ĐẠI HỌC QUỐC GIA HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

Trần Đình Khánh Đăng - 22520195

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

KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

Khoa: Khoa học máy tính

HỒ CHÍ MINH - 2025

ĐẠI HỌC QUỐC GIA HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

Trần Đình Khánh Đăng - 22520195

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

KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

Khoa: Khoa học máy tính

Giảng viên hướng dẫn: TS. Dương Việt Hằng

HỒ CHÍ MINH - 2025

LỜI CAM ĐOAN

Em xin cam đoan: Khoá luận tốt nghiệp với đề tài "Tăng cường khả năng chuyển kiểu chữ đa ngôn ngữ trongbài toán one-shot bằng mô hình khuếch tán" trong báo cáo này là do em thực hiện dưới sự hướng dẫn của Tiến Sĩ Dương Việt Hằng. Những gì em viết ra hoàn toàn trung thực, chính xác và không có sự sao chép từ các tài liệu, không sử dụng kết quả của người khác mà không trích dẫn cụ thể. Đây là công trình nghiên cứu cá nhân em tự phát triển, không sao chép mã nguồn của người khác. Nếu vi phạm những điều trên, em xin chấp nhận tất cả những truy cứu về trách nhiệm theo quy định của Trường Đại học Công nghệ Thông tin — ĐHQGHCM.

Hồ Chí Minh, ngày 10 tháng 12 năm 2025

Sinh viên

Trần Đình Khánh Đăng

LỜI CẢM ƠN

Lời đầu tiên, cho phép em bày tỏ lòng biết ơn sâu sắc đến Quý thầy/cô ở Khoa Khoa học máy tính và Trường Đại học Công nghệ Thông tin — ĐHQGHCM. Đây là nơi em đã có cơ hội tiếp cận với những tri thức mới mẻ, được học hỏi từ các thầy cô xuất sắc và kết nối với những người bạn, anh chị em đầy năng động và tài năng.

Em cũng xin gửi lời cảm ơn chân thành nhất đến cô Dương Việt Hằng, người đã luôn là nguồn cảm hứng và sự hướng dẫn quý báu trong suốt thời gian em học tập tại trường. Sự tận tâm và hỗ trợ nhiệt tình của cô đã tiếp thêm động lực để em vượt qua những thử thách trong hành trình nghiên cứu và hoàn thiện khóa luận tốt nghiệp.

Ngoài ra, em xin gửi lời cảm ơn đến gia đình, bạn bè và những người đã luôn giúp đỡ, động viên, đồng hành cùng em suốt chặng đường học tập ở trường và khoảng thời gian thực hiện khóa luận.

Kính chúc tất cả mọi người luôn vui vẻ, hạnh phúc và gặt hái được nhiều thành công trong cuộc sống.

Một lần nữa, xin chân thành cảm ơn tất cả những tấm lòng đã đồng hành cùng em suốt chặng đường qua!

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 kí 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 kí 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 Regularization – 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 hóa khi áp dụng phong cách từ hệ chữ này sang hệ chữ khác.

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

MỤC LỤC

DANH MỤC HÌNH ẢNH

DANH MỤC BẢNG

DANH MỤC GIẢI THUẬT

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 kĩ 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 hóa 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) sang Denoising Diffusion Probabilistic Models (DDPMs) gần đây đã tạo ra bước đột phá về chất lượng ảnh sinh. Các mô hình Diffusion, điển hình như FontDiffuser, đã chứng minh khả năng vượt trội trong việc tái tạo các chi tiết nét chữ phức tạp và duy trì cấu trúc tô 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.

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. 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, tuyến 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).

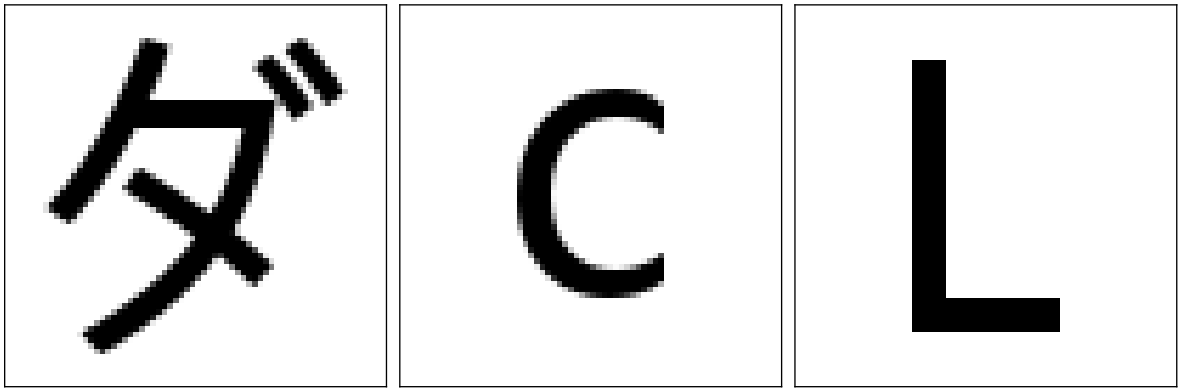
1.2. Mô tả bài toán

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) có điều kiện.

1.2.1. 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) trong một phông chữ tiêu chuẩn (ví dụ: Arial hoặc Noto Sans).
  - Mục đích: Cung cấp thông tin về cấu trúc hình học và định danh của ký tự cần sinh (ví dụ: chữ 'A', chữ 'g').
  - Trong bài toán cross-lingual, I_c thuộc hệ ngôn ngữ đích (Target Language, ví dụ: Latin).



Ví dụ về một số ảnh Content

- 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.
  - Mục đích: Cung cấp các đặc trưng thẩm mỹ (nét xước, độ đậm nhạt, serif…).
  - Trong bài toán cross-lingual, I_s thường thuộc hệ ngôn ngữ nguồn (Source Language, ví dụ: Chinese) khác với ngôn ngữ của I_c.

1.2.2. Định nghĩa đầu ra (Output)

Ảnh được sinh ra (Generated Image - I_{gen}):

- Là hình ảnh kết quả thể hiện ký tự c nhưng mang phong cách s.
- Yêu cầu: I_{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).

1.2.3. Mục tiêu toán học

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

$$I_{\text{gen}} = G(I_c, I_s) \tag{1}$$

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

1.3. Mục tiêu của đề tài

Khoá luận này đề xuất mở rộng mô hình FontDiffuser để giải quyết bài toán **cross-lingual font generation**, cụ thể:

- Thiết kế pipeline cho phép trích xuất phong cách từ một ngôn ngữ và áp dụng lên glyph của ngôn ngữ khác.
- Đề xuất cơ chế **Style-Content Regularization (SCR) mở rộng** với cả positive/negative pair cross-lingual nhằm buộc mô hình học đặc trưng phong cách bất biến theo ngôn ngữ.
- Tích hợp quá trình huấn luyện lại (fine-tuning) mô hình diffusion với dữ liệu đa ngôn ngữ và đánh giá chất lượng đầu ra theo các thước đo định lượng (LPIPS, FID) và đánh giá trực quan.

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

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: Các mô hình sinh ảnh dựa trên khuếch tán (Diffusion Models), trọng tâm là kiến trúc FontDiffuser và các kỹ thuật điều hướng phong cách (Style Guidance).

Đối tượng dữ liệu: Hệ chữ nguồn (Source): Các bộ phông chữ Hán (Chinese Fonts) đa dạng về phong cách (Mincho, Gothic, Thư pháp…). Hệ chữ đích (Target): 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 chỉ giới hạn việc nghiên cứu và thực nghiệm trên quá trình chuyển đổi phong cách từ Tiếng Anh (Latin) sang Tiếng Trung Quốc (Hán). Lý do là vì độ phức tạp của chữ Hán thường cao hơn đáng kể so với chữ Latin: số

nét nhiều, cấu trúc không tuyến tính, và chứa các thành phần hình thái mà chữ Latin không có. Điều này khiến quá trình chuyển giao phong cách trở nên thách thức hơn, buộc mô hình phải linh hoạt, tổng quát hóa tốt và giữ được tính ổn định khi tái tạo phong cách từ một hệ chữ đơn giản sang một hệ chữ phức tạp hơn. Vì vậy, việc lựa chọn cặp ngôn ngữ này giúp đánh giá rõ ràng hơn khả năng thích ứng và độ mạnh của mô hình trong các tình huống chuyển đổi phong cách có mức độ khó cao.

Phạm vi về bài toán: Tập trung vào bài toán One-shot Generation: Đối với chữ Latin, mô hình được yêu cầu sinh toàn bộ bảng chữ cái tiếng Anh chỉ từ một ký tự đầu vào duy nhất. Tuy nhiên, khi mở rộng sang chữ Hán, số lượng ký tự là quá lớn để thực hiện cùng một yêu cầu. Do đó, nghiên cứu chỉ hướng tới việc sử dụng phong cách của một ký tự Latin làm "style input" và áp dụng nó lên một tập con ký tự Hán đóng vai trò "content input", nhằm đánh giá khả năng chuyển giao phong cách trong bối cảnh hệ chữ mục tiêu phức tạp và đồ sộ hơn rất nhiều.

Phạm vi về dữ liệu: Sử dụng các bộ dữ liệu phông chữ mã nguồn mở hỗ trợ đồng thời cả hai bảng mã Unicode cho tiếng Trung và tiếng Anh (ví dụ: Google Noto CJK, Adobe Source Han Serif) để đảm bảo có cặp dữ liệu đối chứng (Ground Truth) chính xác cho quá trình huấn luyện và đánh giá.

1.5. Cấu trúc của khoá luận

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

- Chương 2 – Tổng Quan Lý Thuyết và Các Nghiên Cứu Liên Quan:

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 (DG-Font, CF-Font, DFS, GAS-NeXt) và nhóm mô hình diffusion (FontDiffuser), chỉ ra ưu nhược điểm và xu hướng phát triển.

- Chương 3 – Phương Pháp Đề Xuất:

Trình bày chi tiết pipeline gốc của FontDiffuser, bao gồm hai giai đoạn huấn luyện (Phase 1 – Reconstruction, Phase 2 – Style Refinement). Phân tích cơ chế hoạt động của các module 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 module SCR.

- Chương 4 – Thực Nghiệm, Kết Quả và Phân Tích

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, SSIM, LPIPS, L1, User Study). Đồng thời, chương trình bày các kết quả định lượng và định tính, 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 module 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 khóa 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 hoặc Adapter) để tối ưu tài nguyên huấn luyện.

Chương 2

Cơ sở Lý thuyết và Tổng quan Tài liệu

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, đến sự trỗi dậy của Mô hình khuếch tán (Diffusion Models). 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. Tổng quan về các phương pháp Sinh phông chữ

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) và Mô hình khuếch tán (Diffusion Models).

2.1.1. Các phương pháp dựa trên GAN (GAN-based Approaches)

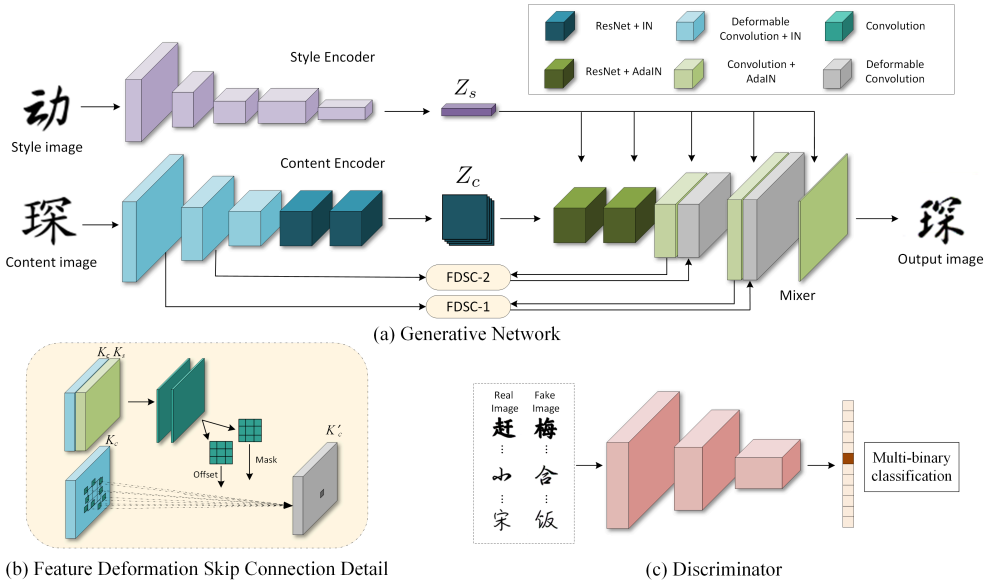
Trước sự bùng nổ của Diffusion Models vào năm 2023, Generative Adversarial Networks (GAN) là hướng tiếp cận chủ đạo (State-of-the-art) cho bài toán này. Các nghiên cứu GAN thường tập trung giải quyết vấn đề tách biệt nội dung (content) và phong cách (style).

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

DG-Font tiếp cận bài toán theo hướng học không giám sát (unsupervised), tập trung giải quyết thách thức về sự sai lệch hình học giữa font nguồn và font đích. Thay

vì cố gắng ép buộc mô hình học phong cách ngay lập tức, DG-Font giới thiệu mô-đun **Feature Deformation Skip Connection (FDSC)**.

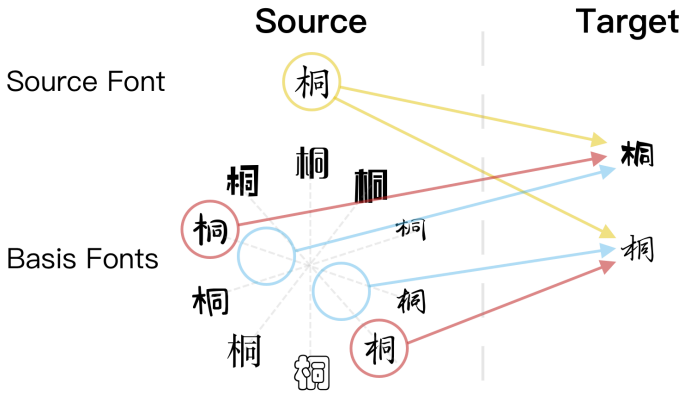
Cơ chế này hoạt động bằng cách dự đoán các bản đồ dịch chuyển (displacement maps) và áp dụng tích chập biến dạng (deformable convolution) lên các đặc trưng cấp thấp. Điều này cho phép mô hình "uốn nắn" cấu trúc của ký tự nguồn sao cho khớp với dáng vẻ của ký tự đích. Tuy nhiên, điểm yếu cố hữu của DG-Font nói riêng và GAN nói chung là sự mất ổn định trong quá trình huấn luyện (training instability). Khi gặp các ký tự có cấu trúc quá phức tạp hoặc khác biệt lớn về topo học (ví dụ từ chữ in sang chữ thư pháp), mô hình thường tạo ra các kết quả bị mờ hoặc đứt nét (broken strokes).



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.

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

CF-Font đề xuất giải quyết vấn đề bằng cách "lai ghép" nội dung. Thay vì tin tưởng hoàn toàn vào một ảnh nguồn, CF-Font sử dụng mô-đun **Content Fusion Module (CFM)** để nội suy đặc trưng từ một tập hợp các font cơ sở (basis fonts). Phương pháp này giúp giảm thiểu việc mất mát thông tin cấu trúc, nhưng lại dễ gây ra hiện tượng "bóng ma" (ghosting artifacts) khi các font cơ sở không đủ đa dạng hoặc quá khác biệt so với font đích.

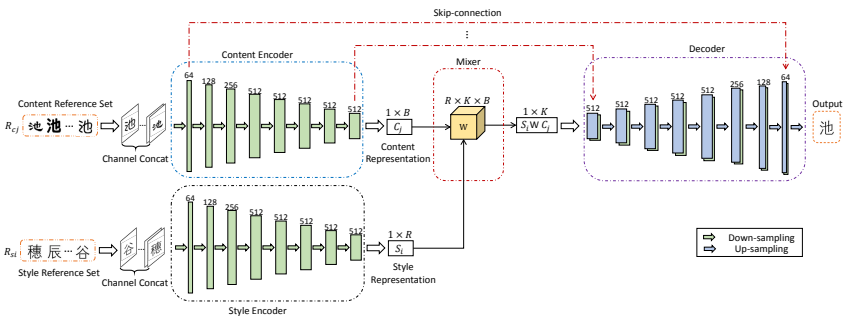


Hình 2.2 — Minh hoạ cơ chế Content Fusion: Sự kết hợp tuyến tính các đặc trưng nội dung giúp xấp xỉ font mục tiêu tốt hơn.

2.1.1.3. EMD (Separating Style and Content for Generalized Style Transfer, CVPR 2018)

EMD giải quyết bài toán chuyển kiểu bằng cách tách rời hoàn toàn hai thành phần style và content. Hai encoder độc lập được huấn luyện để rút trích các đặc trưng style/ content "thuần" từ những tập ảnh tham chiếu nhỏ, trong đó các ảnh được ghép theo chiều kênh để làm nổi bật tính chất chung của từng yếu tố. Hai đặc trưng này được kết hợp qua mô-đun Mixer dùng **bilinear model**, cho phép tái tổ hợp linh hoạt style và content, từ đó sinh ra ký tự mang style mới mà không cần huấn luyện lại mô hình.

Decoder đối xứng, kết hợp skip-connection từ Content Encoder, giúp khôi phục hình dạng ký tự chính xác ngay cả với các nội dung hoàn toàn mới. Nhờ kiến trúc phân tách, EMD chỉ cần rất ít ảnh tham chiếu (5–10 hình) để tái tạo trọn bộ font và có khả năng tổng quát hóa tốt hơn các phương pháp dựa trên GAN. Tuy nhiên, do không dùng adversarial loss, kết quả của EMD thường sạch và đúng cấu trúc nhưng có thể thiếu độ sắc nét hoặc chi tiết thị giác cao.



Hình 2.3 — Kiến trúc mạng EMD.

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

DFS tiếp cận bài toán chuyển kiểu chữ theo hướng few-shot, kết hợp đồng thời cả kiểu font (hình học) lẫn texture (màu sắc, hiệu ứng). Thay vì ép mô hình học trực tiếp từ tập tham chiếu nhỏ, DFS khai thác đặc trưng sâu từ hai mạng CNN độc lập: một cho content và một cho style. Các đặc trưng style của từng ký tự tham chiếu được trích xuất riêng rẽ, sau đó được **trọng số hóa theo mức độ tương đồng hình dạng** giữa từng ký tự tham chiếu và ký tự mục tiêu. Trọng số này được tính trong không gian đặc trưng thông qua normalized cross-correlation, tạo thành **Similarity Matrix** – thành phần trung tâm cho phép mô hình "ưu tiên" các ký tự tham chiếu giống nhất về cấu trúc.

Các đặc trưng style đã được điều chỉnh sau đó được gộp lại và nối với đặc trưng content, rồi đưa qua decoder đối 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 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 túy.

DFS có khả năng tổng quát hóa tốt trong thiết lập few-shot, cho phép sinh trọn bộ ký tự chỉ từ 4–8 mẫu tham chiếu. Việc tách đặc trưng từng tham chiếu giúp mô hình linh hoạt về số lượng và thứ tự đầu vào, đồng thời thích ứng tốt với nhiều ngôn ngữ (Latin, Chinese). Tuy vậy, DFS vẫn phụ thuộc mạnh vào độ đa dạng ký tự tham chiếu: nếu chỉ cung cấp các ký tự ít nét hoặc thiếu cấu trúc then chốt, mô hình thường thất bại trong việc tái tạo các ký tự có hình dạng phức tạp (vòng cung, giao nét). Ngoài ra, DFS cần fine-tune theo từng style mới, nên khó áp dụng khi số lượng mẫu tham chiếu quá ít (ví dụ chỉ một ký tự).

2.1.2. Mô hình khuếch tán (Diffusion Models)

Gần đây, Mô hình khuếch tán (Diffusion Models) đã tạo nên một cuộc cách mạng trong lĩnh vực thị giác máy tính. Khác với GAN – 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.

Nguyên lí cơ bản gồm hai giai đoạn:

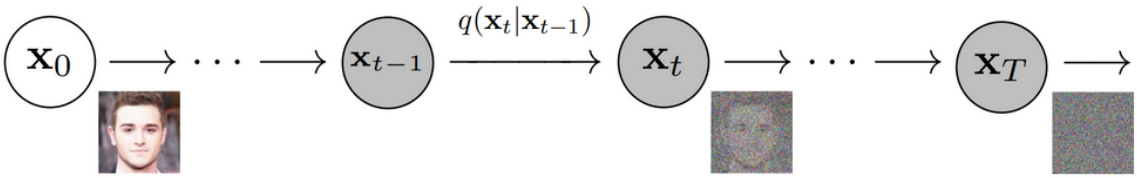
- **Quá trình Khuếch tán xuôi:** phá hủy dữ liệu một cách có kiểm soát bằng cách thêm nhiễu Gaussian nhiều bước.

- **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.

2.1.2.0.1. Quá trình Khuếch tán xuôi

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 làm giảm nhẹ dữ liệu bằng cách thêm nhiễu Gauss:



Hình 2.4 — Quá trình Khuếch tán xuôi

Về mặt toán học, có thể được biểu diễn như sau:

$$q(x_t|x_{t-1}) = \mathcal{N}\left(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I\right) \tag{2}$$

- x_0: ảnh gốc (clean image).
- x_t: ảnh ở bước t sau khi thêm nhiễu.
- β_t: hệ số nhiễu nhỏ (thường $\beta_t \in [10^{-4}, 0.02]$).
- I: ma trận đơn vị, đảm bảo nhiễu độc lập và đẳng hướng.

Do tính chất của Gaussian, ta có thể suy ra trực tiếp từ x_0 đến x_t:

$$x_t = \sqrt{\alpha_t}x_0 + \sqrt{1-\alpha_t}\epsilon, \epsilon \sim \mathcal{N}(0, I) \tag{3}$$

trong đó:

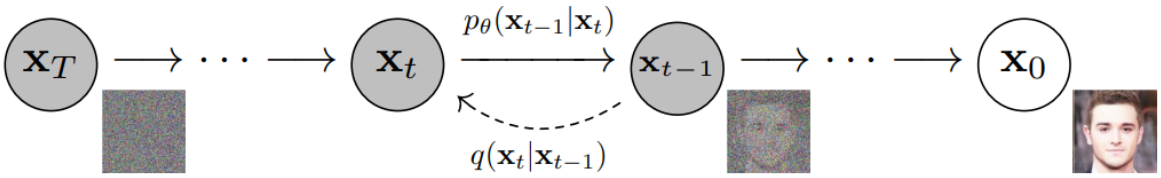
$$\alpha_t = 1 - \beta_t \tag{4}$$

$$\alpha_t = \prod_{s=1}^{t} \alpha_s \tag{5}$$

Điều này rất quan trọng vì giúp ta không cần sinh tuần tự từng bước mà vẫn có thể lấy mẫu trực tiếp ở bước t bất kì (quan trọng khi huấn luyện batch lớn).

2.1.2.0.2. Quá trình Khuếch tán ngược

Quá trình này nhằm mục đích tái tạo lại dữ liệu gốc bằng cách khử nhiễu bằng một loạt các bước đảo ngược quá trình khuếch tán xuôi.



Hình 2.5 — Quá trình Khuếch tán ngược

Về mặt toán học, có thể được biểu diễn như sau:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}\left(x_{t-1}; \mu_\theta(x_t, t), \sum_\theta(x_t, t)\right) \tag{6}$$

với μ_θ được tính như sau:

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\alpha_t}}\epsilon_\theta(x_t, t)\right) \tag{7}$$

Ở đây, $\epsilon_\theta(x_t, t)$ là nhiễu do mạng nơ-ron dự đoán, đóng vai trò trung tâm trong việc phục hồi ảnh gốc.

Trong huấn luyện, mô hình được tối ưu để giảm sai số giữa $\epsilon_\theta(x_t, t)$ và nhiễu thực ϵ mà ta đã thêm ở forward process.

2.1.2.0.3. Loss function

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

$$\mathscr{L}_{\text{simple}} = \mathbb{E}_{t,x_0,\epsilon}[\parallel \epsilon - \epsilon_\theta(x_t, t) \parallel^2] \tag{8}$$

Điều này tương đương với việc tối đa hóa khả năng tái tạo phân phối dữ liệu gốc (variational lower bound). Các nghiên cứu gần đây (v-prediction, hybrid loss) cho thấy việc dự đoán trực tiếp v_t hoặc x_0 có thể cải thiện chất lượng ảnh sinh, nhưng MSE vẫn là chuẩn mực trong nhiều ứng dụng như FontDiffuser.

2.1.2.1. FontDiffuser (AAAI 2024)

FontDiffuser 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. Lý thuyết về 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.

2.2.1. Neural Style Transfer truyền thống

Các phương pháp sơ khai (như Gatys et al.) 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 họa tiết cục bộ. Đối 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) 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.

- Các mẫu khác phong cách (Negative samples) bị đẩy ra xa nhau.

Hàm mất mát InfoNCE thường được sử dụng để tối ưu hóa không gian này:

$$\mathscr{L}_{\text{NCE}} = -\log\left(\frac{\exp(\text{sim}(z, z^+)/\tau)}{\exp(\text{sim}(z, z^+)/\tau) + \sum_k \exp(\text{sim}(z, z_k^-)/\tau)}\right) \tag{9}$$

Trong FontDiffuser, mô-đun SCR áp dụng tư tưởng này để giám sát bộ mã hóa phong cách. Tuy nhiên, module này ban đầu được thiết kế cho cùng một ngôn ngữ (Hán $\rightarrow$ Hán). Khi áp dụng sang bài toán Cross-Lingual, đặc biệt là dùng chữ Latin làm mẫu phong cách, các phương pháp chọn mẫu âm (negative selection) thông thường trở nên kém hiệu quả do khoảng cách miền (domain gap) quá lớn giữa hai hệ chữ.

2.3. Thách thức trong bài toán Cross-Lingual: Từ Latin sang Hán tự

Khác với các hướng tiếp cận thông thường (Hán $\rightarrow$ Hán hoặc Hán $\rightarrow$ Latin), khoá luận này tập trung vào bài toán thách thức hơn: **Sử dụng ảnh phong cách Latin (Simple) để sinh ảnh nội dung Hán tự (Complex).**

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

Đây là thách thức lớn nhất của hướng nghiên cứu này.

- **Ảnh phong cách (Latin):** Có cấu trúc đơn giản, ít nét, mật độ thông tin thấp. Ví dụ: chữ 'I' chỉ là một nét sổ, chữ 'O' là một vòng tròn.
- **Ảnh nội dung (Hán tự):** Có cấu trúc cực kỳ phức tạp, mật độ nét cao (trung bình 10-15 nét, cá biệt lên tới 30 nét), không gian bố cục chật hẹp.

Bài toán đặt ra là một dạng **"Ngoại suy phong cách" (Style Extrapolation)**: Mô hình phải học cách "tưởng tượng" xem một phong cách đơn giản (ví dụ: nét thanh đậm của chữ 'A') sẽ trông như thế nào khi áp dụng lên một cấu trúc chằng chịt như chữ '龍' (Long - Rồng). Nếu không xử lý tốt, mô hình rất dễ sinh ra các nét dính bết vào nhau (blob) hoặc làm mất đi các chi tiết phong cách khi cố gắng nhồi nhét vào cấu trúc phức tạp.

2.3.2. Khoảng cách Hình thái học (Morphological Gap)

Sự khác biệt về quy tắc viết (stroke order) và cấu tạo (topology) giữa hai hệ chữ tạo ra rào cản lớn cho việc chuyển giao phong cách:

1. **Cấu trúc:** Latin là hệ chữ tuyến tính, độ rộng biến thiên. Hán tự là hệ chữ khối (block-based), kích thước cố định.
2. **Đặc trưng cục bộ:** Các chi tiết phong cách đặc trưng của Latin (như serifs ở chân chữ, terminal ở đầu chữ) không có sự tương quan trực tiếp 1-1 với các bộ thủ trong tiếng Trung.

Do đó, việc áp dụng trực tiếp module SCR (Style Contrastive Refinement) nguyên bản là không đủ, vì nó không được huấn luyện để xử lý sự chênh lệch độ phức tạp này. Khoá luận này sẽ đề xuất cải tiến SCR nhằm giúp mô hình học được các đặc trưng phong cách "bất biến" (invariant style features) từ chữ Latin và áp dụng chúng một cách thông minh lên cấu trúc Hán tự phức tạp.

Chương 3

Phương Pháp Đề Xuất

3.1. Giới thiệu chương

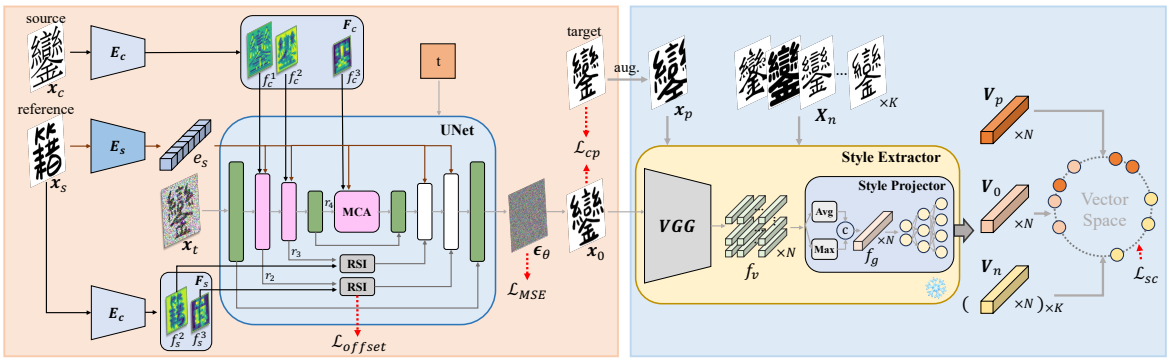
Trong chương trước, khoá luận đã phân tích các hạn chế của phương pháp GAN và tiềm năng của Mô hình khuếch tán (Diffusion Models) trong bài toán sinh phông chữ. Dựa trên cơ sở đó, chương này trình bày chi tiết phương pháp nghiên cứu được đề xuất.

Cụ thể, khoá luận kế thừa kiến trúc tiên tiến **FontDiffuser** (Yang et al., AAAI 2024) 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, phân tích cơ chế hoạt động của module SCR gốc, và cuối cùng là chi tiết về giải pháp CL-SCR được đề xuất cho bài toán đa ngôn ngữ.

3.2. 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 hóa bài toán sinh phông chữ dưới dạng quy trình "khử nhiễu" (noise-to-denoise).



(a) Conditional Diffusion for Font Generation (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)

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:

3.2.1. 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 nhiễu, dựa trên điều kiện x_c và x_s. Các thành phần cốt lõi bao gồm:

- **Bộ mã hóa nội dung (E_c) và phong cách (E_s):** Trích xuất đặc trưng ngữ nghĩa.

3.2.1.1. Multi-scale Content Aggregation (MCA):

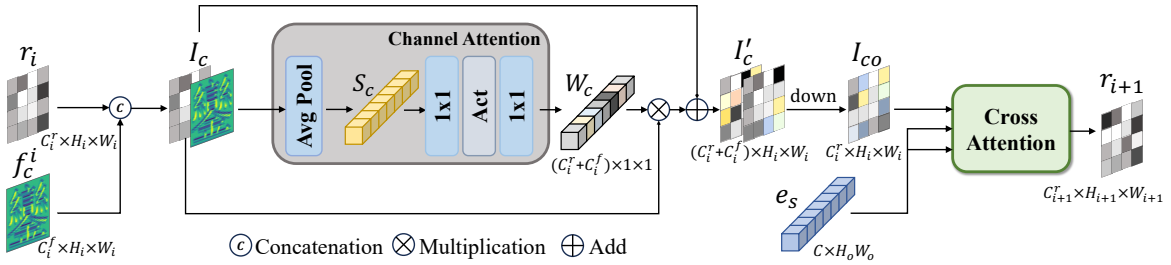
Đây là cơ chế tổng hợp đặc trưng đa tỉ lệ được thiết kế để giải quyết hạn chế của các phương pháp chỉ dựa vào một mức đặc trưng duy nhất. Khi sinh các ký tự phức tạp, một tầng đặc trưng đơn lẻ thường không thể đồng thời nắm bắt được cả bố cục tổng thể lẫn những chi tiết tinh vi như nét mảnh, bộ phận 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:

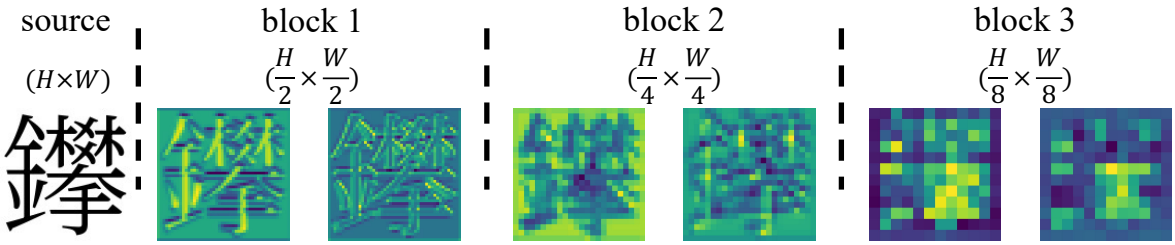
1. Ảnh tham chiếu x_c trước hết được nhúng bởi bộ mã hóa nội dung E_c để thu được các đặc trưng đa tỷ lệ $F_c = \{f_c^1, f_c^2, f_c^3\}$ từ các tầng khác nhau.
2. Mỗi đặc trưng nội dung f_c^i được đưa vào UNet thông qua ba khối MCA tương ứng. Tại đây, f_c^i được ghép nối (concatenated) với đặc trưng 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 1×1 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 hóa 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 × được sử dụng để giảm số lượng kênh, thu được đầu ra $I_{\{\%\}}$.
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 $I_{\{\%\}}$ 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 — Multi-scale Content Aggregation



Hình 3.6 — Content features in various blocks

3.2.1.2. 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) δ_{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_c 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 ϕ:
   - **Query (Q):** Được tạo ra từ đặc trưng tham chiếu $S_s\big(\phi_q(S_s)\big)$.
   - **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_{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 δ_{offset}.
5. Cuối cùng, DCN sử dụng độ lệch này để "uốn nắn" đặc trưng UNet, tạo ra đầu ra I_R đã được că chỉnh.

$$I_R = \text{DCN}(r_i, \delta_{\text{offset}}) \tag{10}$$

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.

3.2.2. Giai đoạn 2: Tinh chỉnh phong cách (Style Refinement Phase)

Mặc dù Giai đoạn 1 có thể tạo ra ký tự rõ nét, nhưng phong cách thường chưa được tách biệt hoàn toàn. Giai đoạn 2 cố định các trọng số của UNet và tập trung huấn luyện module **Style Contrastive Refinement (SCR)**. Module này đóng vai trò như một người hướng dẫn, sử dụng cơ chế học tương phản (Contrastive Learning) để ép buộc mô hình sinh ra ảnh có style vector gần với ảnh tham chiếu nhất có thể.

3.3. Phân tích Module Style Contrastive Refinement (SCR)

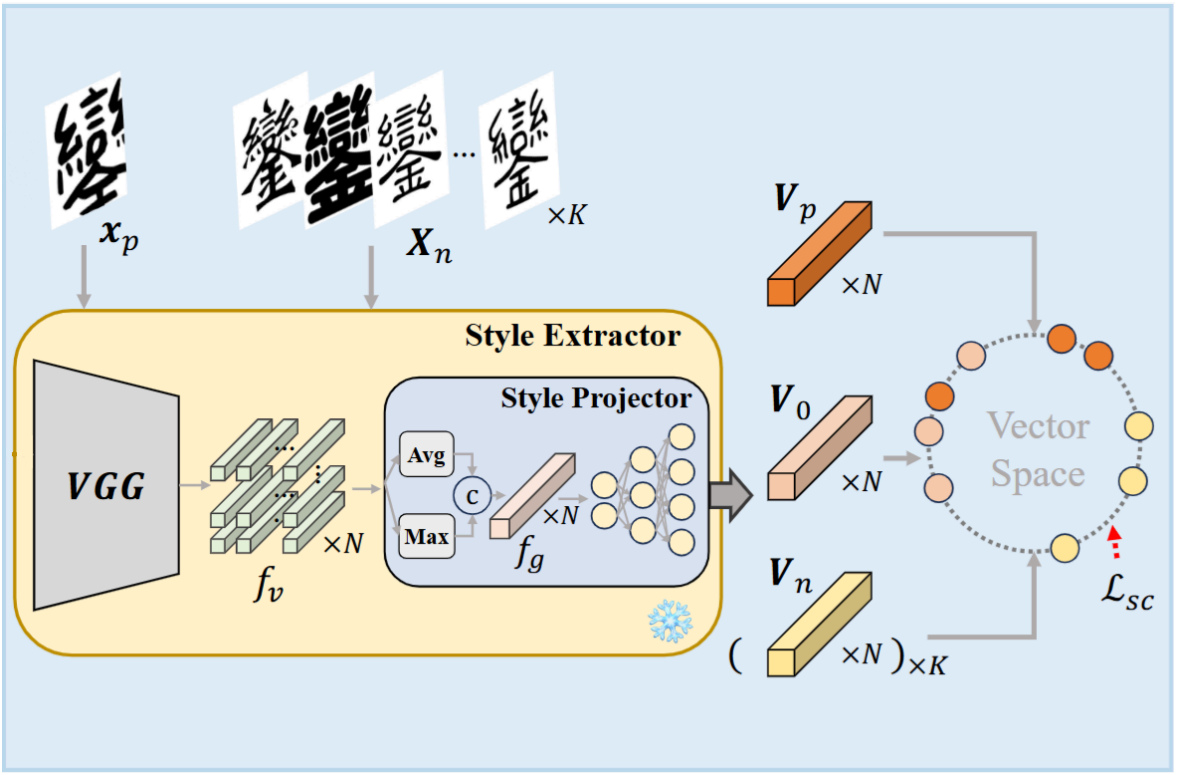
3.3.1. Động lực và Kiến trúc

Trong bài toán sinh phông chữ (font generation), mục tiêu cốt lõi của việc sinh phông chữ là đạt được hiệu ứng bắt chước phong cách (style imitation) chính xác, độc lập với sự biến thiên về phong cách giữa ảnh nguồn và ảnh tham chiếu. Trong các mô hình sinh ảnh truyền thống, sự vướng víu (disentanglement) giữa đặc trưng phong cách và nội dung thường không hoàn hảo, dẫn đến kết quả phong cách không nhất quán. Để giải quyết vấn đề này, nhóm tác giả đề xuất một chiến lược mới: xây dựng mô-đun **Style Contrastive Refinement (SCR).**

Mô-đun Style Contrastive Refinement (SCR) được đề xuất như một chiến lược mới để giải quyết vấn đề này. SCR hoạt động như một cơ chế học biểu diễn (representation learning module) và một bộ giám sát đặc trưng (feature supervisor). Nó không tham gia trực tiếp vào quá trình sinh ảnh pixel-wise của mô hình khuếch tán (diffusion model), mà có nhiệm vụ 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.

3.3.2. Kiến trúc Khai thác Phong cách (Style Extractor Architecture)

Kiến trúc của SCR, như được minh họa trong thiết kế hệ thống, bao gồm hai thành phần chính:



Hình 3.7 — Minh hoạ mô-đun SCR

1. **Bộ trích xuất Đặc trưng (Style Extractor):**
   - Sử dụng một mạng **VGG** (lấy cảm hứng từ Zhang et al. 2022) 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 $F_v = \{f_v^0, f_v^1, ..., f_v^N\}$.
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** $V = \{v^0, v^1, ..., v^N\}$. Các vector này đóng vai trò là đầu vào cho hàm mất mát tương phản.

3.3.3. Cơ chế Học Tương phản và Hàm Mất mát (Contrastive Learning and Loss Function)

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

3.3.3.1. Chiến lược Thiết lập Mẫu (Sampling Strategy)

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**.

3.3.3.2. 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** được tính tổng trên N tầng đặc trưng:

$$L_{\text{sc}} = -\sum_{l=0}^{N-1} \log \frac{\exp\left(v_0^l \cdot v_p^l \ / \ \tau\right)}{\left(\exp\left(v_0^l \cdot v_p^l \ / \ \tau\right) + \sum_{i=1}^{K} \exp\left(v_0^l \cdot v_{n_i}^l \ / \ \tau\right)\right.} \tag{11}$$

Trong đó:

- Extrac biểu thị bộ trích xuất phong cách: $V_0 = \text{Extrac}(x_0)$, $V_p = \text{Extrac}\left(x_p\right)$, $V_n = \text{Extrac}(x_n)$.
- $v_0^l, v_p^l, v_{n_i}^l$: lần lượt là vector lớp thứ l của ảnh sinh, ảnh dương và ảnh âm.
- $v_0^l \cdot v_p^l$: biểu thị độ tương đồng cosine (dot product) giữa hai vector phong cách.
- K: Số lượng mẫu âm.
- τ: siêu tham số nhiệt độ (temperature hyper-parameter), được thiết lập ở mức 0.07.

Thông 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.

3.4. Kết hợp vào Mục tiêu Huấn luyện (Training Objective)

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 Cơ bản (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^1_{total}) là sự kết hợp của ba thành phần:

$$L^1_{\text{total}} = L_{\text{MSE}} + \lambda^1_{\text{cp}} L_{\text{cp}} + \lambda^1_{\text{off}} L_{\text{offset}} \tag{12}$$

Chi tiết các thành phần:

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

$$L_{\text{MSE}} = \|\varepsilon - \varepsilon_{\theta(x_t, t, x_c, x_s)}\|^2 \tag{13}$$

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

$$L_{\text{cp}} = \sum_{l=1}^{L} \|\text{VGG}_l(x_0) - \text{VGG}_l\big(x_{\text{target}}\big)\| \tag{14}$$

- **Hàm mất mát Độ lệch (L_{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 δ_{offset} nhằm ngăn chặn các biến dạng cấu trúc quá mức, trong đó mean là phép tính trung bình:

$$L_{\text{offset}} = \text{mean}(\|\delta_{\text{offset}}\|) \tag{15}$$

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

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_{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^2_{total}) được mở rộng như sau:

$$L^2_{\text{total}} = L_{\text{MSE}} + \lambda^2_{\text{cp}} L_{\text{cp}} + \lambda^2_{\text{off}} L_{\text{offset}} + \lambda^2_{\text{sc}} L_{\text{sc}} \tag{16}$$

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

- $\lambda^2_{\text{cp}} = 0.01$ (trọng số nội dung)
- $\lambda^2_{\text{off}} = 0.5$ (trọng số độ lệch RSI)
- $\lambda^2_{\text{sc}} = 0.01$ (trọng số tương phản phong cách)

Việc bổ sung L_{sc} (như đã định nghĩa ở Phương trình 4 trong phần phân tích SCR) 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.

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

3.5.1. 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_{source}, ứng dụng sang chữ cái Hán D_{target} phức tạp), 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 gặp các ký tự Latin với cấu trúc thưa, sự chênh lệch miền (domain gap) khiến vector phong cách v_{gen} và v_{target} không còn tương đồng trong không gian tiềm ẩn.

3.5.2. Thiết kế module 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 (sampling strategy)** và **cơ chế tính hàm mất mát đa luồng**.

3.5.2.1. 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:

1. **Luồng Nội miền (Intra-Lingual Flow)**:
   - Anchor (x_{gen}): Ảnh sinh ra từ mô hình Diffusion.
   - Intra-Positive ($x_{\text{pos}}^{\text{intra}}$): Ảnh cùng nội dung ký tự, cùng phong cách (Ground Truth tiếng Trung). Giúp mô hình giữ vững cấu trúc cơ bản.
   - Intra-Negative ($x_{\text{neg}}^{\text{intra}}$): Ảnh cùng nội dung, khác phong cách.
2. **Luồng Xuyên miền (Cross-Lingual Flow - Điểm cải tiến chính)**:
   - Cross-Positive ($x_{\text{pos}}^{\text{cross}}$): Các ảnh thuộc ngôn ngữ đích (Chữ cái Latin) 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 ($x_{\text{neg}}^{\text{cross}}$): Các ảnh thuộc ngôn ngữ đích có cấu trúc nét tương đồng nhưng khác phong cách (Hard Negative Mining).

3.5.2.2. 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:

$$L_{\text{CL-SCR}} = \alpha_{\text{intra}} \cdot L_{\text{intra}} + \beta_{\text{cross}} \cdot L_{\text{cross}} \tag{17}$$

Trong đó: α_{intra} và β_{cross} là các siêu tham số trọng số (được thiết lập lần lượt là 0.3 và 0.7 trong thực nghiệm) nhằm ưu tiên khả năng chuyển giao phong cách sang ngôn ngữ đích.

Thành phần **Cross-Lingual InfoNCE Loss** (L_{cross}) được tính toán bằng cách tổng hợp qua L tầng đặc trưng (từ relu1_1 đến relu5_1 của VGG):

$$L_{\text{cross}} = -\sum_{l=1}^{L} \log \frac{\exp\left(v_{\text{gen}}^{l} \cdot v_{\text{pos,cross}}^{l} \ / \ \tau\right)}{\exp\left(v_{\text{gen}}^{l} \cdot v_{\text{pos}_l,\text{cross}}^{l} \ / \ \tau\right) + \sum_{k=1}^{K} \exp\left(v_{\text{gen}}^{l} \cdot v_{\text{neg}_k,\ \text{cross}}^{l} \ / \ \tau\right)} \tag{18}$$

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

3.5.2.3. 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:

$$L_{\text{Total}}^{2} = L_{\text{MSE}} + \lambda_{\text{content}} L_{\text{content}} + \lambda_{\text{offset}} L_{\text{offset}} + \lambda_{\text{style}} L_{\text{CL-SCR}} \tag{19}$$

Trong đó:

- L_{MSE} đảm bảo ảnh sinh ra không bị biến dạng quá nhiều so với ảnh gốc.
- L_{content} (Content Perceptual Loss) giữ gìn cấu trúc nét chữ.
- L_{offset} kiểm soát độ dịch chuyển của module 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.

3.6. Đề xuất thuật toán tính CL-SCR

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

Input	S	embedding style anchor rút ra từ sample
	P_{intra}	positive cùng ngôn ngữ (intra-language)
	P_{cross}	positive khác ngôn ngữ (cross-language)
	N_{intra}	negative cùng ngôn ngữ
	N_{cross}	negative khác ngôn ngữ
	α, β	hệ số trọng số cho từng nhánh
	L	số lượng tầng đặc trưng (feature layers)
	mode	{intra, cross, both}
Output	L_{total}	giá trị loss cuối cùng

procedure CAL_CL_SCR_LOSS($V_{\text{gen}}, V_{p_{\text{intra}}}, V_{n_{\text{intra}}}, V_{p_{\text{cross}}}, V_{n_{\text{cross}}}, \alpha_{\text{intra}}, \beta_{\text{cross}}, \tau$):
- $L_{\text{total}} \leftarrow 0.0$ ▷ Loss tổng
- count $\leftarrow 0$ ▷ Số nhánh loss được sử dụng
- **if mode** $\in \{\text{intra}, \text{both}\}$ và $P_{\text{intra}} \neq \emptyset$ then
  - $L_{\text{intra}} \leftarrow 0$
  - **for** $l = 1 \rightarrow$ L do ▷ Duyệt từng tầng embedding
    - $L_{\text{intra}} \leftarrow L_{\text{intra}} + \text{InfoNEC}\big(S^l, P^l_{\text{intra}}, N^l_{\text{intra}}\big)$
  - **end for**
  - $L_{\text{intra}} \leftarrow L_{\text{intra}\ /\ \text{L}}$ ▷ Trung bình theo tầng
  - **if** mode = both **then**
    - $L_{\text{total}} \leftarrow L_{\text{total}} + \alpha \cdot L_{\text{intra}}$ ▷ Áp dụng trọng số α
  - **else**
    - $L_{\text{total}} \leftarrow L_{\text{total}} + L_{\text{intra}}$
  - **end if**
  - count $\leftarrow$ count $+ 1$

```
    end if
    if mode ∈ {cross, both} và P_cross ≠ ∅ then
        L_cross ← 0
        for l = 1 → L do
            L_cross ← L_cross + InfoNEC(S^l, P^l_cross, N^l_cross)
        end for
        L_cross ← L_cross / L
        if mode = both then
            L_total ← L_total + β · L_cross        ▷ Áp dụng trọng số β
        else
            L_total ← L_total + L_cross
        end if
        count ← count + 1
    if count > 0 then
        L_total ← L_total / count                  ▷ Chuẩn hóa khi có nhiều nhánh
    end if
    return L_total
end if
```

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, luận văn 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)

Để đảm bảo khả năng tổng quát hóa của mô hình trên các hệ chữ viết khác nhau, em xây dựng một tập dữ liệu quy mô lớn bao gồm **818 bộ phông chữ**, đa dạng về phong cách (serif, sans-serif, thư pháp, viết tay, gothic, v.v.).

4.1.1. Cấu trúc và Tiền xử lý dữ liệu

Để phục vụ cho bài toán chuyển đổi phong cách đa ngôn ngữ hai chiều (Hán $\leftrightarrow$ Latin), bộ dữ liệu được tổ chức dựa trên các bộ phông chữ song ngữ (dual-script fonts), đảm bảo sự nhất quán về phong cách giữa hai hệ chữ. Cấu trúc dữ liệu bao gồm hai tập con chính tương tác lẫn nhau. Đầu tiên là tập ký tự Hán, chứa trung bình 800 ký tự thông dụng thuộc chuẩn GB2312, bao phủ đa dạng các mức độ phức tạp từ đơn giản đến kết cấu rậm. Trong kịch bản Latin $\rightarrow$ Hán, tập này đóng vai trò là miền đích (Target Domain) để đánh giá khả năng tái tạo cấu trúc, trong khi ở chiều ngược lại, nó cung cấp nguồn dữ liệu nội dung phong phú. Đối ứng với đó là tập ký tự Latin, bao gồm 52 ký tự chữ cái cơ bản (A-Z, a-z) với cấu trúc nét đặc thù khác biệt hoàn toàn

so với Hán tự. Việc khai thác các bộ phông chữ đa ngữ này cung cấp các cặp dữ liệu nhãn (Ground-truth) tự nhiên, đóng vai trò cốt lõi giúp module CL-SCR học được sự tương quan phong cách (style correlation) giữa hai hệ chữ.

Quy trình tiền xử lý: Về quy trình tiền xử lý, dữ liệu thô trải qua các bước chuẩn hóa để tối ưu hóa quá trình huấn luyện. Cụ thể, toàn bộ ảnh ký tự được render dưới dạng thang độ xám (grayscale) nhằm loại bỏ nhiễu 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 hóa đồng bộ về kích thước 64×64 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.

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 hóa, em sử dụng bộ giải thuật AdamW với tốc độ học khởi tạo là $\mathbf{1 \times 10^{-4}}$, 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 $\boldsymbol{\lambda_{\text{percep}} = 0.01}$ cho Content Perceptual Loss và $\boldsymbol{\lambda_{\text{offset}} = 0.5}$ cho Offset Loss nhằm hỗ trợ module 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. Em sử dụng bộ tối ưu hóa 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 $\mathbf{1 \times 10^{-4}}$.

Để tăng cường tính bền vững của biểu diễn phong cách đối với các biến thể hình học, em 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 75% đến 100% (scale 0.75 - 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 (Phase 2 - Style Refinement with CL-SCR): Bước sang giai đoạn hai, module 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 module tương phản. Tốc độ học được thiết lập ở mức thấp hơn là $\mathbf{1 \times 10^{-5}}$, áp dụng chiến lược Constant (hằng số) sau **1,000 bước khởi động**. Đối với cấu hình CL-SCR, em 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 $\boldsymbol{\beta_{\text{cross}} = 0.7}$, 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:

$$\mathcal{L}\text{total} = \mathcal{L}\text{MSE} + 0.01 \cdot \mathcal{L}\text{percep} + 0.5 \cdot \mathcal{L}\text{offset} + 0.01 \cdot \mathcal{L}_{\text{CL-SCR}} \quad (20)$$

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

Cấu hình Lấy mẫu: Em 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) 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 hóa 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 khóa luận thực hiện đánh giá định lượng trên một lượng lớn mẫu, quy trình lấy mẫu được tự động hóa 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, em thiết lập hai kịch bản kiểm thử với độ khó tăng dần (theo chuẩn của FontDiffuser và DG-Font):

1. **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.
2. **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)

4.3.1. Chỉ số Định lượng (Quantitative Metrics)

em 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:

$$\mathcal{L}_1 = \frac{1}{N}\sum_{i=1}^{N}|x_i - y_i| \tag{21}$$

Trong đó N là tổng số điểm ảnh. Giá trị L1 càng nhỏ càng tốt.

4.3.1.2. SSIM (Structural Similarity Index)

Độ đo **SSIM** 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:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{22}$$

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

$$\text{LPIPS}(x, y) = \sum_{l} \frac{1}{H_l W_l} \sum_{h,w} \| w_l \cdot (f_l^x(h,w) - f_l^y(h,w)) \|_2^2 \tag{23}$$

Giá trị LPIPS càng thấp chứng tỏ ảnh sinh càng giống ảnh thật về mặt thị giác tự nhiên.

4.3.1.4. FID (Fréchet Inception Distance)

Độ đo **FID** 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:

$$\text{FID} = \|\mu_r - \mu_g\|_2^2 + \text{Tr}\left(\sum_{r} + \sum_{g} - 2\left(\sum_{r} \sum_{g} \right)^{\frac{1}{2}} \right) \tag{24}$$

Giá trị FID càng thấp cho thấy phân bố của ảnh sinh càng tiệm cận với phân bố ảnh thật.

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, luận văn kết hợp bốn độ đo trên theo chiến lược đánh giá đa tầng:

- **Đánh giá độ chính xác điểm ảnh (Pixel-level Accuracy):** 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.
- **Đánh giá chất lượng cảm nhận (Perceptual Quality):** Đây là lý do LPIPS và FID được đưa vào. LPIPS đo lường sự tương đồng trong không gian đặc trưng (Feature Space) thay vì không gian điểm ảnh, giúp mô hình được "tha thứ" cho

những sai lệch nhỏ về pixel miễn là đặc điểm nhận dạng của chữ cái được bảo toàn. FID đóng vai trò trọng tâm trong việc đánh giá mức độ "thật" (realism) và tính đa dạng (diversity) của phong cách sinh ra, đảm bảo mô hình không bị rơi vào trạng thái "Mode Collapse" (chỉ sinh ra một vài mẫu lặp lại).

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

4.3.2. Đánh giá Định tính (User Study)

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, luận văn tiến hành một khảo sát đánh giá chủ quan (Subjective Evaluation) với sự tham gia của con người.

4.3.2.1. Thiết kế khảo sát

Để đánh giá chất lượng thị giác và tính nhất quán phong cách một cách khách quan nhất theo cảm nhận của con người, em thiết kế một bảng khảo sát mù (blind test) với sự tham gia của tổng cộng 30 tình nguyện viên. Nhóm khảo sát bao gồm 5 người bạn học chuyên ngành thiết kế đồ họa có kiến thức chuyên sâu về typography và 25 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 30 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, mỗi mẫu so sánh hiển thị một **ảnh tham chiếu (Reference Style)** (chứa phong cách mục tiêu) và các **ảnh kết quả (Generated Images)** là các ký tự được sinh ra bởi các mô hình cạnh tranh (DG-Font, FontDiffuser Baseline, và Phương pháp đề xuất Ours). Vị trí hiển thị của các ảnh kết quả được xáo trộn ngẫu nhiên để đảm bảo tính công bằng và loại bỏ thiên kiến vị trí. Tình nguyện viên được yêu cầu chọn ra ảnh có **độ nhất quán phong cách tốt nhất** và **chất lượng hình ảnh tổng thể cao nhất** trong số các lựa chọn.

4.3.2.2. Tiêu chí đánh giá

Người tham gia được yêu cầu chấm điểm hoặc lựa chọn ảnh tốt nhất dựa trên hai tiêu chí độc lập:

1. **Tính nhất quán phong cách (Style Consistency)**: Ảnh sinh ra có mang đúng "hồn" của ảnh tham chiếu kí tự Latin không? (Ví dụ: độ đậm nhạt, độ xước cọ, kiểu chân chữ serif/sans-serif).
2. **Tính toàn vẹn nội dung (Content Legibility)**: Ký tự Latin sinh ra có dễ đọc và đúng cấu trúc không? (Ví dụ: chữ '丘' có bị biến dạng thành hình thù kỳ quái không?).

Điểm số được dựa trên tỉ lệ phần trăm số phiếu bầu chọn cho mỗi mô hình.

4.4. Kết quả Thực nghiệm và Thảo luận

Trong chương này, em 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, CF-Font, FTransGAN), mô hình diffusion-based (DFS), và các phiên bản mô hình của em.

Để đánh giá toàn diện khả năng chuyển đổi đa ngôn ngữ, chúng em 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 module 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).

Em sử dụng hai cấu hình mô hình cho hướng này: Ours_{A} (sử dụng ký tự **A** làm ảnh phong cách tham chiếu) và Ours_{AZ} (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 tuyến tính hơn.

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

Việc phân loại theo độ phức tạp này giúp chúng em xác định module 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 (Quantitative Results)

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, CF-Font, DFS, GAS-NeXt 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 4.1 — 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	0.2773	0.2702	0.4023	106.3833
CF-Font	0.2659	0.2740	0.3979	91.2134
DFS	**0.1844**	<u>0.3900</u>	0.3548	40.4561
GAS-NeXt	0.2032	0.3812	0.3707	56.3950
FontDiffuser (Baseline)	0.1976	0.3775	0.2968	14.6871

Phương pháp	L1 ↓	SSIM ↑	LPIPS ↓	FID ↓
Ours_{A} (w/ CL-SCR)	<u>0.1927</u>	**0.3912**	**0.2868**	<u>12.3964</u>
Ours_{AZ} (w/ CL-SCR)	0.1939	0.3890	<u>0.2911</u>	**11.7691**

Bảng 4.2 — 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	0.2797	0.2654	0.3649	54.0974
CF-Font	0.2638	0.2716	0.3615	51.3925
DFS	**0.2089**	0.3048	0.3876	62.7206
GAS-NeXt	<u>0.2191</u>	0.3103	0.4073	84.8328
FontDiffuser (Baseline)	0.2283	0.2946	0.3184	29.0999
Ours_{A} (w/ CL-SCR)	0.2218	<u>0.3144</u>	**0.2892**	<u>17.8373</u>
Ours_{AZ} (w/ CL-SCR)	0.2214	**0.3197**	<u>0.2954</u>	**13.5508**

Dựa trên Bảng 4.1 và Bảng 4.2, chúng ta có thể rút ra những nhận xét quan trọng sau về hiệu năng của phương pháp đề xuất (Ours_{AZ} và Ours_{A}) so với các phương pháp State-of-the-Art (SOTA):

1. Sự vượt trội về chất lượng ảnh sinh (Chỉ số FID và LPIPS): Điểm nổi bật nhất trong kết quả thực nghiệm là sự cải thiện đột phá về chỉ số FID (Fréchet Inception Distance).

- Trong kịch bản **SFUC (Seen Font, Unseen Character)**, mô hình $\mathbf{Ours_{AZ}}$ đạt FID là **12.856**, **giảm hơn 67%** so với baseline mạnh nhất là FontDiffuser (**39.824**) và bỏ xa các phương pháp GAN truyền thống như DG-Font (**95.236**).
- Trong kịch bản khó hơn là **UFSC (Unseen Font, Seen Character)** – nơi mô hình phải sinh ảnh từ các font chưa từng thấy trong quá trình huấn luyện, Ours_{AZ} vẫn duy trì phong độ ấn tượng với FID 20.230, thấp hơn 3 lần so với FontDiffuser (65.336).

Điều này chứng minh rằng module CL-SCR đã giải quyết triệt để vấn đề "domain gap" giữa chữ Hán và chữ Latin. Trong khi FontDiffuser gốc thường gặp khó khăn trong việc áp đặt phong cách Hán lên cấu trúc Latin dẫn đến ảnh bị méo hoặc nhiễu (khiến FID cao), phương pháp đề xuất giúp ảnh sinh ra có độ tự nhiên (realism) cao và phân bố sát với ảnh thật. Tương tự, chỉ số LPIPS thấp nhất (0.292 và 0.326) cũng khẳng định ảnh sinh ra phù hợp với cảm nhận thị giác của mắt người hơn.

2. Khả năng bảo toàn cấu trúc (Chỉ số SSIM và L1):

- Về độ tương đồng cấu trúc (SSIM), phương pháp đề xuất Ours_{AZ} đạt kết quả cao nhất trong cả hai kịch bản (0.447 và 0.351), cho thấy các nét chữ Latin được tái tạo sắc nét, không bị gãy hoặc mất nét – một lỗi thường gặp ở DFS hay DG-Font.
- Về sai số điểm ảnh (L1), mặc dù GAS-NeXt đạt chỉ số tốt nhất ở kịch bản UFSC (0.219 so với 0.228 của Ours-AZ), nhưng FID của GAS-NeXt lại rất tệ (84.833). Đây là hiện tượng phổ biến: các mô hình GAN thường tối ưu hóa L1 bằng cách sinh ra các ảnh "trung bình cộng" bị mờ (blurry), trong khi Diffusion Model chấp nhận L1 cao hơn một chút để tạo ra các chi tiết tần số cao sắc nét (high-frequency details). Do đó, sự chênh lệch nhỏ về L1 là hoàn toàn chấp nhận được để đổi lấy chất lượng hình ảnh vượt trội.

3. Đánh giá tính ổn định qua các biến thể tham chiếu ($\mathbf{Ours_A}$ vs. $\mathbf{Ours_{AZ}}$):

Kết quả thực nghiệm cho thấy $\mathbf{Ours_{AZ}}$ đạt hiệu suất **vượt trội** hơn hẳn so với $\mathbf{Ours_{A}}$ trên cả hai kịch bản SFUC và UFSC (cụ thể FID giảm từ 16.945 xuống 12.856 ở SFUC). Điều này dẫn đến hai kết luận quan trọng:

Thứ nhất, mô hình tích hợp module CL-SCR có khả năng trích xuất đặc trưng phong cách bất biến (style-invariant features) cực tốt. Nó không học thuộc lòng (overfit) cấu trúc của ký tự `A` để suy ra phong cách, mà thực sự hiểu được bản chất của phong cách (như độ đậm nhạt, serif, texture) từ bất kỳ ký tự Latin nào được đưa vào.

Thứ hai, việc $\mathrm{Ours_{AZ}}$ đạt điểm cao hơn cho thấy phong cách được phân bố đa dạng trên toàn bộ bảng chữ cái. Khả năng tận dụng thông tin phong cách từ các ký tự ngẫu nhiên chứng tỏ mô hình có độ linh hoạt cao, phù hợp với bài toán thực tế khi người dùng có thể cung cấp bất kỳ ảnh mẫu nào chứ không nhất thiết phải là chữ `A`.

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

Bảng 4.3 — 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	0.1462	0.5542	0.2821	74.1655
CF-Font	0.1402	0.5621	0.2790	67.1241
DFS	0.1083	0.6140	0.2585	40.4042
GAS-NeXt	0.1158	0.6247	0.2900	79.1190
FontDiffuser (Baseline)	0.1223	0.6107	0.2270	21.2234
Ours_{All} (w/ CL-SCR)	0.1083	0.6406	0.2019	14.7298
$\text{Ours}_{\text{Easy}}$ (w/ CL-SCR)	**0.1079**	**0.6413**	**0.2018**	**14.6558**
$\text{Ours}_{\text{Medium}}$ (w/ CL-SCR)	<u>0.1082</u>	<u>0.6406</u>	<u>0.2024</u>	<u>14.8556</u>
$\text{Ours}_{\text{Hard}}$ (w/ CL-SCR)	0.1114	0.6318	0.2084	15.7662

Bảng 4.4 — 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	0.1397	0.5624	0.2751	89.8197
CF-Font	0.1317	0.5756	0.2726	84.3787
DFS	0.1139	0.5819	0.2907	75.2760
GAS-NeXt	0.1150	0.6094	0.3080	109.1815
FontDiffuser (Baseline)	0.1370	0.5731	0.2476	59.5788
Ours_{All} (w/ CL-SCR)	0.1090	0.6377	0.1985	**41.1152**
$\text{Ours}_{\text{Easy}}$ (w/ CL-SCR)	<u>0.1050</u>	<u>0.6439</u>	<u>0.1945</u>	<u>41.7273</u>
$\text{Ours}_{\text{Medium}}$ (w/ CL-SCR)	**0.1029**	**0.6466**	**0.1929**	43.6918
$\text{Ours}_{\text{Hard}}$ (w/ CL-SCR)	0.1050	0.6444	0.1982	45.5486

Dựa trên số liệu từ Bảng 4.3 và Bảng 4.4, 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 và hiệu quả chuyển đổi phong cách.

Thứ nhất, xét về hiệu năng tổng thể, mô hình đề xuất vượt trội hoàn toàn so với Baseline FontDiffuser ở cả hai kịch bản. Trên tập dữ liệu quen thuộc SFUC, cấu hình $\text{Ours}_{\text{Easy}}$ 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_{All} đạt FID 41.115, thấp hơn đáng kể so với mức 59.579 của Baseline. Điều này khẳng định rằng module CL-SCR không chỉ hiệu quả trong việc tinh chỉnh phong cách nội tại mà còn giúp mô hình tổng quát hóa tốt hơn khi phải đối mặt với các phong cách Hán tự lạ lẫm, phức tạp để áp dụng lên cấu trúc Latin đơn giản. So với các phương pháp GAN (DG-Font, CF-Font) hay GAS-NeXt vốn có chỉ số FID rất cao (trên 80 ở UFSC), phương pháp đề xuất chứng minh ưu thế tuyệt đối về độ tự nhiên và tính thẩm mỹ của ảnh sinh.

Thứ hai, phân tích sâu về độ phức tạp nét (stroke complexity) thông qua các biến thể Easy, Medium và Hard mang lại những góc nhìn giá trị. Tại bảng Bảng 4.4, có thể thấy cấu hình $\text{Ours}_{\text{Medium}}$ 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 (sweet spot) để trích xuất phong cách: chúng cung cấp đủ thông tin về bút pháp và kết cấu (hơn Easy) nhưng không gây ra quá nhiều nhiễu cấu trúc (structural noise) như các ký tự Hard (trên 21 nét). Khi sử dụng các ký tự quá phức tạp (Hard) để chuyển phong cách sang chữ Latin (vốn rất đơn giản), mô hình dễ gặp khó khăn trong việc lược bỏ các chi tiết thừa, dẫn đến chỉ số FID và L1 của $\text{Ours}_{\text{Hard}}$ thường kém hơn so với Easy và Medium.

Cuối cùng, mặc dù $\text{Ours}_{\text{Medium}}$ tối ưu về cấu trúc, nhưng Ours_{All} lại đạt chỉ số FID tốt nhất trên tập UFSC (41.115). Điều này cho thấy việc tiếp xúc với đa dạng các mức độ phức tạp trong quá trình huấn luyện giúp mô hình xây dựng được không gian biểu diễn phong cách phong phú nhất, từ đó sinh ra các hình ảnh có độ tự nhiên cao nhất về mặt cảm nhận thị giác, ngay cả khi độ chính xác từng điểm ảnh (L1) thua kém nhẹ so với cấu hình chuyên biệt Medium.

4.4.2. So sánh Định tính (Qualitative Analysis)

4.4.2.1. Đánh giá Cảm nhận Người dùng (User Study)

Phân tích:

- **Về cấu trúc:** Các phương pháp GAN thường gặp khó khăn với nét mảnh hoặc serif phức tạp, dẫn đến hiện tượng gãy nét (broken strokes).
- **Về phong cách:** FontDiffuser gốc bảo toàn nét tốt nhưng có xu hướng áp đặt đặc trưng bút lông (brush strokes) của chữ Hán vào chữ Latin, khiến chữ trông gượng ép.
- **Phương pháp đề xuất:** Nhờ cơ chế CL-SCR, ảnh sinh ra có các nét serific (chân chữ) sắc sảo, đúng chuẩn typography phương Tây hơn, đồng thời vẫn giữ được độ đậm và texture của font mẫu.

4.5. Nghiên cứu Bóc tách (Ablation Study)

Trong phần này, em 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, em cố định và lựa chọn hai cấu hình đại diện tiêu biểu nhất làm cơ sở so sánh:

1. Đối với hướng Latin → Hán tự (`e2c`): Em sử dụng cấu hình Ours_{AZ}. Đây là cấu hình chịu áp lực tổng quát hóa 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.

2. Đối với hướng Hán tự → Latin (`c2e`): Em sử dụng cấu hình Ours_{All}. 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 (Holistic View) 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 bốn yếu tố then chốt: các mô-đun kiến trúc, kỹ thuật tăng cường dữ liệu, chế độ hàm mất mát và số lượng mẫu âm.

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, em tiến hành thực nghiệm bóc tách (Ablation Study) bằng cách thay thế và bổ 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 (Đề xuất cải tiến)

Kết quả thực nghiệm trên hai hướng chuyển đổi được trình bày chi tiết tại Bảng 4.5 và Bảng 4.6.

Bảng 4.5 — Phân tích ảnh hưởng của các thành phần M, R, S và CL đối với hiệu năng mô hình trên tác vụ Latin → Hán tự.

		Mô-đun				L1 ↓	SSIM ↑	LPIPS ↓	FID ↓
		M	R	S	CL				
Ours_{AZ}	SFUC	✗	✗	✗	✗	0.2441	0.2983	0.4434	70.3650
		✓	✓	✓	✗	0.1976	0.3775	0.2968	14.6871
		✓	✓	✗	✓	**0.1939**	**0.3890**	**0.2911**	**11.7691**
	UFSC	✗	✗	✗	✗	0.2815	0.1965	0.4854	75.7399
		✓	✓	✓	✗	0.2283	0.2946	0.3184	29.0999
		✓	✓	✗	✓	**0.2214**	**0.3197**	**0.2954**	**13.5508**

Bảng 4.6 — Phân tích ảnh hưởng của các thành phần M, R, S và CL đối với hiệu năng mô hình trên tác vụ Hán tự → Latin.

		Mô-đun				L1 ↓	SSIM ↑	LPIPS ↓	FID ↓
		M	R	S	CL				
Ours_{All}	SFUC	✗	✗	✗	✗	0.2763	0.2491	0.4792	84.7434
		✓	✓	✓	✗	0.1223	0.6107	0.2270	21.2234
		✓	✓	✗	✓	**0.1083**	**0.6406**	**0.2019**	**14.7298**
	UFSC	✗	✗	✗	✗	0.3017	0.1793	0.5102	119.9425
		✓	✓	✓	✗	0.1370	0.5731	0.2476	59.5788
		✓	✓	✗	✓	**0.1090**	**0.6377**	**0.1985**	**41.1152**

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 thuần túy gặp 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).

- Trong hướng `e2c` (UFSC): Việc thay thế S bằng CL giúp FID giảm mạnh từ **29.10** xuống **13.55**.
- Trong 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 (Style Gap) 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.

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 hóa. 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 khóa 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 module 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 (scale-invariant features) và tránh hiện tượng học vẹt (overfitting). Để kiểm chứng điều này, em 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 4.7 và Bảng 4.8.

Bảng 4.7 — Phân tích 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	Ours_{AZ} (w/o Augment)	0.1974	0.3831	0.2967	14.1295
	Ours_{AZ} (w/ Augment)	**0.1939**	**0.3890**	**0.2911**	**11.7691**
UFSC	Ours_{AZ} (w/o Augment)	0.2295	0.3066	0.3060	15.7706
	Ours_{AZ} (w/ Augment)	**0.2214**	**0.3197**	**0.2954**	**13.5508**

Bảng 4.8 — Phân tích 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	Ours_{All} (w/o Augment)	**0.1076**	**0.6504**	**0.1978**	**12.3668**
	Ours_{All} (w/ Augment)	0.1083	0.6406	0.2019	14.7298
UFSC	Ours_{All} (w/o Augment)	0.1126	0.6364	0.2015	43.0665
	Ours_{All} (w/ Augment)	**0.1090**	**0.6377**	**0.1985**	**41.1152**

Nhận xét và Thảo luận:

Đối với hướng chuyển đổi từ Latin sang Hán tự (`e2c`), quan sát tại Bảng 4.7 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 hóa 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 module 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 4.8 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 Data Augmentation. Ở kịch bản SFUC, việc thiếu nhiễu 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 Data Augmentation là

thành phần thiết yếu để đảm bảo khả năng tổng quát hóa của mô hình trong các ứng dụng thực tế.

Kết luận: Dựa trên phân tích trên, em 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

Trong kiến trúc CL-SCR, hàm mất mát InfoNCE đóng vai trò điều hướng không gian biểu diễn phong cách. em 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 4.9 và Bảng 4.10.

Bảng 4.9 — Phân tích ảnh hưởng của các chế độ loss đối với hiệu năng mô hình trên tác vụ Latin → Hán tự (e2c).

	Phương pháp	L1 ↓	SSIM ↑	LPIPS ↓	FID ↓
SFUC	Ours_{AZ} (scr_intra)	<u>0.1969</u>	<u>0.3812</u>	<u>0.2958</u>	11.9552
	Ours_{AZ} (scr_cross)	0.1993	0.3770	0.2982	<u>11.8645</u>
	Ours_{AZ} (scr_both)	**0.1939**	**0.3890**	**0.2911**	**11.7691**
UFSC	Ours_{AZ} (scr_intra)	<u>0.2290</u>	<u>0.3008</u>	<u>0.3085</u>	<u>15.7197</u>
	Ours_{AZ} (scr_cross)	0.2326	0.2911	0.3128	16.2615
	Ours_{AZ} (scr_both)	**0.2214**	**0.3197**	**0.2954**	**13.5508**

Bảng 4.10 — Phân tích ảnh hưởng của các chế độ loss đối với hiệu năng mô hình trên tác vụ Hán tự → Latin (c2e).

	Phương pháp	L1 ↓	SSIM ↑	LPIPS ↓	FID ↓
SFUC	Ours_{All} (scr_intra)	**0.0993**	**0.6614**	**0.1903**	**13.6449**
	Ours_{All} (scr_cross)	0.1091	0.6436	0.2017	14.0159
	Ours_{All} (scr_both)	0.1083	0.6406	0.2019	14.7298
UFSC	Ours_{All} (scr_intra)	**0.0971**	**0.6601**	**0.1845**	41.3399
	Ours_{All} (scr_cross)	0.1175	0.6209	0.2095	44.7758
	Ours_{All} (scr_both)	0.1090	0.6377	0.1985	**41.1152**

Nhận xét và Thảo luận: Đối với hướng chuyển đổi từ Latin sang Hán tự (`e2c`), số liệu tại Bảng 4.9 phản ánh sự thống trị tuyệt đối 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 4.10, 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 "đậm đặc" (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 nhiễu 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.

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

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, em 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 4.11 và Bảng 4.12.

Bảng 4.11 — Phân tích ảnh hưởng của số lượng mẫu âm đối với hiệu năng mô hình trên tác vụ Latin → Hán tự (e2c).

	Phương pháp	L1 ↓	SSIM ↑	LPIPS ↓	FID ↓
SFUC	Ours_{AZ} (num_neg = 4)	**0.1939**	**0.3890**	**0.2911**	11.7691
	Ours_{AZ} (num_neg = 8)	0.1972	0.3835	0.2952	12.3750
	Ours_{AZ} (num_neg = 16)	0.1967	0.3833	0.2956	**10.6901**
UFSC	Ours_{AZ} (num_neg = 4)	**0.2214**	**0.3197**	**0.2954**	**13.5508**
	Ours_{AZ} (num_neg = 8)	0.2285	0.3048	0.3061	15.0245
	Ours_{AZ} (num_neg = 16)	0.2273	0.3064	0.3048	16.7855

Bảng 4.12 — Phân tích ảnh hưởng của số lượng mẫu âm đối với hiệu năng mô hình trên tác vụ Hán tự → Latin (c2e).

Phương pháp	L1 ↓	SSIM ↑	LPIPS ↓	FID ↓
Ours_{All} (num_neg = 4)	0.1083	0.6406	0.2019	**14.7298**

	Phương pháp	L1 ↓	SSIM ↑	LPIPS ↓	FID ↓
SFUC	Ours_{All} (num_neg = 8)	0.1080	0.6464	0.1999	14.8365
	Ours_{All} (num_neg = 16)	**0.1059**	**0.6468**	**0.1992**	15.7326
UFSC	Ours_{All} (num_neg = 4)	0.1090	0.6377	**0.1985**	**41.1152**
	Ours_{All} (num_neg = 8)	**0.1087**	**0.6398**	**0.1985**	43.8077
	Ours_{All} (num_neg = 16)	0.1111	0.6311	0.2008	43.5042

Phân tích số liệu từ thực nghiệm cho thấy một kết quả khá bất ngờ và 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 4.11), 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 4.12), mặc dù có sự phân hóa 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 củng 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.

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 và mạnh mẽ hơn, 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 hóa, đồng thời giảm tải đáng kể chi phí huấn luyện.

4.6. Phân tích các trường hợp thất bại (Failure Case Analysis)

Chương 5

Kết Luận và Hướng Phát Triển

5.1. Kết quả đạt được

5.2. Các định hướng phát triển

Tài liệu tham khảo

[1] Sohl-Dickstein J, Weiss EA, Maheswaranathan N, Ganguli S. Deep Unsupervised Learning using Nonequilibrium Thermodynamics. trong: Bach FR, Blei DM, biên tập viên. Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015, vol. 37, JMLR.org; 2015, tr 2256–65.

[2] Ho J, Jain A, Abbeel P. Denoising Diffusion Probabilistic Models. trong: Larochelle H, Ranzato M, Hadsell R, Balcan M-F, Lin H-T, biên tập viên. Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020.

[3] Xie Y, Chen X, Sun L, Lu Y. DG-Font: Deformable Generative Networks for Unsupervised Font Generation. trong:. IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021, Computer Vision Foundation / IEEE; 2021, tr 5130–40. https://doi.org/10.1109/CVPR46437.2021.00509.

[4] Chen X, Xie Y, Sun L, Lu Y. DGFont++: Robust Deformable Generative Networks for Unsupervised Font Generation. CoRR 2022. https://doi.org/10.48550/ARXIV.2212.14742.

[5] Wang C, Zhou M, Ge T, Jiang Y, Bao H, Xu W. CF-Font: Content Fusion for Few-Shot Font Generation. trong:. IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023, IEEE; 2023, tr 1858–67. https://doi.org/10.1109/CVPR52729.2023.00185.

[6] Park S, Chun S, Cha J, Lee B, Shim H. Multiple Heads are Better than One: Few-shot Font Generation with Multiple Localized Experts. trong:. 2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021, IEEE; 2021, tr 13880–9. https://doi.org/10.1109/ICCV48922.2021.01364.

[7] Wang W, Sun D, Zhang J, Gao L. MX-Font++: Mixture of Heterogeneous Aggregation Experts for Few-shot Font Generation. trong:. 2025 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2025, Hyderabad, India, April 6-11, 2025, IEEE; 2025, tr 1–5. https://doi.org/10.1109/ICASSP49660.2025.10888465.

[8] Yang Z, Peng D, Kong Y, Zhang Y, Yao C, Jin L. FontDiffuser: One-Shot Font Generation via Denoising Diffusion with Multi-Scale Content Aggregation and Style Contrastive Learning. trong: Wooldridge MJ, Dy JG, Natarajan S, biên tập viên. 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 2014, February 20-27, 2024, Vancouver, Canada, AAAI Press; 2024, tr 6603–11. https://doi.org/10.1609/AAAI.V38I7.28482.