neural Algorithm of Artistic Style,” *CoRR*, 2015, [Online]. Available: <http://arxiv.org/abs/1508.06576>
- [25] A. van den Oord, Y. Li, and O. Vinyals, “Representation Learning with Contrastive Predictive Coding,” *CoRR*, 2018, [Online]. Available: <http://arxiv.org/abs/1807.03748>
- [26] K. Tam, “Designing with the Hanzi Script.”
- [27] D. Sun, Q. Zhang, and J. Yang, “Pyramid Embedded Generative Adversarial Network for Automated Font Generation,” in *24th International Conference on Pattern Recognition, ICPR 2018, Beijing, China, August 20-24, 2018*, IEEE Computer Society, 2018, pp. 976–981. doi: [10.1109/ICPR.2018.8545701](https://doi.org/10.1109/ICPR.2018.8545701).
- [28] J. Ho and T. Salimans, “Classifier-Free Diffusion Guidance,” *CoRR*, 2022, doi: [10.48550/ARXIV.2207.12598](https://doi.org/10.48550/ARXIV.2207.12598).

- [29] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. [Online]. Available: <http://arxiv.org/abs/1409.1556>
- [30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” in *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, 2012, pp. 1106–1114. [Online]. Available: <https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html>

# Công trình khoa học

D. K. D. Tran and V. H. Duong, “CL-SCR: Decoupling Style and Structure for One-Shot Cross-Script Font Generation,” *The Journal of Supercomputing (under review)*, 2026.

# Phụ lục

## A. Chi tiết Kiến trúc mạng UNet

Mạng UNet đóng vai trò là bộ xương sống (backbone) trong mô hình khuếch tán, chịu trách nhiệm dự đoán nhiều tại từng bước thời gian. Cấu trúc chi tiết của mạng được trình bày tại [Bảng A.1](#), bao gồm các khối mã hoá (Encoder), giải mã (Decoder) và các mô-đun tích hợp đặc trưng như MCA và SI.

**Bảng A.1** — Chi tiết kiến trúc mạng UNet trong FontDiffuser. Trong đó: MCA là khối Tổng hợp nội dung đa quy mô, SI là khối Chèn phong cách (Style Insertion) sử dụng cơ chế Cross-Attention.

Loại Block	Số lượng	Kích thước đầu vào ( $C \times H \times W$ )	Kích thước đầu ra ( $C \times H \times W$ )
Conv block	1	$3 \times H \times W$	$64 \times H \times W$
Down block	2	$64 \times H \times W$	$64 \times \frac{H}{2} \times \frac{W}{2}$
MCA block	2	$64 \times \frac{H}{2} \times \frac{W}{2}$	$128 \times \frac{H}{4} \times \frac{W}{4}$
MCA block	2	$128 \times \frac{H}{4} \times \frac{W}{4}$	$256 \times \frac{H}{8} \times \frac{W}{8}$
Down block	2	$256 \times \frac{H}{8} \times \frac{W}{8}$	$512 \times \frac{H}{8} \times \frac{W}{8}$
MCA block	1	$512 \times \frac{H}{8} \times \frac{W}{8}$	$512 \times \frac{H}{8} \times \frac{W}{8}$
Up block	3	$512 \times \frac{H}{8} \times \frac{W}{8}$	$256 \times \frac{H}{4} \times \frac{W}{4}$
SI block	3	$256 \times \frac{H}{4} \times \frac{W}{4}$	$256 \times \frac{H}{2} \times \frac{W}{2}$
SI block	3	$256 \times \frac{H}{2} \times \frac{W}{2}$	$128 \times H \times W$
Up block	3	$128 \times H \times W$	$64 \times H \times W$
Conv block	1	$64 \times H \times W$	$3 \times H \times W$

## B. Chi tiết Kiến trúc mô-đun CL-SCR

Mô-đun CL-SCR được thiết kế dựa trên mạng VGG-19 pre-trained để trích xuất đặc trưng phong cách đa tầng, kết hợp với các lớp chiếu (Projector) để đưa về không gian vector phục vụ học tương phản. Cấu trúc chi tiết của mạng được trình bày tại [Bảng B.2](#).

**Bảng B.2** — Chi tiết kiến trúc và luồng dữ liệu của mô-đun CL-SCR. Các ký hiệu  $\text{ReLU}_1^x$  biểu thị lớp kích hoạt đầu tiên trong mỗi khối VGG.

Thành phần	Lớp / Thao tác	Kích thước đầu ra (Batch $\times$ C $\times$ H $\times$ W)
<b>Input Processing</b>	Input Image ( $x$ )	$B \times 3 \times 64 \times 64$
	Data Augmentation (RandomResizedCrop + Normalize)	$B \times 3 \times 64 \times 64$
<b>Style Extractor</b> (Backbone: VGG-19)	Block 1 ( $\text{ReLU}_1^1$ )	$B \times 64 \times 64 \times 64$
	Block 2 ( $\text{ReLU}_1^2$ )	$B \times 128 \times 32 \times 32$
	Block 3 ( $\text{ReLU}_1^3$ )	$B \times 256 \times 16 \times 16$
	Block 4 ( $\text{ReLU}_1^4$ )	$B \times 512 \times 8 \times 8$
	Block 5 ( $\text{ReLU}_1^5$ )	$B \times 512 \times 4 \times 4$
<b>Style Projector</b> (Shared weights)	Fusion (Avg+Max Pool $\rightarrow$ Conv1x1)	$B \times C_{\text{reduced}}$
	MLP Layer 1 (Linear + ReLU)	$B \times 1024$
	MLP Layer 2 (Linear + ReLU)	$B \times 2048$
	Output Layer (Linear + Normalize)	$B \times 2048$ (Style Vector $v$ )

<b>Thành phần</b>	<b>Lớp / Thao tác</b>	<b>Kích thước đầu ra (Batch × C × H × W)</b>
<b>Contrastive Loss</b> (InfoNCE)	Dynamic Sampling (Intra/ Cross)	$K = 4$ mẫu âm / step
	Loss Computation	Scalar ( $\mathcal{L}_{\text{CL-SCR}}$ )

## C. Các siêu tham số Tiên huấn luyện CL-SCR

Bảng C.3 dưới đây tóm tắt các thiết lập thực nghiệm chính xác được sử dụng cho giai đoạn tiền huấn luyện mô-đun CL-SCR.

**Bảng C.3** — Bảng tổng hợp các siêu tham số cho giai đoạn tiền huấn luyện CL-SCR.

Giai đoạn	Tham số	Giá trị
Tiền huấn luyện SCR (Pre-training)	Kích thước Batch (Batch Size)	16
	Tổng số bước lặp (Max Steps)	200,000
	Tốc độ học (Learning Rate)	$1 \times 10^{-4}$
	Nhiệt độ InfoNCE ( $\tau$ )	0.07
	Bộ tối ưu hoá (Optimizer)	Adam
	Kích thước ảnh (Resolution)	$64 \times 64$
	Augmentation	RandomResizedCrop (scale 0.8-1.0)
	Chế độ Loss	both $(\alpha = 0.3, \beta = 0.7)$
	Các lớp trích xuất (NCE Layers)	0, 1, 2, 3, 4, 5

## D. Các siêu tham số huấn luyện

Bảng D.4 dưới đây tóm tắt các thiết lập thực nghiệm chính xác được sử dụng trong mã nguồn huấn luyện (`train.py` và các script `.sh`).

**Bảng D.4** — Bảng tổng hợp các siêu tham số huấn luyện cho cả hai giai đoạn.

Giai đoạn	Tham số	Giá trị
<b>Giai đoạn 1:</b> <b>Tái tạo cấu trúc</b>	Độ phân giải ảnh (Resolution)	$64 \times 64$
	Kích thước Batch (Batch Size)	4
	Tổng số bước lặp (Max Steps)	400,000
	Tốc độ học (Learning Rate)	$1 \times 10^{-4}$ (Linear Decay)
	Số bước khởi động (Warmup)	10,000
	Trọng số Loss ( $\lambda_{\text{percep}}, \lambda_{\text{offset}}$ )	0.01, 0.5
	Bộ tối ưu hoá (Optimizer)	AdamW ( $\beta_1 = 0.9, \beta_2 = 0.999$ )
	Phần cứng	1 × NVIDIA Tesla P100
<b>Giai đoạn 2:</b> <b>Tinh chỉnh phong cách</b> (w/ CL-SCR)	Kích thước Batch (Batch Size)	4
	Tổng số bước lặp (Max Steps)	30,000
	Tốc độ học (Learning Rate)	$1 \times 10^{-5}$ (Constant)

Giai đoạn	Tham số	Giá trị
	Số bước khởi động (Warmup)	1,000
	Số lượng mẫu âm ( $K$ )	4
	Chế độ SCR ( <code>scr_mode</code> )	both (Intra + Cross)
	Trọng số Loss CL-SCR ( $\lambda_{sc}$ )	0.01
	Augmentation (SCR Input)	RandomResizedCrop (scale 0.8-1.0)

## E. Các tham số quá trình suy luận

Bảng E.5 dưới đây liệt kê chi tiết các thiết lập được sử dụng trong mã nguồn thực nghiệm (`new_inference.py` và `sample.py`) để đánh giá mô hình.

**Bảng E.5** — Bảng các tham số cấu hình cho quá trình suy luận (Inference).

Phân loại	Tham số	Giá trị thiết lập
Cấu hình Lấy mẫu (Sampling Config)	Thuật toán (Algorithm Type)	<code>dpm_solver++</code>
	Loại dự đoán (Model Prediction)	<code>noise</code> (dự đoán nhiễu $\varepsilon$ )
	Số bước suy luận (Inference Steps)	20 steps
	Bậc bộ giải (Solver Order)	2
	Chế độ hướng dẫn (Guidance Type)	<code>classifier-free</code>
	Trọng số hướng dẫn (Guidance Scale)	7.5
	Độ phân giải ảnh (Resolution)	$64 \times 64$

## F. Chi phí Tính toán và Thời gian

Do đặc thù của kiến trúc khuếch tán (Diffusion Models), phương pháp đề xuất có sự khác biệt rõ rệt về tài nguyên tiêu thụ so với các phương pháp GAN hay CNN truyền thống. Phần này bóc tách chi tiết thời gian huấn luyện và suy diễn.

### F.1. Thời gian Huấn luyện

Việc huấn luyện mô hình đề xuất (Ours) là một quy trình đa giai đoạn, đòi hỏi tài nguyên tính toán đáng kể để đảm bảo sự hội tụ của cả cấu trúc và phong cách.

#### a) Chi tiết các giai đoạn huấn luyện của phương pháp đề xuất (Ours Breakdown):

**Bảng F.6** dưới đây liệt kê thời gian tiêu tốn cho từng thành phần riêng biệt khi huấn luyện trên cấu hình phần cứng tham chiếu (01 GPU NVIDIA Tesla P100).

**Bảng F.6** — Thời gian huấn luyện cho từng giai đoạn của phương pháp đề xuất (Ours).

Thành phần	Mô tả	Số bước lặp	Batch Size	Thời gian ước tính
Pre-train SCR	Huấn luyện bộ trích xuất phong cách CL-SCR	200,000	16	≈ 80 giờ
Phase 1	Giai đoạn Tái tạo (Reconstruction)	400,000	4	≈ 24 giờ
Phase 2	Giai đoạn Tinh chỉnh (Refinement)	30,000	4	≈ 12 giờ
Tổng thời gian huấn luyện toàn bộ Pipeline				≈ 5 ngày

**b) So sánh tổng thời gian huấn luyện với các Baseline:**

So với các phương pháp hiện có, FontDiffuser yêu cầu thời gian huấn luyện dài hơn do bản chất hội tụ chậm của quá trình khử nhiễu và yêu cầu số bước lặp lớn.

**Bảng F.7** — So sánh tổng thời gian huấn luyện giữa phương pháp đề xuất và các Baseline.

Mô hình	Cơ chế lõi	Tổng thời gian Huấn luyện (Ước tính trên GPU đơn)
DG-Font [5]	Unsupervised GAN (Deformable)	Trung bình (≈ 1 - 2 ngày)
CF-Font [6]	GAN (Content Fusion)	Trung bình (≈ 1 - 2 ngày)
DFS [15]	cGAN (Feature Matching)	Trung bình (≈ 20 - 24 giờ)
FTransGAN [16]	GAN (Multi-level Attention)	Trung bình (≈ 20 - 24 giờ)
<b>Ours (FontDiffuser)</b>	<b>Diffusion (Denoising)</b>	<b>Lâu</b> (≈ 5 ngày)

## F.2. Tốc độ Suy diễn

Trong giai đoạn triển khai (Inference), tốc độ sinh ảnh là yếu tố quan trọng đối với trải nghiệm người dùng.

Bảng dưới đây so sánh thời gian trung bình để sinh ra **một ký tự ảnh** (kích thước  $64 \times 64$ ). Phương pháp để xuất sử dụng bộ giải **DPM-Solver++** với 20 bước lấy mẫu.

**Bảng F.8 — So sánh tốc độ suy diễn.**

Mô hình	Cơ chế sinh	Số bước chuyển tiếp (Forward Passes)	Thời gian / 1 ảnh
DG-Font [5]	Feed-forward (One-step)	1	Rất nhanh ( $< 0.05s$ )
CF-Font [6]	Feed-forward (One-step)	1	Rất nhanh ( $< 0.05s$ )
DFS [15]	Feed-forward (One-step)	1	Rất nhanh ( $< 0.05s$ )
FTransGAN [16]	Feed-forward (One-step)	1	Nhanh ( $\approx 0.1s$ )
<b>Ours (FontDiffuser)</b>	<b>Iterative Denoising</b>	<b>20</b>	<b>Chậm</b> ( $\approx 2.0 - 3.0s$ )