

# 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

1. Giới thiệu
2. Thách thức và Cơ sở lý thuyết
3. Phương pháp đề xuất
4. Thực nghiệm và Đánh giá
5. Kết luận

# Mục lục

## 1. Giới thiệu

## 2. Thách thức và Cơ sở lý thuyết

## 3. Phương pháp đề xuất

## 4. Thực nghiệm và Đánh giá

## 5. Kết luận

# Thiết kế phông chữ



# Ứng dụng rộng rãi của các phong chữ trong đời sống thực



# Thách thức của thiết kế truyền thống

Mặc dù nhu cầu sử dụng phong 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 (Scale):

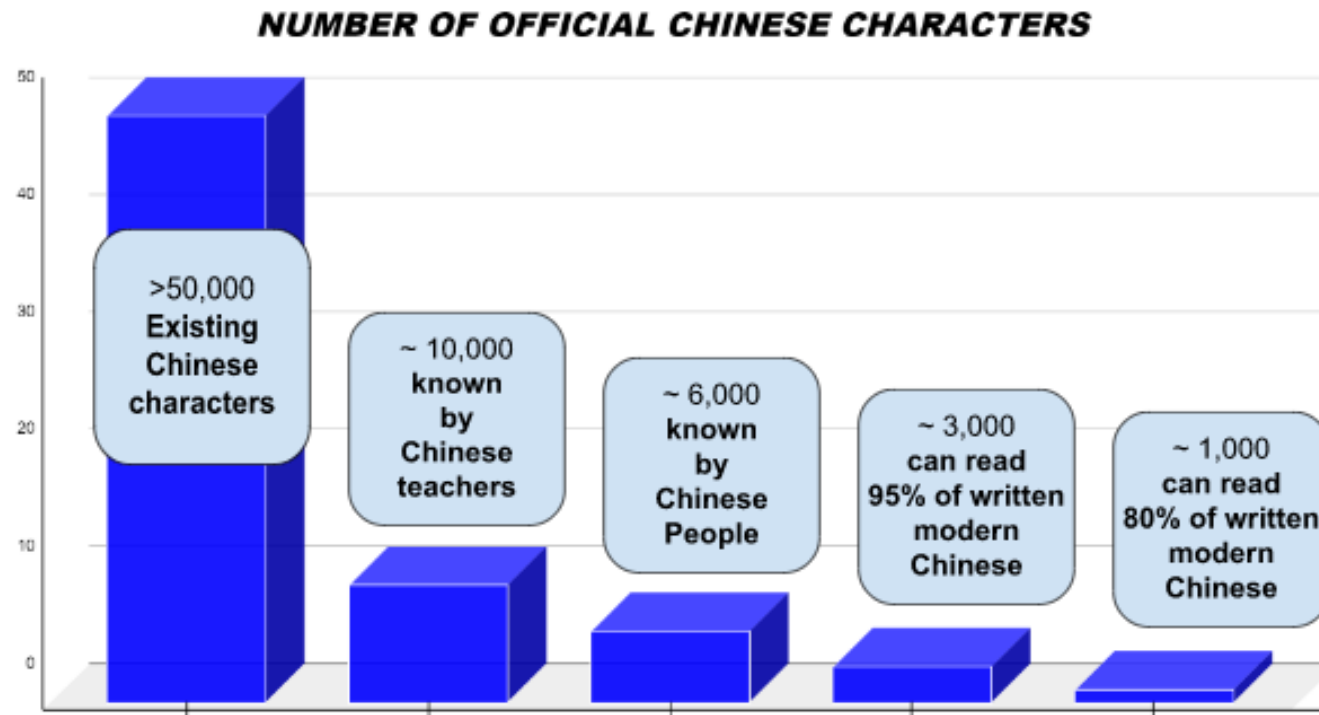
- Latin: Chỉ 52 ký tự.
- **CJK (Hán/Nôm)**: Hàng chục nghìn ký tự. → **Rất tốn kém nếu làm thủ công hoàn toàn.**

## 3. Hạn chế về hỗ trợ đa ngôn ngữ (Localization Barrier):

- 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 (Localization):** 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  $\rightarrow$  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

1. Giới thiệu

**2. Thách thức và Cơ sở lý thuyết**

3. Phương pháp đề xuất

4. Thực nghiệm và Đánh giá

5. Kết luận

# Khoảng cách hình thái học

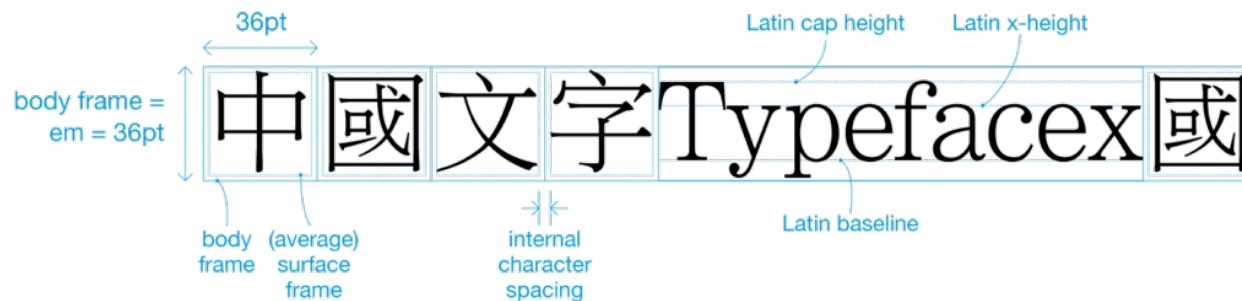
Tại sao cặp Latin - Hán tự lại là thách thức lớn nhất?

## 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: 2. Tại sao chọn Diffusion Model?

(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.

- **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.

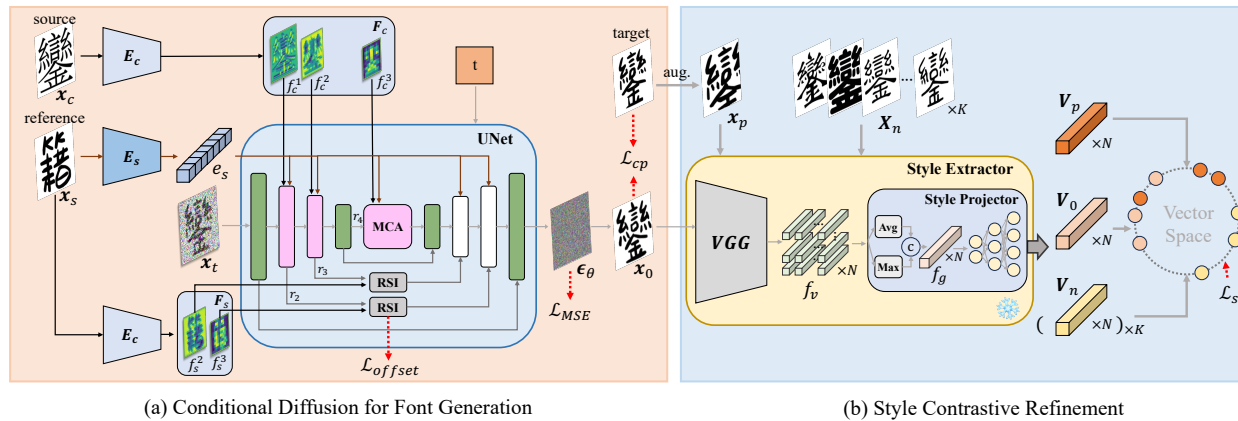
→ Đây là chìa khóa để bắc cầu qua “Morphological Gap”.

# Mục lục

1. Giới thiệu
2. Thách thức và Cơ sở lý thuyết
- 3. Phương pháp đề xuất**
4. Thực nghiệm và Đánh giá
5. Kết luận

# Tổng quan mô hình cải tiến

Kiến trúc FontDiffuser + CL-SCR Kế thừa FontDiffuser (AAAI'24) và đề xuất mô-đun mới cho đa ngôn ngữ.



## 3 Thành phần chính:

1. **MCA**: Tổng hợp nội dung đa tỷ lệ (Giữ nét).
2. **RSI**: Tương tác cấu trúc (Chống biến dạng).
3. **CL-SCR (New)**: Tinh chỉnh phong cách xuyên ngôn ngữ.



# Trọng tâm: Module CL-SCR

Cross-Lingual Style Contrastive Refinement Giải pháp cho vấn đề “Domain Gap” giữa hai ngôn ngữ.

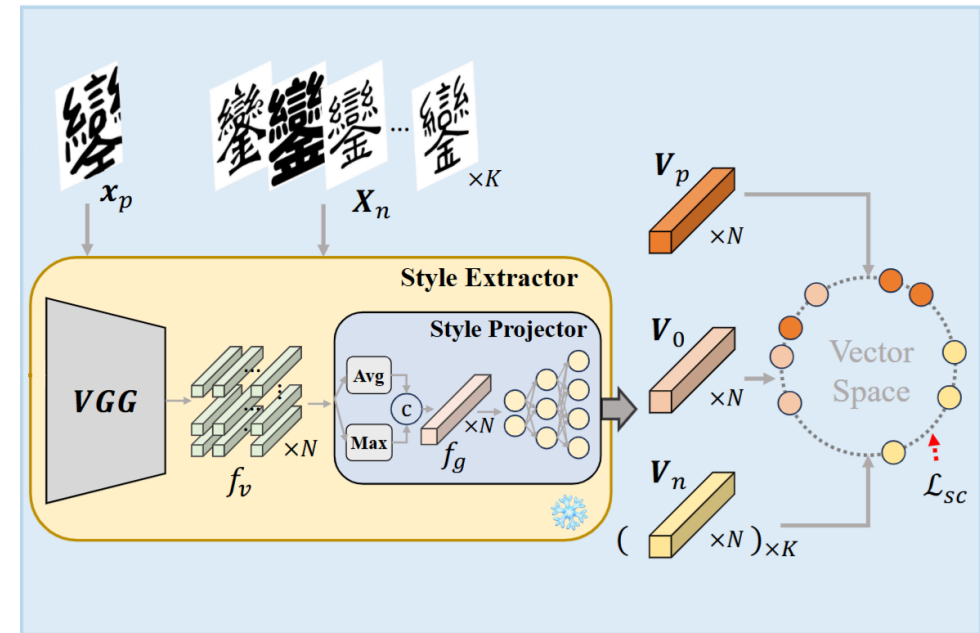
## Cải tiến cốt lõi:

- **Mở rộng mẫu âm (Negative Samples):** Lấy mẫu từ cả hai miền ngôn ngữ.

- **Chiến lược Loss hỗn hợp:**

$$L = \alpha \cdot L_{\text{intra}} + \beta \cdot L_{\text{cross}}$$

- Tăng trọng số  $\beta = 0.7$  để ép mô hình học sự tương đồng phong cách bất kể ngôn ngữ nào.



## Cơ chế tương phản đa miền

# Mục lục

1. Giới thiệu
2. Thách thức và Cơ sở lý thuyết
3. Phương pháp đề xuất
4. Thực nghiệm và Đánh giá
5. Kết luận

# Bộ dữ liệu

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

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

Latin: **52** ký tự cơ bản.

Hán tự: **800** ký tự thông dụng (GB2312).

- **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 ( $\text{lr} = 1 \times 10^{-4}$ ).
- **Mục tiêu:** Học cấu trúc nội dung và phong cách cơ bản.

## 2. Tiền huấn luyện CL-SCR:

- **Quy mô:** 200k bước, Batch 16, Adam[cite: 914].
- **Augmentation:** Random Resized Crop (Scale 0.8-1.0)  $\rightarrow$  Tăng tính bền vững với biến thể hình học.

## 3. Giai đoạn 2 - Tinh chỉnh:

- **Thiết lập:** 30k bước, Batch 4, giảm  $\text{lr} = 1 \times 10^{-5}$  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	<b>0.2008</b>	0.3048	0.3876	62.72
	FTransGAN	<b>0.1844</b>	<b>0.3900</b>	0.3548	40.45	<u>0.2089</u>	<u>0.3109</u>	0.3329	42.10
	FontDiffuser (Baseline)	0.1976	0.3775	<u>0.2968</u>	<u>14.68</u>	0.2283	0.2946	<u>0.3184</u>	<u>29.09</u>
	Ours	<u>0.1939</u>	<u>0.3890</u>	<b>0.2911</b>	<b>11.76</b>	0.2214	<b>0.3197</b>	<b>0.2954</b>	<b>13.55</b>
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	<b>0.1083</b>	<u>0.6140</u>	0.2585	40.4042	<u>0.1139</u>	<u>0.5819</u>	0.2907	75.2760
	FTransGAN	0.1381	0.5291	0.2851	55.5859	0.1456	0.4949	0.3023	88.4450
	FontDiffuser (Baseline)	<u>0.1223</u>	0.6107	<u>0.2270</u>	<u>21.2234</u>	0.1370	0.5731	<u>0.2476</u>	<u>59.5788</u>
	Ours	<b>0.1083</b>	<b>0.6406</b>	<b>0.2019</b>	<b>14.7298</b>	<b>0.1090</b>	<b>0.6377</b>	<b>0.1985</b>	<b>41.1152</b>

# Kết quả định tính

So sánh trực quan (Visual Comparison)

## Chiều Latin → Hán:

- **Ours:** Tái tạo đúng nét cọ xước, đậm nhạt.
- **Baseline:** Nét đôi khi bị cứng hoặc sai độ đậm.

## Chiều Hán → Latin:

- **Ours:** Giữ cấu trúc chữ rõ ràng.
- **DG-Font:** Bị lỗi “Content Leakage” (chữ Latin biến thành Hán).



# Kết quả định tính

Source	c	d	e	f	g	毛	毫	民	气	水
Reference										
DG-Font										
CF-Font										
DFS										
FTransGAN										
FontDiffuser (Baseline)										
Ours										
Target										

# Mục lục

1. Giới thiệu
2. Thách thức và Cơ sở lý thuyết
3. Phương pháp đề xuất
4. Thực nghiệm và Đánh giá
- 5. Kết luận**

# Tổng kết & Hướng phát triển

- ★ **Vấn đề:** Đã giải quyết bài toán chuyển đổi phong cách đa ngôn ngữ (Cross-Lingual) khó nhằn.
  - ✓ **Giải pháp:** Đề xuất mô-đun **CL-SCR** với cơ chế Loss hỗn hợp.
  - ✓ **Kết quả:** Vượt trội SOTA hiện tại (FID giảm 50% ở chiều E2C).
- 

## Hướng phát triển:

- Tối ưu tốc độ sinh ảnh (Fast Sampling/Distillation).
- Mở rộng sang tiếng Việt (Thư pháp/Quốc ngữ) và tiếng Nhật.

# Công bố liên quan

## 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 Hăng**

Lớp khoá học: **KHMT2022.1**

Khoa: **Khoa học máy tính**