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

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

Sinh viên thực hiện: **Trần Đình Khánh Đăng**
Giảng viên hướng dẫn: **TS. Dương Việt Hằng**
Lớp khoá học: KHMT2022.1
Khoa: Khoa học máy tính

Mục lục

Mục lục

Thiết kế phông chữ

Ứng dụng rộng rãi của các phông chữ trong đời sống thực

Thách thức của thiết kế truyền thống

Mặc dù nhu cầu sử dụng phông chữ rất lớn, quy trình thiết kế truyền thống gặp nhiều trở ngại:

1. Tốn kém chi phí & thời gian:

- Phải vẽ thủ công từng nét để đảm bảo tính thẩm mỹ.
- Quy trình lặp lại nhàm chán.

2. Thách thức về quy mô:

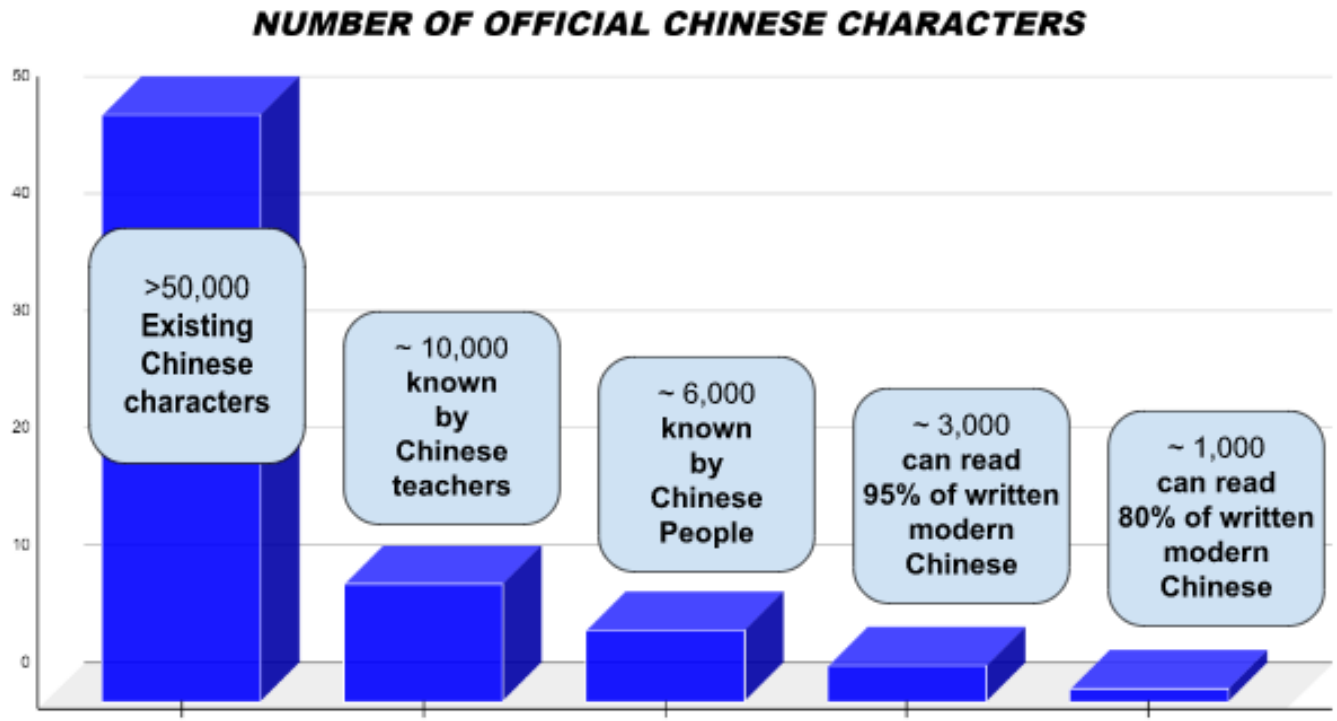
- Latin: Chỉ 52 ký tự.
- **CJK (Hán/Nôm):** Hàng chục nghìn ký tự. → **Rất tốn kém nếu làm thủ công hoàn toàn.**

3. Hạn chế về hỗ trợ đa ngôn ngữ:

- Các font nghệ thuật đẹp thường chỉ hỗ trợ ngôn ngữ phổ biến (Anh, Trung).
- Thiếu các **glyph Latin mở rộng** (như tiếng Việt: ă, â, đ...) hoặc hệ chữ ít phổ biến (Thái, Lào).

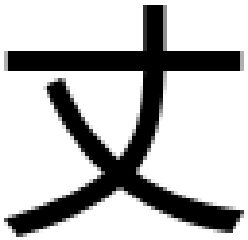
Thách thức của thiết kế truyền thống

→ Không thể tái sử dụng trực tiếp nếu không tự thiết kế bổ sung các ký tự thiếu.



Giải pháp: One-shot Font Generation

Thay vì vẽ thủ công hàng chục nghìn ký tự, AI sẽ "học" phong cách từ **một chữ mẫu duy nhất** để sinh ra toàn bộ bộ font.



Nội dung
(Ký tự gốc)

+

Phong cách
(1 Mẫu)

→

Kết quả
(Font mới)

Giải pháp: One-shot Font Generation

Giải quyết triệt để 3 thách thức trên:

- ✓ **Tốc độ & Chi phí:** Rút ngắn quy trình từ hàng tháng xuống vài giây.
- ✓ **Mở rộng quy mô:** Sinh tự động hàng vạn ký tự Hán/Nôm mà không tốn sức người.
- ✓ **Hỗ trợ đa ngôn ngữ:** Tự động sinh các **glyph thiếu** (như dấu tiếng Việt, ký tự Thái) từ các font nước ngoài, giúp tái sử dụng tài nguyên font hiệu quả.

Mục tiêu & Đóng góp của Khoá luận

Tuy nhiên, đa số mô hình hiện tại chỉ làm tốt trên đơn ngữ (VD: Hán → Hán).

Mục tiêu khoá luận: Xây dựng giải pháp **Cross-Lingual (Đa ngôn ngữ)** tổng quát.

→ **Phạm vi kiểm chứng (Scope):** Tập trung vào cặp **Latin - Hán tự**. (Lý do: Đây là cặp có cấu trúc khác biệt lớn nhất, đóng vai trò là trường hợp khó nhất để đánh giá khả năng của mô hình).

Đóng góp chính:

1. Xây dựng pipeline dựa trên **Diffusion Model** (thay vì GAN).
2. Đề xuất mô-đun **CL-SCR** để xử lý sự chênh lệch cấu trúc giữa hai hệ chữ này.

Mục lục

Khoảng cách hình thái học

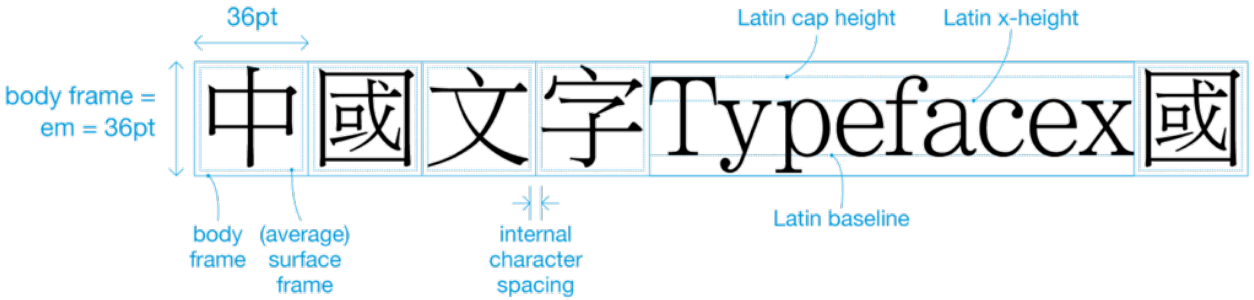
Tại sao cặp Latin - Hán tự lại là thách thức lớn nhất?

1. Latin (Đại diện hệ chữ cái):

- Cấu trúc tuyến tính (Linear).
- Ít nét, mật độ thưa.
- **Vấn đề:** Thiếu thông tin để suy diễn sang chữ phức tạp.

2. Hán tự (Đại diện hệ tượng hình):

- Cấu trúc khối vuông (Square block).
- Nét dày đặc, phức tạp.
- **Vấn đề:** Dễ bị biến dạng cấu trúc khi áp dụng phong cách lạ.



Khoảng cách hình thái học

→ Khoảng cách (Gap) giữa hai nhóm này chính là rào cản lớn nhất mà mô hình cần vượt qua.

Tiếp cận giải quyết vấn đề

Với khoảng cách hình thái lớn như vậy, các phương pháp hiện tại xử lý ra sao?

1. Các phương pháp dựa trên GAN: (Ví dụ: DG-Font, FTransGAN)

- **Cơ chế:** Cố gắng học ánh xạ trực tiếp giữa hai miền ảnh.
- **Thất bại:** Do cấu trúc quá khác biệt, GAN thường sinh ra ảnh bị **Mode Collapse** (biến dạng) hoặc **Blur** (mờ) khi cố gắng "ép" chữ Latin thành khối vuông Hán tự và ngược lại.

2. Tại sao chọn Diffusion Model?

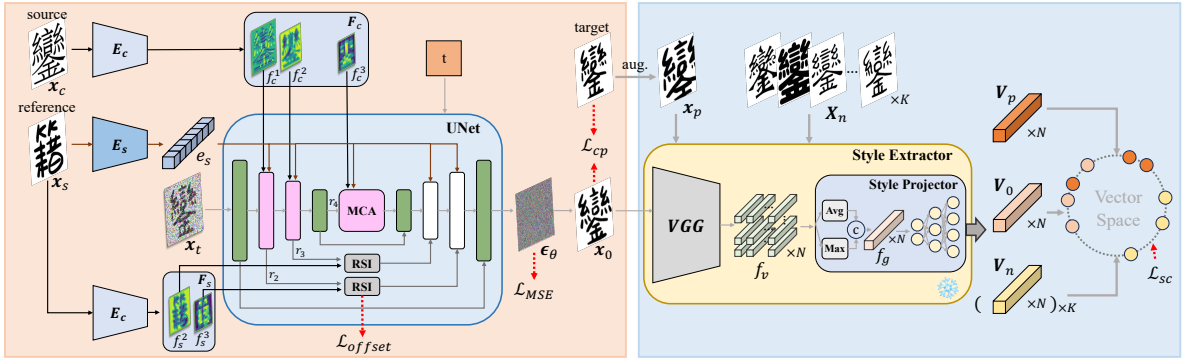
- **Cơ chế:** Khử nhiễu dần dần (Denoising) từ trạng thái vô định hình.
- **Ưu điểm:** Cho phép kiểm soát cấu trúc (Structure) và phong cách (Style) tách biệt tốt hơn.

→ **Đây là chìa khoá để bắc cầu qua "Morphological Gap".**

Mục lục

Tổng quan kiến trúc

Kiến trúc dựa trên FontDiffuser với quy trình huấn luyện 2 giai đoạn:



(a) Conditional Diffusion for Font Generation

(b) Style Contrastive Refinement

Các thành phần chính:

1. **MCA (Phase 1):** Tổng hợp nội dung đa tỷ lệ, đảm bảo giữ nét chi tiết.
2. **RSI (Phase 1):** Sử dụng Deformable Conv để xử lý biến dạng cấu trúc.

3. CL-SCR (Phase 2 - New): Thay thế mô-đun SCR cũ. Chịu trách nhiệm học phong cách **xuyên ngôn ngữ** (Cross-Lingual).

Mô-đun CL-SCR

Giải pháp cho vấn đề "Domain Gap" giữa hai ngôn ngữ.

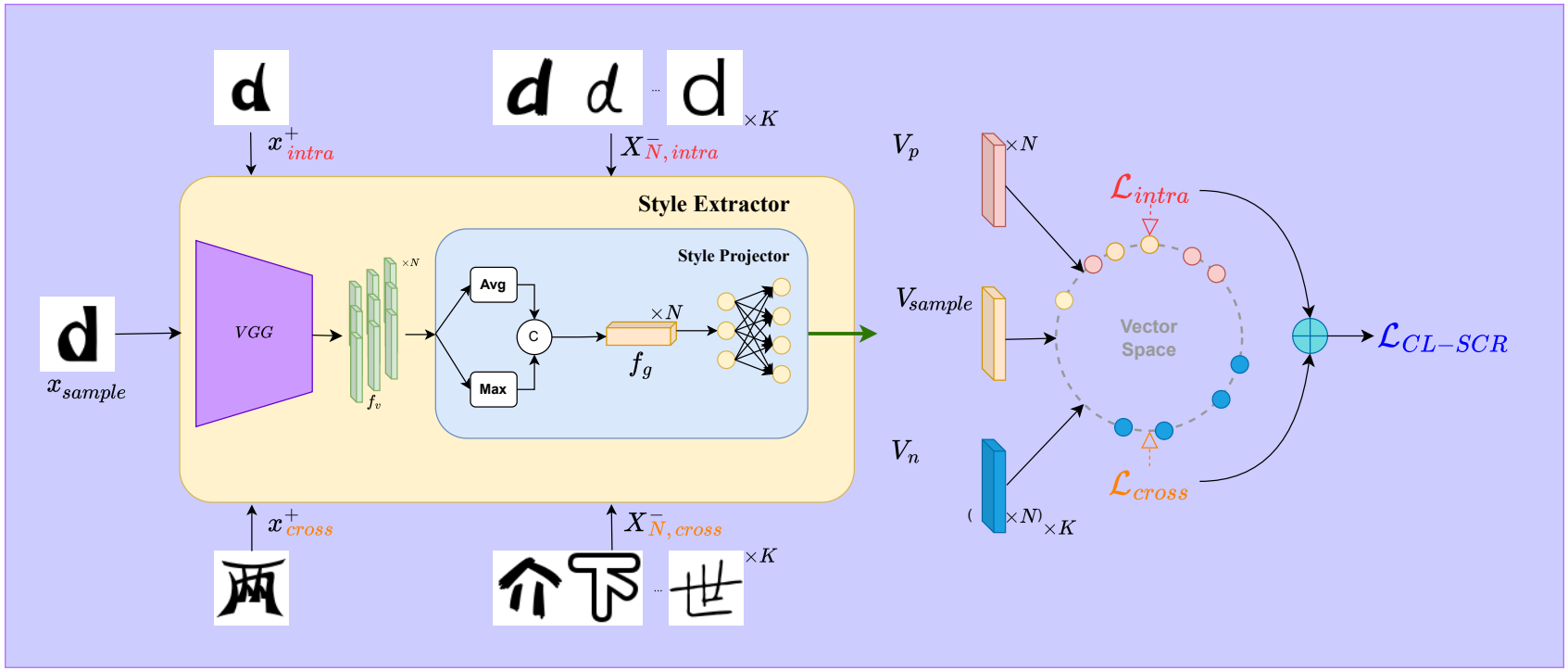
Cải tiến chiến lược lấy mẫu (Sampling Strategy):

Thay vì chỉ lấy mẫu trong cùng ngôn ngữ, module CL-SCR mở rộng phạm vi tương phản để bắt cầu nối giữa hai miền dữ liệu:

- **Intra-domain (Nội miền):** Sử dụng cặp ảnh thuộc **cùng ngôn ngữ**.
  → Mục tiêu: Giữ sự ổn định và nhất quán của phong cách nội tại.
- **Cross-domain (Liên miền):** Sử dụng cặp ảnh thuộc **hai ngôn ngữ khác nhau** (Latin ↔ Chinese).
  → Mục tiêu: Học cách chuyển giao đặc trưng phong cách sang ngôn ngữ đích.

→ **Kỹ thuật:** Tăng số lượng mẫu âm ($K = 4$) buộc mô hình phải học các đặc trưng phong cách tinh tế hơn (fine-grained features) thay vì chỉ học vẹt.

Mô-đun CL-SCR



Hình 3.7: Kiến trúc mạng CL-SCR với hai luồng giám sát Intra và Cross.

Công thức tính Loss

Hàm mất mát được xây dựng dựa trên nguyên lý **InfoNCE**, tối ưu hoá đồng thời hai luồng:

1. Intra-Lingual Loss (L_{intra}): Mẫu dương (+) và mẫu âm (-) được lấy từ **cùng tập ngôn ngữ** với ảnh đang sinh.

$$L_{\text{intra}} = -\log \frac{\exp\left(\frac{q \cdot k^+}{\tau}\right)}{\exp\left(\frac{q \cdot k^+}{\tau}\right) + \sum_{i=0}^{K} \exp\left(\frac{q \cdot k_i^-}{\tau}\right)}$$

Mục tiêu: Đảm bảo tính nhất quán phong cách trong nội bộ ngôn ngữ nguồn.

2. Cross-Lingual Loss (L_{cross}): Mẫu dương (+) và mẫu âm (-) được lấy từ **tập ngôn ngữ còn lại** (ngôn ngữ đích).

$$L_{\text{cross}} = -\log \frac{\exp\left(\frac{q \cdot k_{\text{cross}}^+}{\tau}\right)}{\exp\left(\frac{q \cdot k_{\text{cross}}^+}{\tau}\right) + \sum_{i=0}^{K} \exp\left(\frac{q \cdot k_{\text{cross,i}}^-}{\tau}\right)}$$

Mục tiêu: Kéo phong cách của ảnh sinh lại gần phong cách của ngôn ngữ đích bất chấp khác biệt cấu trúc.

Công thức tính Loss

Tổng hợp CL-SCR Loss:

$$L_{\text{CL-SCR}} = \alpha \cdot L_{\text{intra}} + \beta \cdot L_{\text{cross}}$$

Hàm mục tiêu tổng quát

Mô hình được huấn luyện để tối ưu hoá đồng thời độ chính xác nội dung, cấu trúc và phong cách:

$$L_{\text{total}} = \underbrace{L_{\text{MSE}}}_{\text{Tái tạo ảnh}} + \lambda_1 \underbrace{L_{\text{percep}}}_{\text{Nội dung}} + \lambda_2 \underbrace{L_{\text{offset}}}_{\text{Biến dạng}} + \lambda_3 \underbrace{L_{\text{CL-SCR}}}_{\text{Phong cách (Đề xuất)}}$$

- $\boldsymbol{L_{\text{MSE}}}$: Đảm bảo ảnh sinh ra giống ảnh gốc ở cấp độ pixel (Phase 1).
- $\boldsymbol{L_{\text{percep}}}$: Giữ lại các đặc trưng thị giác cấp cao (VGG features).
- $\boldsymbol{L_{\text{offset}}}$: Ràng buộc sự biến dạng của RSI để không phá vỡ cấu trúc chữ.
- $\boldsymbol{L_{\text{CL-SCR}}}$: Đóng góp chính của khoá luận, giải quyết bài toán đa ngôn ngữ (Phase 2).

Mục lục

Bộ dữ liệu

Kế thừa bộ dữ liệu chuẩn từ **FTransGAN**.

- **Quy mô: 818** bộ phông chữ song ngữ (Bao gồm Serif, Sans-serif, Thư pháp...).
- **Cấu trúc cặp:**

  Latin: **52** ký tự cơ bản.

  Hán tự: **800** ký tự thông dụng.
- **Đặc điểm:** Nhất quán tuyệt đối về phong cách giữa hai hệ chữ → Cung cấp **Ground-truth** tự nhiên cho việc học.

Kịch bản đánh giá

Tuân theo chuẩn của FTransGAN và FontDiffuser.

SFUC (Seen Font, Unseen Char):

- Font đã biết, sinh ký tự mới.
- **Mục tiêu:** Đánh giá khả năng **nội suy phong cách**.

UFSC (Unseen Font, Seen Char):

- Font **mới hoàn toàn** (chưa từng thấy khi train).
- **Mục tiêu:** Đánh giá khả năng **One-shot Generalization** (Kịch bản khó nhất & Quan trọng nhất).

Cấu hình Huấn luyện & Suy diễn

1. Môi trường & Giai đoạn 1:

- **Phần cứng:** Kaggle Cloud, GPU NVIDIA Tesla P100 (16GB).
- **Framework:** PyTorch, Diffusers.
- **Phase 1:** 400.000 bước, Batch 4, AdamW ($\texttt{lr} = 1 \times 10^{-4}$).
- **Mục tiêu:** Học cấu trúc nội dung và phong cách cơ bản.

2. Tiền huấn luyện CL-SCR:

- **Quy mô:** 200.000 bước, Batch 16, AdamW
- **Augmentation:** Random Resized Crop (Scale 0.8-1.0) → Tăng tính bền vững với biến thể hình học.

3. Giai đoạn 2 - Tinh chỉnh:

- **Thiết lập:** 30k bước, Batch 4, giảm $\texttt{lr} = 1 \times 10^{-5}$ để tránh phá vỡ cấu trúc.
- **CL-SCR:** Chế độ `both` (Nội miền + Xuyên miền), $\alpha = 0.3, \beta = 0.7, K = 4$.
- **Hàm Loss tổng hợp:**

$$L_{\text{total}} = L_{\text{MSE}} + 0.01L_{\text{percep}} + 0.5L_{\text{offset}} + 0.01L_{\text{CL-SCR}}$$

4. Quy trình Inference:

- **Sampler:** DPM-Solver++ (20 steps) để cân bằng tốc độ/chất lượng.
- **Guidance:** Classifier-free Guidance (CFG).

Các thước đo đánh giá

Để đánh giá toàn diện, khoá luận sử dụng hệ thống đo lường đa tầng:

1. Định lượng (Quantitative):

L1 & SSIM	Độ chính xác về điểm ảnh & cấu trúc (Pixel-level).
LPIPS	Độ tương đồng nhận thức (Perceptual distance).
FID	**(Quan trọng nhất)** Đo khoảng cách phân bố giữa ảnh sinh và ảnh thật (Độ chân thực).

2. Định tính (Qualitative):

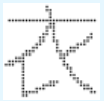
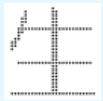
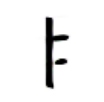
- **Visual Inspection:** So sánh bằng mắt thường các chi tiết nét (gai, xước, đậm/nhạt).
- **User Study:** Khảo sát mù (Blind Test) trên 20 người dùng để đánh giá độ hài lòng thị giác.

→ **Kết hợp cả độ chính xác máy học và cảm nhận con người.**

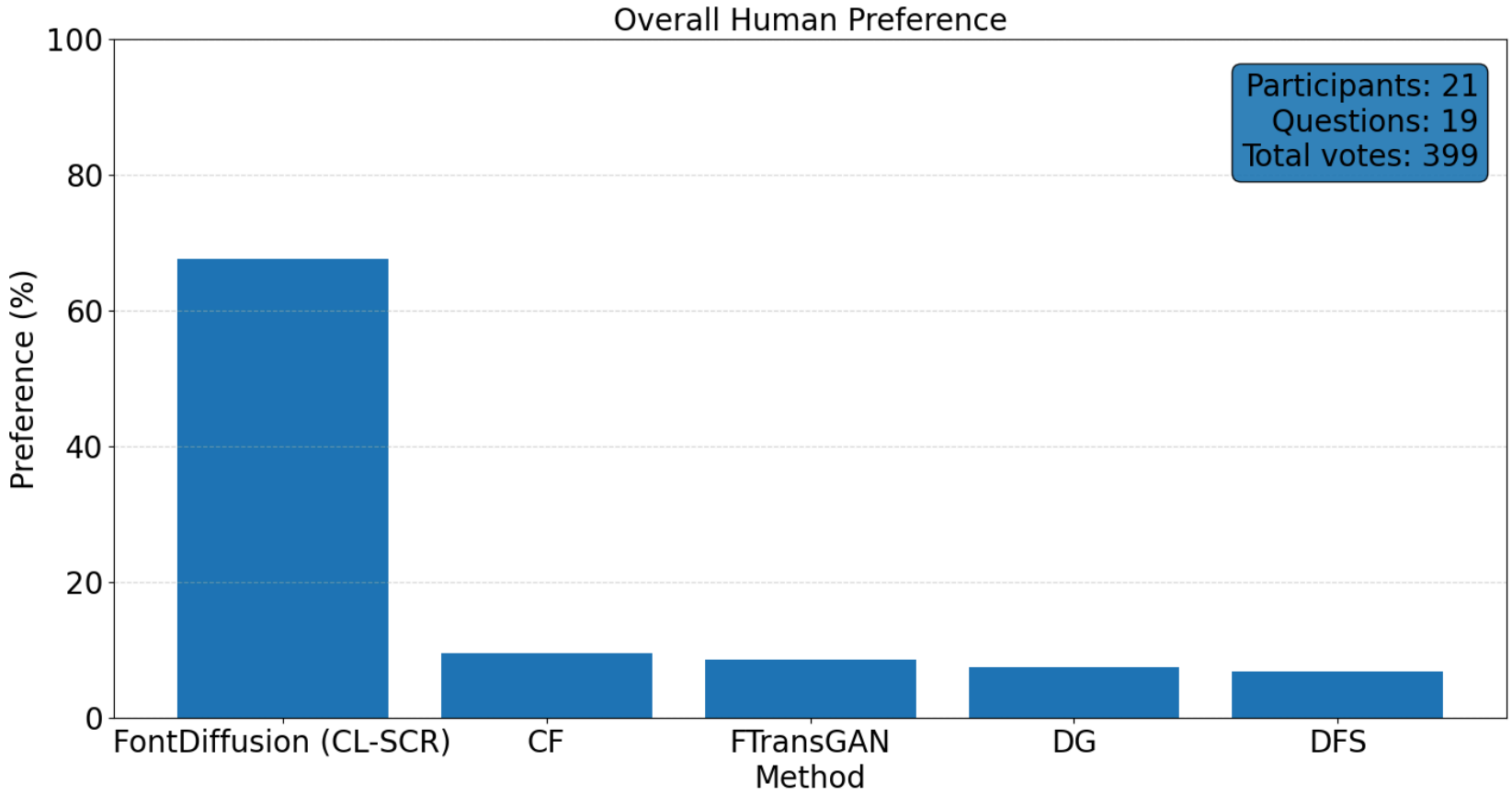
Kết quả định lượng

	Model	SFUC				UFSC			
		L1 ↓	SSIM ↑	LPIPS ↓	FID ↓	L1 ↓	SSIM ↑	LPIPS ↓	FID ↓
L → C	DG-Font	0.2773	0.2702	0.4023	106.38	0.2797	0.2654	0.3649	54.09
	CF-Font	0.2659	0.2740	0.3979	91.21	0.2638	0.2716	0.3615	51.39
	DFS	0.2131	0.3558	0.3812	45.42	**0.2008**	0.3048	0.3876	62.72
	FTransGAN	**0.1844**	**0.3900**	0.3548	40.45	0.2089	0.3109	0.3329	42.10
	FontDiffuser (Baseline)	0.1976	0.3775	0.2968	14.68	0.2283	0.2946	0.3184	29.09
	Ours	0.1939	0.3890	**0.2911**	**11.76**	0.2214	**0.3197**	**0.2954**	**13.55**
C → L	DG-Font	0.1462	0.5542	0.2821	74.1655	0.1397	0.5624	0.2751	89.8197
	CF-Font	0.1402	0.5621	0.2790	67.1241	0.1317	0.5756	0.2726	84.3787
	DFS	**0.1083**	0.6140	0.2585	40.4042	0.1139	0.5819	0.2907	75.2760
	FTransGAN	0.1381	0.5291	0.2851	55.5859	0.1456	0.4949	0.3023	88.4450
	FontDiffuser (Baseline)	0.1223	0.6107	0.2270	21.2234	0.1370	0.5731	0.2476	59.5788
	Ours	**0.1083**	**0.6406**	**0.2019**	**14.7298**	**0.1090**	**0.6377**	**0.1985**	**41.1152**

Kết quả định tính

Source										
Reference										
DG-Font										
CF-Font										
DFS										
FTransGAN										
FontDiffuser (Baseline)										
Ours										
Target										

Đánh giá người dùng



Hiệu quả của các mô-đun kiến trúc

	Mô-đun				SFUC				UFSC			
	M	R	S	CL	L1 ↓	SSIM ↑	LPIPS ↓	FID ↓	L1 ↓	SSIM ↑	LPIPS ↓	FID ↓
C	✘	✘	✘	✘	0.2441	0.2983	0.4434	70.3650	0.2815	0.1965	0.4854	75.7399
C → L	✓	✓	✓	✘	0.1976	0.3775	0.2968	14.6871	0.2283	0.2946	0.3184	29.0999
C → L	✓	✓	✘	✓	**0.1939**	**0.3890**	**0.2911**	**11.7691**	**0.2214**	**0.3197**	**0.2954**	**13.5508**
L	✘	✘	✘	✘	0.2763	0.2491	0.4792	84.7434	0.3017	0.1793	0.5102	119.9425
L → C	✓	✓	✓	✘	0.1223	0.6107	0.2270	21.2234	0.1370	0.5731	0.2476	59.5788
L → C	✓	✓	✘	✓	**0.1083**	**0.6406**	**0.2019**	**14.7298**	**0.1090**	**0.6377**	**0.1985**	**41.1152**

Mục lục

Tổng kết đóng góp

Khoá luận đã hoàn thành các mục tiêu đề ra ban đầu:

- ⋆ **Giải quyết bài toán khó:** Xây dựng thành công pipeline chuyển đổi phong cách đa ngôn ngữ (Cross-Lingual) giữa Latin và Hán tự.
- ✓ **Đóng góp kỹ thuật:** Đề xuất mô-đun **CL-SCR** với cơ chế Loss hỗn hợp (Intra + Cross), giúp tách biệt hiệu quả nội dung và phong cách.
- ✓ **Hiệu quả thực nghiệm:** Vượt trội SOTA hiện tại (FID giảm ∼50% ở chiều Latin → Hán), khắc phục được lỗi "bóng ma" và "biến dạng cấu trúc" của các dòng GAN.

Hạn chế & Hướng phát triển

Hạn chế:

- **Tốc độ suy diễn chậm:** Do bản chất của Diffusion (20 bước khử nhiễu) → Chậm hơn GAN 60 lần.
- **Tài nguyên tính toán:** Yêu cầu VRAM lớn hơn để lưu trữ các trạng thái trung gian.

→ **Chưa phù hợp cho ứng dụng Real-time.**

Hướng phát triển:

- **Tối ưu tốc độ:** Áp dụng **Consistency Distillation** hoặc **Latent Diffusion** để giảm số bước lấy mẫu (4-8 bước).
- **Mở rộng ngôn ngữ:** Thử nghiệm trên tiếng Việt (Thư pháp/Quốc ngữ), tiếng Thái.
- **Đa dạng đầu ra:** Sinh font dạng Vector (SVG) để designer dễ dàng chỉnh sửa.

Công trình khoa học

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

Lời cảm ơn

Xin cảm ơn Thầy Cô và Hội đồng đã theo dõi và lắng nghe!

Sinh viên thực hiện:	**Trần Đình Khánh Đăng**
Giảng viên hướng dẫn:	**TS. Dương Việt Hằng**
Lớp khoá học:	KHMT2022.1
Khoa:	Khoa học máy tính

Phụ lục

So sánh chỉ số quan trọng nhất (**FID**) trên kịch bản khó (**UFSC**):

Tối ưu hoá CL-SCR

Cơ sở thực nghiệm để lựa chọn các siêu tham số tốt nhất.

a. Chế độ Hàm Loss (Loss Modes): Tại sao phải kết hợp cả Intra và Cross?

Chế độ	FID (UFSC) ↓	
	L → C	C → L
Intra-only	<u>15.72</u>	<u>41.34</u>
Cross-only	16.26	44.78
Both	**13.55**	**41.12**

→ **Both** tận dụng sự ổn định của Intra và khả năng chuyển đổi của Cross.

b. Trọng số Alpha (α) & Beta (β): Tại sao ưu tiên $\beta = 0.7$?

α	β	FID (UFSC) ↓	
		L → C	C → L
0.7	0.3	<u>14.48</u>	45.23
0.5	0.5	15.18	<u>43.42</u>
0.3	**0.7**	**13.55**	**41.12**

→ Bài toán Cross-Lingual cần ưu tiên học các đặc trưng xuyên ngôn ngữ (β lớn).

Phân tích độ nhạy

Ảnh hưởng của Số mẫu âm & Guidance Scale

c. Số lượng mẫu âm (K): Trong hàm loss InfoNCE.

K	FID (UFSC) ↓	
	L → C	C → L
4	**13.55**	**41.12**
8	<u>15.02</u>	43.81
16	16.79	<u>43.50</u>

→ **K=4** là điểm cân bằng tối ưu cho cả hai chiều.

d. Trọng số hướng dẫn (Scale - s): Cân bằng giữa đa dạng và chính xác.

Scale (s)	FID (UFSC) ↓	
	L → C	C → L
2.5	**13.28**	<u>40.05</u>
5.0	<u>13.39</u>	**40.00**
7.5	13.55	41.12
10.0	13.78	44.74
12.5	14.78	47.15
15.0	17.01	52.76

→ s thấp ($\in * [2.5, 7.5] *$) cho kết quả tốt nhất.

Phân tích độ nhạy

Đánh giá hiệu quả của chiến lược Tăng cường dữ liệu (Data Augmentation).

e. Tăng cường dữ liệu: So sánh mô hình khi dùng/ không dùng kỹ thuật tăng cường dữ liệu.

Cấu hình	FID (UFSC) ↓	
	L → C	C → L
w/o Augmentation	15.77	43.07
w/ Augmentation	**13.55**	**41.12**

→ Việc áp dụng Augmentation giúp giảm đáng kể FID, chứng tỏ mô hình học được các đặc trưng phong cách **bền vững** hơn, tránh bị Overfitting.

Chiến lược: Random Resized Crop

- **Scale (0.8 — 1.0):** Cắt ngẫu nhiên nhưng giữ lại phần lớn cấu trúc chữ.
- **Ratio (0.8 — 1.2):** Thay đổi tỷ lệ khung hình nhẹ để mô phỏng các biến thể viết tay.

→ Giúp mô-đun **CL-SCR** không bị "học vẹt" (memorize) các vị trí pixel cố định.