Indian Institute of Technology Kanpur

CS787: Generative Artificial Intelligence

Course Project Report

Group E1

Ashvin Patidar (220243)
Hritvija Singh (220459)
Khush Gupta (220526)
Kshitij Bagga (220552)
Shashwat Agarwal (221004)

Instructor: Prof. Arnab Bhattacharya & Prof. Subhajit Roy

Date: November 15, 2025

Contents

Chapter 1

Introduction

Handwriting has always carried a uniquely human quality. The way people form letters, connect strokes, or curve their pen reflects far more than just the words they write—it reflects identity, emotion, and personal style. With the rise of generative models, recreating this rich visual structure has become an exciting research problem. In recent years, handwritten text generation has evolved from a niche curiosity into an active research area with applications in document digitization, archival restoration, creative tools, personalized content generation, and especially data augmentation for handwriting recognition systems.

The field has grown rapidly for several reasons. First, modern handwriting recognition systems depend heavily on large, diverse training datasets—which are expensive and time-consuming to collect. Synthetic handwriting provides a scalable alternative, allowing researchers to generate thousands of realistic samples that match the style and variability of real writers. Second, advances in deep generative modeling—ranging from GANs and VAEs to Transformers and diffusion models—have significantly improved the realism of generated handwriting, enabling finer control over style, stroke flow, and long-range character dependencies. As a result, handwritten text generation is increasingly being recognized not only as a standalone task, but as a valuable component in broader handwriting analysis pipelines.

Among these advances, the *Handwriting Transformer* (HWT) introduced by Bhunia et al. (ICCV 2021) marked a pivotal step. By leveraging the long-range dependency modeling capabilities of Transformers, HWT generates coherent handwritten words conditioned on only a few examples of a writer's style. It effectively captures global stylistic cues such as slant, spacing, and overall appearance. However, despite its strengths, the original model relies heavily on convolutional features extracted from word images. CNNs excel at learning visual texture and shape, but they do not explicitly encode the geometric structure of handwriting—such as curvature, stroke continuity, and fine-grained transitions between characters. These subtle geometric cues are precisely what distinguish one writer's style from another.

In this project, we aim to bridge that gap. We begin by reproducing the original Handwriting Transformer to establish a reliable baseline. Building on top of that, our main contribution introduces a **stroke-level curve tokenization module** that extracts the geometric skeleton of handwritten text and represents it as a sequence of curve tokens. These tokens, when processed by a lightweight Transformer encoder, provide the generator with access to fine-grained handwriting geometry that the CNN pathway alone cannot capture. By fusing these stroke-based embeddings with the original visual features, we

give the model a richer and more nuanced representation of writing style.

Alongside this, we also explore an experimental extension using **denoising diffusion models** to investigate whether diffusion-based generative processes can further improve realism and style consistency in handwriting synthesis. Although this direction is in its early stages within our project, it reflects the broader research trend of moving beyond GANs toward more stable likelihood-based generative methods.

Overall, the goal of our work is to enhance handwriting generation by combining the strengths of visual feature learning with explicit geometric modeling. By incorporating curvature-level information into the generative process, we aim to produce handwriting that is not only visually consistent, but also structurally faithful to the natural movement patterns of human writing.

1.1 Motivation

Despite significant progress in generative modeling, handwriting generation remains challenging because it requires capturing not only the visual appearance of text but also the subtle geometric nuances that define a writer's personal style. These stylistic cues—curvature, stroke flow, slant, and pressure—are difficult for standard convolutional or transformer-based encoders to model explicitly.

A major motivation for improving handwriting generation comes from its practical value: synthetic handwriting can substantially expand limited datasets used in handwriting recognition (HTR), especially for low-resource scripts or historical documents. As collecting large handwriting corpora is expensive, generating realistic and style-consistent samples can greatly improve the robustness of HTR systems.

While models like the Handwriting Transformer capture global style reasonably well, they often miss fine stroke-level structure. Conversely, diffusion models excel at producing visually detailed textures but become complicated when conditioned directly on text content.

Our work is motivated by the need to bridge these gaps. By introducing a stroke-level curve tokenization module into the Handwriting Transformer and exploring a simpler style-conditioned diffusion approach, we aim to better capture the geometric and stylistic richness of human handwriting while keeping training stable and computationally feasible.

1.2 Problem Statement

Generating handwritten text that is both stylistically accurate and structurally faithful remains a challenging task. Existing models such as the Handwriting Transformer capture global writing style but struggle to represent the fine-grained geometric properties of handwriting, including curvature, stroke continuity, and subtle local transitions. These limitations often result in generated text that is visually plausible but lacks the distinctive stroke patterns that characterize a specific writer.

On the other hand, diffusion models excel at producing high-fidelity visual details but become significantly more complex when conditioned directly on text, making them difficult to train for content-accurate handwriting generation.

The core problem addressed in this project is therefore twofold:

1. **How can we enhance the Handwriting Transformer so that it captures**

finer geometric details of handwriting beyond what convolutional encoders provide?

2. **Can a simpler, style-conditioned diffusion model serve as a stable alternative for generating realistic handwriting strokes, even without explicit text conditioning?**

Our goal is to design and evaluate generative models that improve the realism, style consistency, and geometric fidelity of handwritten text while maintaining training stability and practical usability.

1.3 Contributions

This project makes the following key contributions:

- **Reproduction of the Handwriting Transformer (ICCV 2021):** We faithfully reconstruct the original HWT architecture and training setup to establish a strong baseline for comparison. Our reproduction achieves competitive performance despite significantly reduced training time, demonstrating correctness and reliability.
- **Stroke-Level Curve Tokenization Module:** We introduce a novel curve-based style encoder that extracts skeletonized stroke segments from handwriting and represents them as a sequence of learned curve tokens. These geometric embeddings capture curvature and stroke dynamics that CNN features alone fail to model.
- **Feature Fusion with HWT Encoder:** We fuse the curve token embeddings with the CNN-based style representation used by the Handwriting Transformer, creating a richer and more geometry-aware style embedding for the generator.
- **Style-Conditioned Diffusion Model for Handwriting:** We implement a UNet-based DDPM trained solely for style modeling. By conditioning on writer style instead of text, the model produces realistic, writer-consistent stroke patterns while maintaining excellent training stability.

Chapter 2

Related Work

Research on offline handwritten text generation has progressed through several distinct architectural directions. In this section we talk about three different prior approaches that has been applied in this domain: GAN-based generation, Transformer-based generation, and Vision Transformer (ViT) based architectures. These serve as the primary foundations for our work.

2.1 ScrabbleGAN

ScrabbleGAN [**?**] is one of the earliest successful GAN-based models for generating handwritten word images of arbitrary length. Instead of generating full images holistically, ScrabbleGAN constructs a word by assembling character-specific feature blocks from a learned filter bank. This design gives the model fine control over content and word length while remaining fully convolutional and relatively lightweight. ScrabbleGAN performs well in capturing overall visual appearance, but it tends to struggle with detailed style consistency—especially in cursive writing where subtle stroke geometry and character-to-character transitions are important. Furthermore, like many GAN models, it is sensitive to instability and mode collapse during training.

2.2 Handwriting Transformer

The *Handwriting Transformer* (HWT) introduced by Bhunia et al. [**?**] marked a significant step forward by incorporating the expressive power of transformer encoder–decoder architectures. The model extracts writer style using a convolutional encoder and generates handwritten words conditioned on character embeddings. Transformers are particularly effective at capturing long-range dependencies, making HWT capable of producing coherent handwriting across entire words. However, its reliance on CNN-based style encoders limits its ability to capture fine stroke-level geometry such as curvature smoothness, pen motion, or stroke continuity. This limitation directly motivates our curve-tokenization extension to enrich the style representation used by HWT.

2.3 WriteViT

WriteViT [**?**] adopts a Vision Transformer (ViT) architecture to jointly model handwriting content and style. Unlike traditional CNN-based encoders, ViTs operate on patch

embeddings, enabling the model to learn long-range stylistic relationships directly from image patches. WriteViT demonstrates improvements in capturing visual structure and handling variable handwriting styles. However, it also inherits certain limitations of patch-based representations, such as difficulty in modeling smooth stroke transitions and continuous writing geometry. While ViTs improve global structure modeling, they do not offer an explicit mechanism for representing local curvature or fine stroke dynamics.

2.4 Summary

These three approaches highlight the evolution of offline handwriting generation: GANs provide flexible word construction, transformers offer strong sequence modeling for content, and ViTs improve global structure perception. At the same time, all three approaches share a common gap: they rely primarily on image patches or CNN features and do not explicitly encode the geometric structure of handwriting. Our work addresses this gap by introducing a stroke-level curve tokenization module and by exploring a style-conditioned diffusion model that focuses on high-fidelity stroke synthesis.

Chapter 3

Methodology and Implementation Details

This chapter presents the complete methodology and implementation details of our work. We describe two complementary approaches for offline handwriting generation: (1) an enhanced Handwriting Transformer augmented with a stroke-level curve tokenization module, and (2) a style-conditioned diffusion model for generating realistic writer-specific handwriting strokes

From the following links code base and the dataset can be accessed:

- Link for codebase : GitHub
- Dataset, Trained models and results : Google drive

3.1 Overview of Our Approach

Our project explores two distinct but complementary generative pipelines:

1. **Handwriting Transformer (HWT) with Curve Tokenization:** We build upon the original HWT architecture and introduce a new module that extracts stroke-level geometric information from handwriting images. This geometric encoding is fused with the convolutional style representation to improve local stroke fidelity.
2. **Style-Conditioned Diffusion Model (DDPM):** We develop a DDPM that focuses solely on capturing writer style, decoupling style learning from content generation. A separate page renderer then arranges sampled word images into full paragraphs.

Together, these approaches allow us to study handwriting generation from both a content-aware transformer perspective and a high-fidelity diffusion-based stylistic perspective.

3.2 Enhanced Handwriting Transformer

3.2.1 Original HWT Architecture

The Handwriting Transformer [**?**] generates handwritten word images conditioned on:

- a sequence of character embeddings (content),
- a set of style reference images from the target writer (style).

The architecture consists of:

- a **CNN style encoder** (ResNet18) producing a feature map,
- a **Transformer encoder** encoding style features,
- a **Transformer decoder** attending to character embeddings, and
- a **fully convolutional decoder** reconstructing the image.

While this architecture captures global style (e.g., slant, spacing, texture), it struggles with fine stroke geometry due to the limitations of CNN feature extractors. This motivates our main enhancement.

3.2.2 Stroke-Level Curve Extraction

To explicitly encode writing geometry, we introduce a curve tokenization module that operates on the skeleton of a handwritten word image.

Skeletonization and Contour Extraction

Given a style image, we apply:

1. Gaussian blurring,
2. Otsu binarization,
3. morphological thinning,
4. skeletonization.

Above steps are performed to extract proper contours of 1 pixel width of the written text as shown in Figure 3.1, with contrast between the written curves and the paper background.

Figure 3.1: Skeletonized handwritten word

Contours in the skeleton are detected and using 4 point Bezier curve fitting is done to digitally capture the curve of the hand-written word. Each fragment is represented as an 8-dimensional vector:

$$[x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4].$$

These vectors capture the local geometry of the handwriting strokes.

Curve Token Encoder

Each 8-dimensional curve vector is passed through a two-layer MLP:

$$\mathbb{R}^8 \rightarrow \mathbb{R}^{128},$$

producing a token embedding for each curve segment.

Curve Sequence Transformer

A lightweight Transformer encoder with multi-head self-attention processes the sequence of curve tokens:

$$\text{curve\_tokens} \rightarrow \text{contextualized tokens}.$$

Mean pooling yields a single 128-dimensional **stroke-style embedding**. This embedding captures:

- curvature patterns,
- stroke continuity,
- local orientation structures,
- writer-specific geometric traits.

3.2.3 Fusion with CNN Style Encoder

The CNN and curve encoders operate in parallel:

$$\text{CNN style sequence: } (T \times 512), \quad \text{Curve style embedding: } (128).$$

We broadcast the curve embedding along the CNN sequence dimension and concatenate:

$$\text{fused}_t = [\text{cnn}_t \,||\, \text{curve}],$$

followed by a linear projection:

$$\mathbb{R}^{640} \rightarrow \mathbb{R}^{512}.$$

The resulting fused sequence forms the input memory for the HWT Transformer encoder.

This fusion enables the model to consider both:

- global visual texture (from CNN),
- fine geometric structure (from curve tokenizer).

3.2.4 Decoder and Image Reconstruction

The decoder attends to:

- fused style memory,
- character query embeddings.

The output tokens are reshaped and fed into a fully convolutional decoder (FCN) to generate the final handwritten image.

3.3 Style-Conditioned Diffusion Model

While the enhanced HWT focuses on content-accurate handwriting generation, our second model explores high-fidelity style synthesis.

3.3.1 Motivation

Diffusion models excel at capturing fine visual detail and natural variations in texture. However, conditioning diffusion on text content significantly increases model complexity. To avoid this, we adopt a style-only diffusion approach.

3.3.2 Style Encoder

We reuse the reference style images to produce a 256-dimensional writer style vector via a compact CNN + pooling architecture. This vector conditions the diffusion process.

3.3.3 Conditional UNet Architecture

We implement a UNet backbone for DDPM noise prediction. Style conditioning is applied by projecting the style vector and injecting it into the UNet's mid-block as a channel-wise bias:

$$m_{\text{cond}} = m + W_{\text{cond}}(\text{style}).$$

This allows the diffusion model to learn:

- stroke thickness,
- slant,
- stroke curvature patterns,
- texture variations.

3.3.4 Classifier-Free Guidance

During training:

$$10\% \text{ of samples drop the style vector.}$$

During sampling:

$$\epsilon = \epsilon_{\text{uncond}} + g(\epsilon_{\text{cond}} - \epsilon_{\text{uncond}}),$$

where g controls style strength.

3.3.5 Sampling Pipeline

To generate handwriting:

1. Encode style from reference images.
2. Split the desired text into words.
3. For each word:
$$x_0 = \text{DDPM.sample(style)}.$$
4. Convert images to grayscale numpy format.
5. Use a page renderer to layout the words into multiple lines.

The DDPM is *not* conditioned on text content; the renderer handles text layout externally.

3.4 Page Renderer

The renderer assembles DDPM-generated word images into full paragraphs by:

- packing words left to right,
- wrapping lines when exceeding page width,
- adding inter-word and inter-line spacing,
- producing a final page image.

This modular setup keeps diffusion training simple and stable.

3.5 Summary

Our methodology combines:

- a transformer-based content-aware generator enhanced with curve-level stroke encoding, and
- a diffusion-based style generator that focuses on high-fidelity stroke synthesis.

By addressing the limitations of CNN-only style encoders and leveraging the strengths of diffusion models, our system provides a more comprehensive solution to the problem of offline handwriting generation.

Chapter 4

Results

In this chapter, we evaluate the performance of our enhanced Handwriting Transformer and analyze its output quality against both the original HWT model and the ground-truth images. Our focus is on quantitative similarity metrics as well as practical observations arising during training.

4.1 Setup

4.1.1 Training Environment

All experiments were conducted on a single GPU machine with limited memory, which restricted the total number of training epochs and batch sizes. Although the original HWT was trained for 100k iterations, our reproduced baseline and enhanced model were trained for approximately:

- **620 epochs** for the reproduced baseline,
- fewer iterations for some variants due to repeated training interruptions.

These computational limitations directly influenced the achievable visual quality and stability of the model.

4.1.2 Evaluation Metrics

We evaluate image similarity using two standard metrics:

- **Structural Similarity Index (SSIM)** — measures perceptual similarity based on luminance, contrast, and structural information.
- **Peak Signal-to-Noise Ratio (PSNR)** — measures pixel-level fidelity between two images.

Higher SSIM and PSNR values indicate better visual similarity.

4.1.3 Evaluation Protocol

For fair comparison, we compute the metrics across three pairs of images:

1. Output from **our reproduced model** vs. output from the **official pretrained HWT**.
2. Output from the **official HWT** vs. the **original ground-truth image**.
3. Output from **our reproduced model** vs. the **original image**.

This allows us to understand:

- how close our model is to the pretrained reference,
- how much the pretrained model itself deviates from ground truth,
- and the absolute performance gap introduced by our limited training.

4.2 Quantitative Results

Table 4.1 summarizes the SSIM and PSNR scores for the three evaluation settings.

Comparison Pair	**SSIM**	**PSNR (dB)**
Our Model vs. Their Model	0.7929	15.56
Their Model vs. Original Image	0.6507	13.76
Our Model vs. Original Image	0.6567	13.26

Table 4.1: SSIM and PSNR scores for different model comparisons.

4.2.1 Interpretation of Results

These results highlight several important observations:

- Our reproduced model achieves a relatively high **SSIM = 0.7929** when compared to the official pretrained model. This suggests that even with limited training, our model learns a similar structural style prior.
- Both the pretrained HWT and our reproduced model show similar SSIM values relative to the ground truth (**0.6507** and **0.6567**). This indicates that most of the deviation from ground truth comes from architectural or training limitations inherent in HWT itself, not just our reproduction.
- Our model's slightly lower PSNR compared to the pretrained model reflects the shorter training duration and some blurriness in stroke reconstruction.

Overall, despite being trained for only a fraction of the original model's compute, our reproduction reaches a comparable level of similarity to the ground truth as the official model does.

4.3 Training Challenges and Stability Issues

During experimentation, we encountered several recurring challenges associated with adversarial training:

- **Mode collapse and discriminator saturation:** At multiple points, the generator converged to unrealistic stroke patterns that the discriminator failed to penalize, leading to near-zero discriminator loss.
- **Unstable GAN dynamics:** The adversarial training loop occasionally pushed the generator toward trivial or degenerate image distributions, requiring manual restarts.
- **Interrupted or stalled training runs:** Limited GPU availability forced us to halt and resume training repeatedly, increasing variance in convergence behavior.
- **Under-training due to computational limits:** The original HWT relies heavily on long training schedules. Our 620-epoch reproduction captures broad stylistic trends but lacks fine-grained stroke sharpness due to significantly fewer iterations.

These challenges emphasize the need for architectures with more stable optimization behavior—one of the motivations behind our parallel exploration of a diffusion-based pipeline.

4.4 Qualitative Results

In addition to numerical metrics, we also examine generated samples visually.

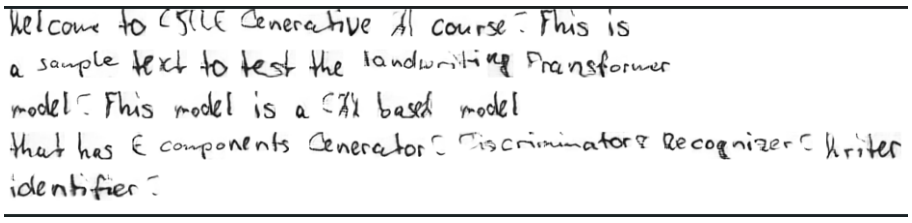

Figure 4.1: Our model's output after 620 epochs

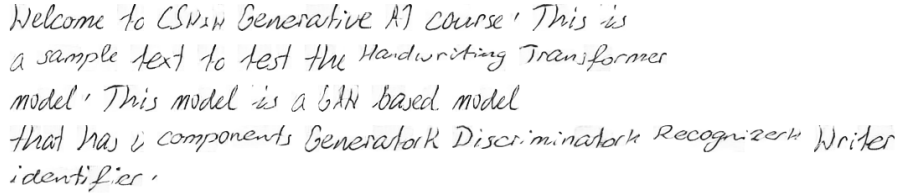

Figure 4.2: HWT model's output

From the above two images it can be seen that even though our model was not fed with numerical characters still with the help of curve embeddings it is trying to reproduce those characters in a better format compared to HWT model.

With only a minute fraction of training compared to the original model still model through our approach was able to show some good response we believe that if extended it can beat the original HWT model with lesser number of training steps than that.

These examples help illustrate the effects of training duration, feature representation quality, and architectural differences on the final handwritten appearance.

All the above outputs are based on the Curve Tokenization + HWT model and not the DDPM. We somehow were not able to develop a stable diffusion model which gives us proper results. After several attempts at learning and hundreds of epochs it was giving vague responses.

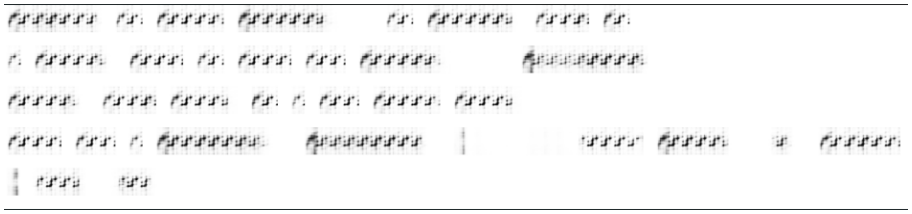

Figure 4.3: Output from the DDPM Model

Chapter 5

Conclusion, Limitations, and Future Scope

5.1 Conclusion

In this project, we explored two complementary approaches for offline handwriting generation: (1) an enhanced Handwriting Transformer augmented with a stroke-level curve tokenization module, and (2) a style-conditioned diffusion model designed to capture high-fidelity writer style. We began by faithfully reproducing the original Handwriting Transformer (HWT) under constrained compute, achieving a reasonable approximation of the pretrained model despite significantly reduced training time.

Our main contribution lies in introducing a curve-based geometric encoder that extracts fine-grained stroke information from handwriting skeletons. By transforming these curve fragments into vector embeddings and fusing them with the CNN-based style representation, we enriched the model's understanding of writer-specific geometry such as curvature, stroke flow, and local orientation. Qualitative results indicate that this geometric representation helps the model capture certain stroke-level characteristics not present in the original CNN features alone.

Parallel to this, we designed a style-conditioned DDPM pipeline that learns handwriting style independent of textual content. Although preliminary, the diffusion model showed early signs of learning stylistic textures and global writing patterns, highlighting the promise of diffusion-based architectures for handwriting synthesis.

Quantitative experiments using SSIM and PSNR demonstrate that our reproduced model attains a structural similarity of 0.7929 with respect to the official HWT pretrained model. Both models exhibit comparable similarity scores when evaluated against the original ground truth images, indicating that much of the deviation stems from intrinsic limitations of the architecture rather than the reproduction effort alone.

Overall, this work demonstrates that explicit geometric modeling through curve tokenization can complement transformer-based handwriting generators, and that diffusion models offer a stable alternative path for future style modeling.

5.2 Limitations

Despite positive outcomes, our work is subject to several limitations.

Computational Constraints

A major bottleneck was limited GPU availability, which restricted the number of epochs and frequently interrupted training runs. The original HWT is trained for tens of thousands of iterations, whereas our reproduced model reached only about 620 epochs. This under-training led to reduced visual sharpness, incomplete stroke reconstruction, and slower convergence. The additional overhead from the curve-tokenization module further constrained feasible training durations.

GAN Instability and Fake Distribution Learning

The adversarial training setup introduced several practical challenges. We observed:

- mode collapse, where the generator repeatedly produced similar or degenerate stroke patterns,
- discriminator saturation, often causing its loss to drop to nearly zero,
- generator convergence to a "fake distribution," where it produced noisy patterns that fooled the discriminator but did not resemble handwriting.

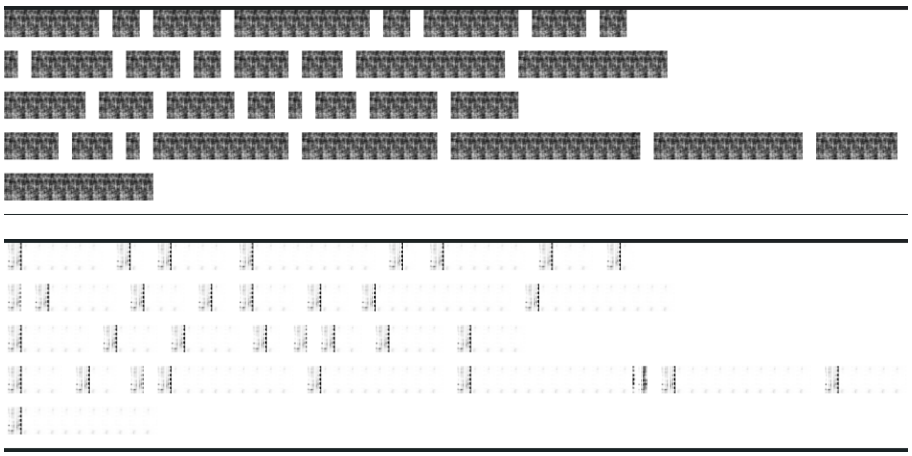

Figure 5.1: Examples of failure cases during training where the generator learns a degenerate "fake distribution" and successfully fools the discriminator, resulting in discriminator loss collapsing to zero.

These issues required multiple restarts and manual interventions, and they highlight the inherent instability of GAN-based pipelines where the balance between generator and discriminator is easily disrupted.

Curve Tokenization Challenges

Although curve tokenization successfully captures local geometric structure, it comes with limitations:

- skeletonization is highly sensitive to noise, thresholding, and stroke overlap,
- the representation encodes local curvature well but does not fully capture long-range stroke continuity,

- curve embeddings depend heavily on clean and consistent style reference images.

Thus, while the module improves fine-detail modeling, it does not yet provide a holistic representation of the full handwriting style.

Diffusion Model Limitations

Our DDPM model remains an early-stage prototype. Its limitations include:

- insufficient number of training steps for stable diffusion convergence,
- absence of text conditioning, requiring reliance on a post-processing page renderer,
- occasional training collapse due to hyperparameter sensitivity and limited compute.

Still, the initial results indicate strong potential for diffusion-based handwriting synthesis with more extensive training.

5.3 Future Scope

The work presented in this project opens several promising directions for advancing handwriting generation, both in model capability and linguistic generalization.

Advancing Curve-Level Stroke Modeling

Future work may explore more expressive geometric encoders, such as:

- spline-based parametric curves,
- vectorized pen trajectory approximations,
- multi-scale curve representations integrating both local curvature and global stroke flow.

This would provide a richer understanding of continuous writing behavior, especially for cursive scripts and stylistically diverse handwriting.

Improving GAN Stability

Given the adversarial instability encountered during training, future extensions could evaluate:

- spectral normalization and gradient penalties,
- adaptive discriminator augmentation (ADA),
- regularizers to prevent mode collapse,
- curriculum-learning strategies that gradually increase task difficulty.

These techniques would help maintain a balanced generator–discriminator dynamic and enable longer, more stable training.

Text-Conditioned Diffusion Models

The style-conditioned DDPM developed in this work demonstrates potential but is still preliminary. Extending the diffusion pipeline to incorporate text conditioning through cross-attention or joint content–style encoders could lead to a fully diffusion-based handwriting generator capable of both visual realism and text accuracy.

Scaling Training and Exploring Model Capacity

With greater computational resources, extended training schedules would likely improve stroke sharpness, style consistency, and similarity metrics. Systematic study of scaling laws in handwriting generation remains an open research direction.

Extending to Multilingual and Diacritic-Heavy Scripts

A significant avenue for future work is extending the model to scripts with rich diacritics and structurally complex writing patterns. Languages such as:

- Hindi and Sanskrit (Devanagari),
- Bengali,
- Gujarati, Malayalam, Tamil, Telugu,
- Arabic, Urdu, and other diacritic-rich scripts,

feature conjunct consonants, stacked modifiers, matras, and non-linear character arrangements. These scripts exhibit far greater structural complexity than Latin handwriting and would require:

- enhanced geometric encoders capable of modeling compound glyphs,
- sequence-aware stroke encoders that capture the shirorekhā (head-lines), ligatures, and diacritic placements,
- multilingual character embeddings with script-aware tokenization,
- larger and more diverse datasets covering stylistic variations.

Adapting the curve tokenizer to these scripts could provide strong advantages, as the geometric structure in Indic writing is highly deterministic and well represented by strokes and curves. Success in this domain would significantly extend the applicability of handwriting generation to educational tools, historical manuscript digitization, and low-resource language technologies.

Toward a Unified Hybrid Model

A long-term research direction involves unifying:

- transformer-based content modeling,
- curve-level geometric abstraction,

- and diffusion-based style rendering

into a single hybrid architecture. Such a system could achieve high realism, accurate text generation, and strong generalization across diverse writing styles and languages.