Bridging Online and Offline Handwriting via Differentiable Physical Rendering

Seonmi Park[1], Seunghyun Shin[1], Vihaan Misra[2], Dongmin Shin[3], Ukcheol Shin[4⋆], Jean Oh[2], and Hae-Gon Jeon[3⋆]

[1] Department of AI Convergence, Gwangju Institute of Science and Technology(GIST)
[2] Robotics Institute, Carnegie Mellon University
[3] Department of Artificial Intelligence, Yonsei University
[4] Energy AI, Korea Institute of Energy Technology(KENTECH)
{bluesky1000, seunghyuns98}@gm.gist.ac.kr,
{vihaanm, hyaejino}@andrew.cmu.edu,
{newdm2000, earboll}@yonsei.ac.kr, ushin@kentech.ac.kr
https://seonmip.github.io/onoff/

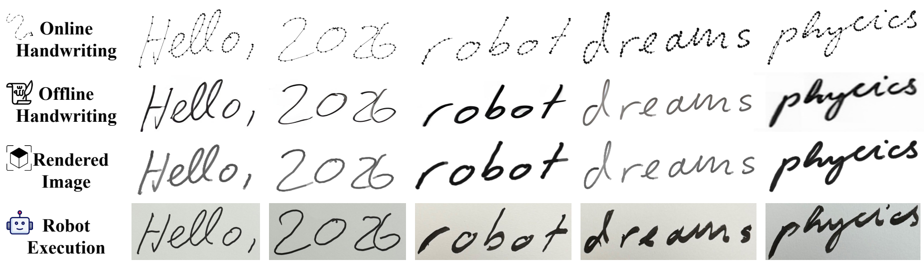


Fig. 1: A unified framework for online and offline handwriting generation. Our differentiable physical renderer connects stroke trajectories and images, enabling online handwriting trajectory estimation, offline handwriting image rendering, offline handwriting image refinement, and direct robot execution within a single framework.

Abstract. Realistic handwritten text generation plays an important role in numerous applications, such as font design, biometric authentication, and robotic calligraphy. Existing methods are typically divided into two independent paradigms: online approaches that estimate handwriting trajectories and offline approaches that synthesize realistic handwriting images. While online models capture structural and temporal dynamics, they often lack fine-grained textures, whereas offline models reproduce realistic appearance but discard stroke order. However, unifying online and offline models remains challenging due to (1) the lack of an explicit physical model linking stroke kinematics to pixel-level appearance and (2) the absence of paired trajectory–image datasets. Moreover, enabling end-to-end learning requires a differentiable rendering process across motion and appearance domains. To address these challenges, we propose a compact physical brush model that bridges stroke dynamics and visual appearance, together with a differentiable rendering module that converts stroke trajectories into stylized images. By integrating these components, we propose a unified online–offline handwriting generation framework

⋆ Corresponding author

via differentiable brush rendering. The proposed framework consists of four core modules: 1) a text-to-stroke generator that predicts the target stroke conditioned on the given text and style image, 2) a brush parameter observer that extracts brush model parameters from style references, 3) a differentiable brush renderer that maps a stroke sequence and physical brush parameters into a handwritten image, and 4) a zero-shot image refiner that refines rendered images via diffusion models. Extensive experiments and real-world robotic calligraphy demonstrations validate our approach, achieving both structural and visual fidelity.

Keywords: Online handwriting · Offline handwriting · Brush Modeling

1 Introduction

Generating realistic handwritten text is a cornerstone task in numerous practical applications, including artistic expression, graphics, security, robotics, and assistive technologies [5, 13]. For example, it enables the creation of personalized fonts, the synthesis of large-scale training data for Optical Character Recognition (OCR) systems, and the development of assistive technologies for individuals with physical impairments who cannot write by hand. More recently, it has played a key role in developing physical AI applications, such as robotic calligraphy systems that write handwritten text by a robot manipulator with a brush or pen [20]. To support these diverse requirements, handwriting generation has traditionally been studied under two independent paradigms: online and offline.

Online handwriting generation [2, 10, 12, 14, 18, 26, 29] aims to estimate a sequential trajectory, a time-ordered sequence of pen coordinates (x, y) and pen states (up/down) over time. By explicitly modeling the writing process, online methods accurately capture the underlying motion dynamics and naturally provide an executable motion plan, making them well-suited for robotic or physically grounded applications. However, such trajectory-based representations often omit fine-grained visual attributes, such as stroke width, ink bleeding, texture, or shape patterns that define a unique visual style. In contrast, offline handwriting generation [6, 9, 11, 23, 25, 32] directly synthesizes static pixel-level images. The image-based representation can effectively capture rich texture details, including ink bleeding, brush patterns, slant, and stroke thickness. Nevertheless, by operating solely in the image space, they completely discard the temporal stroke order and the dynamic writing process, restricting their applicability to purely image-level generation without physical executability.

The clear and complementary nature of the two paradigms suggests that a joint online–offline handwriting model can bring theoretical and practical expansion for the field. Integrating online and offline handwriting methods enables the model to leverage rich visual textures while benefiting from the precise structural and temporal information provided by handwriting trajectory sequences. Therefore, the joint online–offline handwriting generation model can extend the application space by simultaneously producing visually realistic images and executable trajectories, supporting both image-level tasks and physically

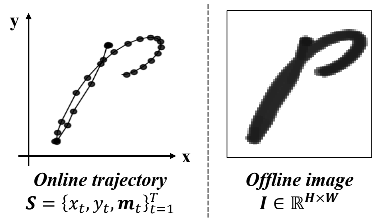


Fig. 2: Illustration of online and offline handwriting.

Table 1: Comparison of online, offline, and joint handwriting representations.

	Online	Offline	Joint
Representation	Trajectory	Image	Both
Structure info	✓	Limited	✓
Visual style	Limited	✓	✓
Physical execution	✓	×	✓
Appearance realism	Limited	✓	✓

grounded applications. Also, by learning the correspondence between online stroke kinematics and offline visual appearance, the unified framework enables cross-modal consistency, improves structural and stylistic fidelity, and provides a principled foundation for trajectory estimation from images and image synthesis from trajectories, even under partially observed or weakly paired data settings.

However, training a unified online–offline handwriting model is fundamentally challenging due to two core bottlenecks. First, there is a lack of explicit physical models that bridge stroke kinematics and pixel-level brush appearance, making the mapping between motion and texture highly under-constrained. Second, no existing dataset simultaneously provides both high-quality offline images and aligned sequential stroke trajectories, severely limiting its training process. Furthermore, to enable end-to-end joint learning and to allow gradient backpropagation across the motion and appearance domains, the physical model that bridges stroke kinematics and brush appearance must be differentiable.

To address the two problems, we first define a compact physical brush model that parameterizes brush behaviors using six core parameters, serving as a principled bridge between stroke kinematics and visual text appearance. With this parameterization, we further design a differentiable rendering module that converts sequential stroke trajectories into pixel-level images. By integrating these two key components, a physical brush model and a differentiable renderer, we propose a unified framework for joint online and offline handwriting generation. Moreover, our proposed differentiable physical renderer enables the synthesis of stylized offline images from existing online stroke datasets, significantly expanding the applicability of trajectory-only data. The main contributions of this paper, whose examples are shown in Fig. 1, are summarized as below:

- We introduce a compact physical brush model with six core parameters that explicitly connect stroke kinematics to visual text appearance, providing a principled bridge between online and offline representations.
- We design a differentiable rendering module that transforms sequential stroke trajectories into pixel-level images, enabling gradient backpropagation across motion and appearance domains.
- We propose a unified handwriting generation framework that jointly models online trajectories and offline images via differentiable physical rendering.

- Our renderer allows offline image synthesis from existing online stroke datasets, reducing dependency on paired datasets and broadening real-world utility.
- We demonstrate the superior performance of our proposed method through extensive experiments and real-world robotic calligraphy demonstrations.

2 Related Work

Our research builds upon two primary lines of work: handwriting generation, divided into online and offline methods and summarized in Tab. 1, and physical brush modeling from the field of computer graphics.

2.1 Handwriting Generation

Online Handwriting Generation. Online handwriting generation aims to estimate an online data, a sequence of pen coordinates (x_t, y_t) along with pen-up/down states, speed, or pressure information, thereby capturing the temporal dynamics of writing. It usually takes target text strings (*e.g.*, "letter") as input and outputs temporal handwriting trajectory data as shown in Fig. 2. This format is widely used for digital note-taking [2] and robotic calligraphy [20]. The seminal work [12] shows that recurrent neural networks can predict future pen positions point-by-point conditioned on target text strings. Later models, such as Deepwriting [2] and CoSE [1], introduce latent disentangled models to successfully separate handwriting style from textual content, enabling editing and style transfer in trajectory space.

Recent works have tended to focus on style-conditioned stroke sequence generation. Write Like You [29] formulates arbitrary-style online Chinese handwriting generation as a sequence-to-sequence problem with metric-based meta learning. SDT [10] further disentangles writer-wise and character-wise style representations for stylized online character generation. Elegantly Written [18] studies online handwriting enhancement by transferring fine-grained style from a small number of user traces, while Ren *et al.* [26] move beyond isolated characters and address line-level online generation by decoupling layout from glyph formation. Other works have also explored using implicit neural representations for sketches or leveraging large language models to help users write abbreviated text [3, 30].

Offline Handwriting Generation. In Fig. 2, offline handwriting generation synthesizes static pixel-level images instead of dynamic trajectories. It takes text strings (*e.g.*, "letter") and style-reference images as input and estimates styled text images. This approach is widely used to expand datasets for text recognition models [14] and to design customized fonts for individuals [32]. Early learning-based methods like GANwriting [14] and HiGAN+ [11] use Generative Adversarial Networks (GANs) to extract style features from a reference image and combine them to generate style-reflected text images. Transformer-based methods strengthen style-content fusion and generalization. HWT [6] captures both global and local style patterns with self-attention, and Pippi *et al.* [23] introduce visual archetypes to improve generalization for rare characters and unseen styles. Recently, Diffusion Models and autoregressive architectures have been proposed to generate high-quality, arbitrary-length text lines from just a single style image [9, 25]. Despite these great improvements in visual quality and diversity, offline methods cannot capture the natural stroke dynamics and writing order of human handwriting [6, 20].

2.2 Physical Brush Modeling and Stroke Rendering

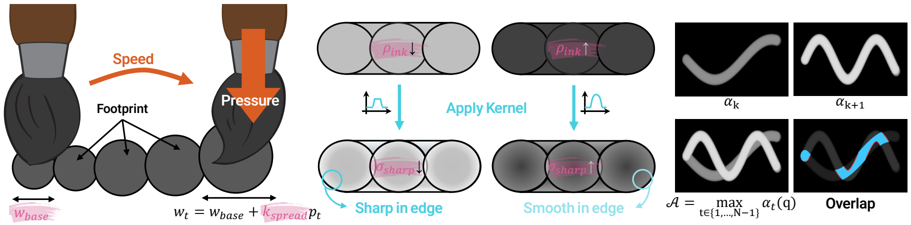
Our work is deeply inspired by classical research [4,7,8,16,28] in computer graphics that simulates the physical process of writing and drawing. These models provide a strong foundation for parameterizing the complex interactions among a writing tool, carried ink, and a paper surface. Hairy Brushes [28] decomposes this process into a brush (*i.e.*, a collection of bristles), a stroke (*i.e.*, a trajectory of position and pressure), dip (*i.e.*, ink application), and paper. In that formulation, pressure affects both bristle spreading and paper contact, while overlapping marks are composed according to the darkness of the darkest stroke. This work establishes the core idea of decoupling the stroke path from the tool's physical properties.

Subsequent research pursues higher-fidelity simulations. Lee [16] introduce soft 3D brushes whose bristle shapes vary dynamically under forces from the paper and further coupled brush rendering with paper texture and ink diffusion for oriental black-ink painting. Chu and Tai [8] propose a skeleton-surface brush model solved by a constrained energy minimization, enabling brush flattening, bristle spreading, wetted-brush plasticity, and paper resistance to be reproduced in real time. DAB [4] further develops a deformable 3D brush with a spring-mass skeleton and subdivision surface, together with bidirectional paint transfer and complex brush loading for interactive painting. Wetbrush [7] pushes this line to the bristle level by jointly simulating brush, paint, and canvas interactions through coupled dynamic simulation and liquid transfer. More recent work, InkBrush [31], utilizes simplified geometric stroke meshes and adjustable appearance parameters to produce convincing 3D ink strokes in real time.

Although classical models can simulate brush deformation and ink transport for the writing process well, they are computationally expensive and generally incompatible with gradient-based learning, as they are neither differentiable nor easily invertible. Therefore, our goal is not to create a perfect physical simulator but to design an inverse-friendly and differentiable model of the writing process.

3 Differentiable Brush Renderer $\mathcal{R}$

Our framework aims to jointly infer an online stroke sequence and generate an offline handwriting image through physically interpretable rendering parameters. We first introduce the differentiable renderer $\mathcal{R}$, a key component of the overall framework shown in Fig. 5. Given an online stroke sequence S and brush parameters θ, $\mathcal{R}$ produces a raster handwriting image I_{rend}. We present $\mathcal{R}$ first because it is used to synthesize paired online–offline training data and later serves as the image formation module in the full pipeline.



(a) Geometric Deformation **(b) Optical properties of the ink transfer** **(c) Maximum Composition**

Fig. 3: Conceptual overview of the proposed brush model. (a) w_{base} and k_{spread} determine the brush footprint geometry, with the pressure proxy bounded by p_{min} and p_{max}. (b) ρ_{ink} controls stroke intensity, while σ_{sharp} controls boundary falloff. (c) Per-step alpha maps are combined using max composition.

3.1 Brush Parameterization

Our renderer is motivated by classical physical brush models, but is not a direct implementation of them. Instead, we distill the physical properties that are most relevant to handwriting appearance into a low-dimensional differentiable surrogate. Chu and Tai [8] identify several factors that are critical for rendering visually plausible strokes. These include the footprint of the tool tip on the paper, pressure-dependent deformation, ink spreading under contact, and the state of the ink supply. Here, the footprint refers to the contact area formed between the tip and the paper surface. We compress these factors as brush model parameters:

$$\theta = \{w_{\text{base}}, k_{\text{spread}}, \rho_{\text{ink}}, \sigma_{\text{sharp}}, p_{\text{min}}, p_{\text{max}}\}, \tag{1}$$

where w_{base} denotes the base stroke width, k_{spread} controls the sensitivity of the width to applied pressure, ρ_{ink} determines the visual density of the deposited ink, and σ_{sharp} controls the boundary falloff of the stroke footprint. These four parameters control the visible stroke appearance, including stroke width, spreading, ink intensity, and edge softness. The remaining parameters, p_{min} and p_{max}, define the tool-specific dynamic range of the velocity-derived pressure proxy used in the rendering process. Since standard online handwriting datasets typically lack physical pressure measurements, we employ a kinematic proxy estimated from the stroke's drawing velocity, assuming an inverse relationship between speed and applied pressure. This abstraction deliberately omits full 3D brush dynamics, wetness diffusion, and paper absorption, retaining only factors that are visually important and identifiable from style images.

3.2 Differentiable Rendering Pipeline

Our differentiable brush renderer $\mathcal{R}$ converts an online handwriting trajectory $S = \{(\mathbf{u}_t, \mathbf{m}_t)\}_{t=1}^{N}$, where $\mathbf{u}_t = (x_t, y_t) \in \mathbb{R}^2$ is the 2D pen position and $\mathbf{m}_t \in \{0,1\}^3$ denotes the one-hot pen-state indicating pen-down, pen-up, and end-of-stroke at time step t, with θ into a rendered style-reflected image I_{rend}. The whole pipeline is depicted in Fig. 3.

Pressure proxy. Since online handwriting datasets do not provide pen pressure, we approximate it from the local writing speed. Specifically, for the t-th stroke segment connecting consecutive pen positions $\mathbf{u}_t$ and $\mathbf{u}_{t+1}$, we compute the local writing speed as

$$v_t = \|\mathbf{u}_{t+1} - \mathbf{u}_t\|_2, \qquad t = 1, \ldots, N-1. \tag{2}$$

and convert its stroke-wise normalized version $\tilde{v}_t \in [0, 1]$ into an instantaneous pressure proxy:

$$p_t^{\text{proxy}} = p_{\min} + (p_{\max} - p_{\min})(1 - \tilde{v}_t), \tag{3}$$

where $p_{\min}, p_{\max} \in \theta$ are predicted lower and upper bounds that capture the physical response range of the writing tool. To obtain a temporally consistent pressure signal for rendering, we apply exponential smoothing:

$$p_t = (1 - \beta)p_{t-1} + \beta p_t^{\text{proxy}}, \tag{4}$$

where $\beta \in (0, 1)$ controls the smoothing strength. We can observe that pressure decreases as speed increases.

Footprint and ink transfer. Classical brush models represent a stroke as a trajectory of position and pressure whose footprint varies with pressure. Hairy Brushes further notes that increasing pressure affects both spreading and paper contact, and that spreading is taken to be linearly proportional to pressure by default [28]. We therefore approximate the effective stroke width as follows:

$$w_t = w_{\text{base}} + k_{\text{spread}} p_t. \tag{5}$$

For the segment $(\mathbf{u}_t, \mathbf{u}_{t+1})$, let $d_t(q)$ be the shortest Euclidean distance from pixel q to the segment. We define the normalized distance as:

$$\hat{d}_t(q) = \frac{d_t(q)}{w_t/2}, \tag{6}$$

and a footprint kernel as:

$$K_t(q) = \max\Big(0, 1 - \hat{d}_t(q)^{\gamma}\Big) \quad \text{s.t.} \quad \gamma = \gamma_{min} + \lambda(1 - \sigma_{\text{sharp}}), \tag{7}$$

where γ controls how sharply the response decays away from the stroke centerline. Visible ink transfer is modeled as follows:

$$O_t = 1 - \exp(-\rho_{\text{ink}} p_t), \qquad \alpha_t(q) = s_t \, K_t(q) \, O_t, \tag{8}$$

where O_t is the opacity transferred at rendering step t, $s_t = m_{t,\text{down}} \in \{0, 1\}$ is the pen-down indicator extracted from the one-hot pen state $\mathbf{m}_t$, and $\alpha_t(q)$ denotes the ink contribution of the t-th stroke segment at pixel q.

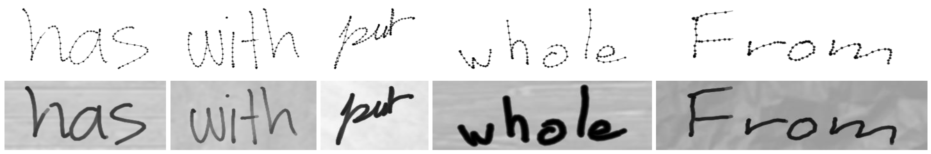
Rendering and composition. The renderer outputs a continuous coverage map as below:

$$\mathcal{A}(q) = \max_{t \in \{1, \ldots, N-1\}} \alpha_t(q), \tag{9}$$

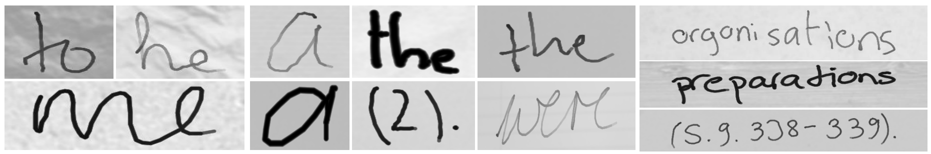
and the grayscale rendering is defined as:

$$I_{\text{rend}}(q) = 1 - \mathcal{A}(q), \tag{10}$$

where $\mathcal{A}$ denotes the accumulated coverage. We use a maximum composition rather than additive blending to avoid over-darkening in self-overlapping strokes and to obtain a stable differentiable surrogate for brush-mark accumulation.



(a) Online stroke trajectories (top), synthetic offline image generated by our renderer (bottom).



(b) Additional generated samples showing diverse backgrounds, styles, and word lengths

Fig. 4: Online-Offline paired dataset generated by our differentiable renderer.

3.3 Online-Offline Paired Data Construction via Brush Rendering

A primary challenge in joint online–offline handwriting modeling is the scarcity of paired datasets that provide both raster handwriting images and their corresponding stroke trajectories. Existing online datasets only contain trajectory sequences, while offline datasets provide static images without any stroke dynamic. To overcome this limitation, we construct a realistic paired synthetic dataset $\mathcal{D}_{\mathrm{syn}}$ from an online handwriting dataset with our differentiable brush renderer.

Let $\mathcal{D}_{\mathrm{on}} = \{(T, S)\}$ denote an online handwriting dataset consisting of text strings T and their corresponding stroke trajectories S. For each data sample, we randomly assign a set of physical brush parameters θ from our predefined distributions. Using the coverage computation of $\mathcal{R}$, the online stroke sequence S and brush parameters θ are converted into a continuous coverage map $\mathcal{A}$, which represents the spatial distribution of deposited ink.

To generate more realistic offline handwriting images, we additionally incorporate a background composition strategy inspired by Emuru [25]. We curate a collection of real-world background textures $\mathcal{B}$, including notebook paper, cardboard surfaces, and wooden boards. For each synthesis, a background patch B is randomly sampled from $\mathcal{B}$ with random location and orientation. The final synthetic image I_{syn} is then obtained by compositing the ink onto the sampled background as follows:

$$I_{\mathrm{syn}} = \mathcal{A} \cdot C_{\mathrm{ink}} + (1 - \mathcal{A}) \cdot B, \tag{11}$$

where C_{ink} denotes the ink color (e.g., black). This composition introduces realistic background textures and appearance variations commonly observed in offline handwriting images, allowing the model to learn complex intensity variations and noise inherent in real-world offline images.

The resulting dataset $\mathcal{D}_{\mathrm{syn}} = \{(T, S, I_{\mathrm{syn}}, \theta)\}$ provides paired supervision across both online and offline domains: (i) (I_{syn}, S) enables the stroke generator G to learn trajectory structures conditioned on visual style, and (ii) $(I_{\mathrm{syn}}, \theta)$ allows the brush parameter observer O to recover physically interpretable rendering parameters from appearance. The examples are shown in Fig. 4.

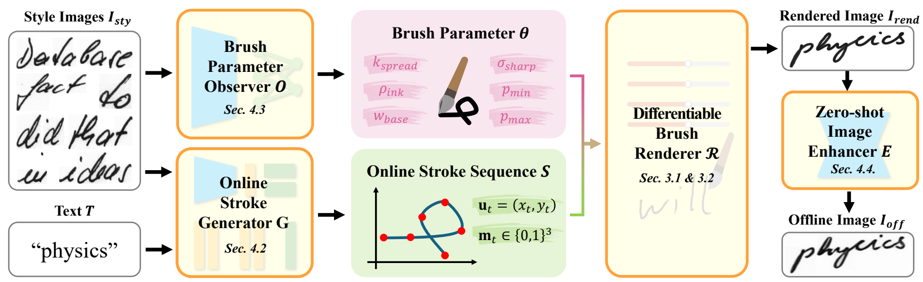


Fig. 5: Overview of the proposed joint online–offline handwriting framework. Given style reference images I_{sty} and a target text T, the brush parameter observer O estimates physics-aware brush parameters θ, while the stroke generator G predicts an online stroke sequence S. A differentiable brush renderer $\mathcal{R}$ converts the online stroke sequence and brush parameters into a rendered handwriting image I_{rend}. Finally, a zero-shot image enhancer E refines the rendered result to produce the final offline handwriting image I_{off} with a realistic appearance.

4 Unified Online and Offline Handwriting Framework

4.1 Framework Overview

Given a set of handwriting style reference images I_{sty} and a target text string $T = \{c_i\}_{i=1}^{L}$ where c_i is a character, the brush parameter observer O first estimates a set of physical brush parameters θ, which encode the physical characteristics of the writing style, from the style references I_{sty}. Conditioned on the text input T and style references I_{sty}, the stroke generator G predicts an online stroke sequence $S = \{(\mathbf{u}_t, \mathbf{m}_t)\}_{t=1}^{N}$. After that, the renderer $\mathcal{R}$ converts a stroke sequence S and brush parameters θ to a rendered handwriting image I_{rend} in a differentiable manner. Lastly, the image enhancer E further refines the rendered output I_{rend} via off-the-shelf diffusion models, which is optional.

Therefore, the overall inference process can be factorized as follows:

$$S = G(I_{sty}, T),\ \theta = O(I_{sty}),\ I_{\text{rend}} = \mathcal{R}(S, \theta),\ I_{off} = E(I_{\text{rend}}, I_{sty}, T), \quad (12)$$

where given a set of style references I_{sty} and a target text string T, our framework can estimate (i) an online stroke sequence S, (ii) a rendered image I_{rend} and (iii) a refined image I_{off}.

4.2 Online Stroke Generator G

The role of the online stroke generator G is to produce a temporally ordered trajectory S that defines the structural layout of the handwritten word. To account for this sequential writing process, we build G upon the transformer-based trajectory estimation architecture [10]. However, the original model [10] is designed for single-character synthesis, which limits its applicability to practical handwriting generation scenarios where words or full text lines are typically produced. We thus extend it to operate at the word level (*i.e.*, multiple characters) rather than single characters.

Specifically, we replace the original single-character content token with a sequence of character embeddings derived from the target text string $T = \{c_i\}_{i=1}^{L}$, enabling the model to process multi-character inputs. To capture contextual dependencies across characters, we introduce an additional cross-attention layer in the decoder. This layer allows the trajectory representation to dynamically attend to the entire character sequence during generation.

With the above modifications, the generator G operates at the word level and predicts the corresponding stroke sequence $S = \{(\mathbf{u}_t, \mathbf{m}_t)\}_{t=1}^{N}$ from the target text string T and style images I_{sty} as follows:

$$S = G(T, I_{sty}). \tag{13}$$

Loss function. The stroke generator G predicts an online stroke sequence autoregressively. At each time step, it outputs a K-component 2D Gaussian mixture for the pen position and logits for the pen state, where $K = 20$. We train G using the negative log-likelihood of the ground-truth pen position under the predicted mixture distribution and a cross-entropy loss for the pen state:

$$\mathcal{L}_G = -\sum_t \log p(\mathbf{u}_t) + \lambda_{\text{pen}} \sum_t \text{CE}(\hat{\mathbf{m}}_t, \mathbf{m}_t), \tag{14}$$

where $p(\mathbf{u}_t)$ is the predicted Gaussian mixture density, $\hat{\mathbf{m}}_t$ denotes the predicted pen-state logits, and $\mathbf{m}_t$ denotes the ground-truth pen state. Further details are provided in the Supplementary Material.

4.3 Brush Parameter Observer O

The brush parameter observer O predicts brush parameters θ, which controls how an online stroke sequence is rendered into a stylistic image. Given style reference images I_{sty}, the observer estimates the corresponding parameters as:

$$\theta = O(I_{sty}). \tag{15}$$

The estimated parameters θ can provide two functionalities: controllability and explainability. First, they control the renderer $\mathcal{R}$ to generate a style-reflected image with the handwriting trajectory. Second, they provide an interpretable description of the writing style that can be exploited for downstream systems, such as robotic calligraphy.

Architecture. To capture both image-specific cues and the shared writing style, O utilizes DINOv3 [27] to extract features from the reference images. The extracted features are utilized as tokens, with a learnable style token added as the first token for each reference image. The concatenated tokens are processed by an aggregator consisting of Transformer layers. The aggregator combines two approaches: intra-image self-attention, which captures local stylistic details within each individual image, and inter-image attention, which extracts a consistent shared writing style across multiple style references. The final representation of the style token is then projected through an MLP head to predict the normalized brush parameters.

Loss function. We optimize O using parameter-space supervision and visual rendering consistency through our differentiable brush renderer $\mathcal{R}$:

$$\mathcal{L} = \mathcal{L}_{params} + \mathcal{L}_{render}.$$

The loss functions $\mathcal{L}_{params} = \mathbf{MSE}(\theta, \hat{\theta})$ and $\mathcal{L}_{render} = \mathbf{MSE}(I_{rend}, \hat{I}_{rend})$ ensure direct alignment and accurate reconstruction of the final visual appearance. Here, $\hat{I}_{\text{rend}} = \mathcal{R}(S, \hat{\theta})$ and $I_{\text{rend}} = \mathcal{R}(S, \theta)$.

4.4 Zero-shot Offline Image Enhancer E

The rendered image I_{rend} effectively captures the desired word structure and a style-reflected brush footprint. However, several real-world handwriting effects, commonly observed in offline images, remain beyond the scope of the renderer, such as paper texture, scanner artifacts, local intensity fluctuations and dataset-specific appearance statistics. To mitigate this gap, we employ an off-the-shelf handwriting diffusion model [9, 22] as a zero-shot image enhancer E.

A key design choice is to keep the diffusion model completely unchanged. Rather than retraining the model or modifying its architecture, we utilize the rendered image I_{rend} as a structural prior during the diffusion sampling process. This strategy allows us to exploit a pretrained handwriting diffusion model in a zero-shot manner while preserving the stroke structure predicted from our framework.

Let $\mathcal{E}$ denote the VAE encoder of the diffusion model. We first encode the rendered image into the latent space to obtain a structural prior z_{rend}:

$$z_{\text{rend}} = \mathcal{E}(I_{\text{rend}}). \tag{16}$$

At a chosen intermediate timestep t_0, we initialize the diffusion latent as:

$$z_{t_0} = \sqrt{\bar{\alpha}_{t_0}}\, z_{\text{rend}} + \sqrt{1 - \bar{\alpha}_{t_0}}\, \epsilon \quad \text{s.t.} \quad \epsilon \sim \mathcal{N}(0, I), \tag{17}$$

and then run the standard conditional denoising process conditioned on (T, I_{sty}) to estimate the denoised latent z_0. Starting the diffusion process from a noised version of the rendered structure rather than pure noise preserves the generated layout while allowing the model to refine the appearance. Lastly, the refined offline handwriting image I_{off} is obtained by decoding the denoised latent z_0 with the VAE decoder $\mathcal{D}$:

$$I_{off} = \mathcal{D}(z_0). \tag{18}$$

5 Experiments

Here, we describe the training and evaluation datasets used for our frameworks. Please refer to the Supplementary Material for implementation details, evaluation metrics, and further experimental settings.

Training: online-offline paired dataset. For training our proposed framework, we construct an online–offline paired dataset by combining IAM-OnDB [19] and

CASIA-OLHWDB [17] with the data construction pipeline that utilizes our differentiable renderer (*i.e.*, Sec. 3.3). For each stroke sequence S and text string T, we sample brush parameters θ and render a corresponding handwriting image I_{syn}, forming synthetic pairs $(T, S, I_{\mathrm{syn}}, \theta)$ used for training the stroke generator and brush parameter observer. The resulting dataset contains 155,840 word-level samples, split by writer into 303 writers for training and 78 for testing (*i.e.*, online handwriting evaluation).

Evaluation: Online handwriting dataset. We utilize IAM-OnDB [19] and CASIA-OLHWDB (1.0–1.2) [17] to generate a paired dataset and evaluate online handwriting results. Each trajectory contains on average $\sim$60 points and is truncated to a maximum length of 250 points. The maximum word length is 9 characters with an average of 3.5, and the dataset includes 25,912 unique word types. Since IAM-OnDB provides line-level annotations, we segment each line into word-level samples using the corresponding labels and retain only words verified through an OCR-based filtering step (see Supplementary Material for details).

Evaluation: Offline handwriting datasets. For offline handwriting evaluation, we use the test sets of IAM [21] and CVL [15], containing 161 and 283 writers respectively. Words longer than 9 characters are discarded. All rendered images are resized to a fixed height of 64 pixels while preserving the original aspect ratio.

5.1 Evaluation Results

Quantitative Comparison For evaluation, we utilize a test set comprising IAM-OnDB [21] and CASIA [17], along with their rendered images. The performance is assessed in both single-letter and multi-letter scenarios. In the multi-letter setting, since SDT [10] is designed for single-letter online handwriting, we perform inference on individual characters and concatenate them for a fair comparison. As shown in table 2, our stroke generator outperforms the state-of-the-art single-character model in terms of quality, demonstrating the robustness and scalability of our proposed method

In table 3, comparative analysis reveals that standard offline handwriting models struggle to generalize when conditioned on our paired dataset, primarily due to an over-reliance on pixel-level distributions. Since our dataset utilizes real human trajectories, the failure of these baselines highlights their inability to capture the underlying motion. Our model overcomes this by explicitly decoupling and modeling the stroke-level physics; the performance gains confirm that our Stroke Generator effectively bridges the gap between digital trajectories and realistic image synthesis.

For Ours variants, the noise-injection start step is tuned for each baseline–dataset setting: 50 for DiffPen+Ours on both IAM and CVL, 900 for One-DM+Ours on IAM, and 200 for One-DM+Ours on CVL. Table 4 summarizes the quantitative comparison on IAM Words and CVL Words. Among the baseline models, One-DM [9] shows the strongest overall performance, achieving the best FID, BFID, and CER on both datasets, while DiffPen [22] yields the lowest

Table 2: Evaluation of online handwriting models in two scenarios.

Model	DTW ↓ multi-letter	DTW ↓ single-letter
SDT	0.8155	0.6056
Ours	**0.2936**	**0.3462**

Table 3: Evaluation on our On-Offline Paired Dataset. **Bold**: Best, Underline: Second-best.

Model	FID ↓	BFID ↓	HWD ↓	CER ↓	LPIPS ↓
VATr++	74.28	33.44	2.5990	<u>0.329</u>	0.622
DiffPen	89.06	118.04	<u>2.0030</u>	0.496	0.602
One-DM	<u>52.26</u>	<u>19.88</u>	2.3650	**0.229**	<u>0.539</u>
Emuru	89.97	60.88	2.6220	6.126	0.583
Our Renderer	**11.81**	**11.03**	**1.0833**	0.361	**0.001**

Table 4: Quantitative evaluation for baseline models and our framework on IAM and CVL datasets. **Bold**: Best, Underline: Second-best.

Dataset	IAM Words					CVL Words				
Model	FID↓	BFID↓	HWD↓	CER↓	LPIPS↓	FID↓	BFID↓	HWD↓	CER↓	LPIPS↓
VATr++	52.57	34.17	2.1668	0.3900	0.4465	25.11	20.12	2.2394	0.4775	0.4034
DiffPen	77.70	151.96	**1.6275**	0.6672	0.5380	46.31	108.21	<u>1.6057</u>	0.6056	0.4680
One-DM	<u>29.00</u>	<u>16.35</u>	1.8306	**0.3117**	<u>0.3726</u>	<u>19.45</u>	<u>15.18</u>	2.1500	**0.2877**	<u>0.3898</u>
Emuru	89.89	72.64	2.2464	3.0583	0.5453	68.06	54.52	2.2366	3.5744	0.4816
DiffPen+Ours	76.86	97.19	2.3116	0.7103	0.5155	30.68	64.99	2.1506	<u>0.3918</u>	0.4485
One-DM+Ours	**25.89**	**8.15**	<u>1.6442</u>	**0.3117**	**0.3288**	**14.45**	**10.47**	**1.5264**	0.6285	**0.3441**

HWD [24] on both datasets. By contrast, Emuru [25] performs noticeably worse across most metrics.

Across two diffusion baselines and two datasets, attaching our renderer consistently improves FID and BFID, showing that the proposed prerender-guided initialization improves visual realism. Other metrics, such as HWD and CER, show backbone- and dataset-dependent trade-offs, indicating that the optimal noise-injection start step should be tuned for each setting rather than fixed globally. Overall, our framework acts as a complementary guidance module whose effect depends on the backbone model, primarily improving realism-oriented metrics while improving handwriting-style consistency in some settings.

Qualitative Results. We present results from our method and competing methods in Fig. 6. Our approach combines a differentiable stroke renderer with a diffusion-based sampling process. The renderer provides physically grounded guidance that preserves style-relevant attributes such as stroke thickness and ink density, while the diffusion model enhances visual realism and texture. As a result, the generated samples remain faithful to the underlying writer-specific stroke characteristics while maintaining realistic appearance.

We further conduct a ranking-based user study for perceptual style similarity. One-DM+Ours obtains the best average rank of 2.840, improving over One-DM with 3.406. Detailed protocols and full results are provided in the Supplementary Material.

5.2 Application: Robotic Writing Demonstration

To demonstrate the physical applicability of our framework, we conduct a robotic writing demonstration using the predicted stroke trajectories S. Given a reference

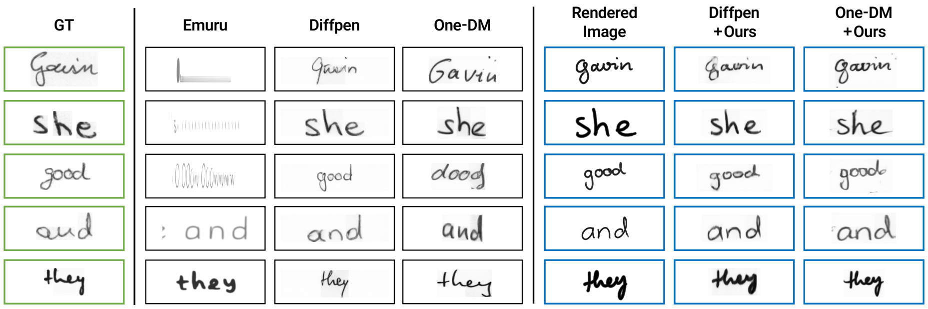


Fig. 6: Qualitative comparison of offline handwriting generation results. From left to right: ground-truth images, existing offline handwriting generation methods, and our results (*i.e.*, rendered result + zero-shot enhancer E). While previous methods often fail to preserve the correct word structure or produce unstable strokes, combining them with our method improves structural consistency and visual realism.

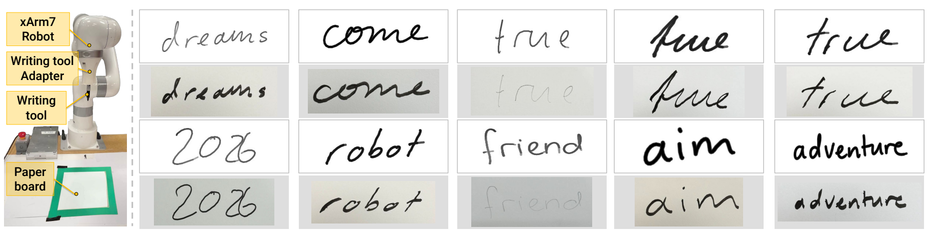


Fig. 7: Qualitative results of our framework executed by a real robotic system. For each word, the upper row shows the offline handwriting image rendered by our differentiable brush renderer, while the lower row shows the corresponding robotic writing obtained by executing the predicted stroke trajectory. These examples demonstrate that our method generates visually consistent handwriting while producing physically executable trajectories for robotic manipulation.

handwriting style image, our model first estimates the corresponding stroke parameters θ, which characterizes properties such as base stroke width and ink density. Based on these predicted parameters, we select a physical writing tool (*i.e.*, pencil, sharpie, marker pen, etc) whose tip characteristics best match the estimated stroke footprint and ink deposition behavior. A writing tool is fixed to the robot end-effector (*i.e.*, 0-DoF rotation), and the generated trajectory S is directly executed as Cartesian motions. The 2D stroke coordinates from S define the (x, y) path, while the pressure proxy is mapped to the vertical position z, controlling the contact depth between the pen and the surface. This simple mapping enables the robot to reproduce the generated handwriting without additional optimization or post-processing, showing that our framework can be directly deployed in robotic writing applications, as shown in Fig 7.

6 Conclusion

We introduce a compact physics-aware brush parameterization and a differentiable brush renderer that connects stroke trajectories to pixel-level handwriting images. With this formulation, we develop a unified framework that bridges online and offline handwriting generation, consisting of a differentiable brush renderer, an online stroke generator, a brush parameter observer and a zero-shot offline image enhancer. Extensive experiments demonstrate that our method can generate visually consistent handwriting images while producing physically executable trajectories that can be directly deployed on a robotic writing system.
Limitations. Although the proposed brush parameters are physically interpretable, several of them are renderer-level surrogate parameters rather than quantities directly calibrated to real-world physical units. Therefore, transferring the predicted parameters to physical writing tools may require additional calibration. Furthermore, the current architecture is optimized for word-level generation. Extending the framework to coherent long-form sentence generation remains an important direction for future work.

Acknowledgements

This work was supported by GIST-IREF from Gwangju Institute of Science and Technology(GIST), Korea Institute of Science and Technology (KIST) Institutional Program (26K0030) and the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean Government (MSIT) (No. RS-2024-00457882 and No.RS-2025-25441838) and the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT)(RS-2024-00338439).

References

1. Aksan, E., Deselaers, T., Tagliasacchi, A., Hilliges, O.: Cose: Compositional stroke embeddings. Proceedings of the Neural Information Processing Systems (NeurIPS) **33**, 10041–10052 (2020)
2. Aksan, E., Pece, F., Hilliges, O.: Deepwriting: Making digital ink editable via deep generative modeling. In: Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI) (2018)
3. Bandyopadhyay, H., Bhunia, A.K., Chowdhury, P.N., Sain, A., Xiang, T., Hospedales, T., Song, Y.Z.: Sketchinr: A first look into sketches as implicit neural representations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2024)
4. Baxter, B., Scheib, V., Lin, M.C., Manocha, D.: Dab: Interactive haptic painting with 3d virtual brushes. In: Proceedings of ACM SIGGRAPH (2001)
5. Berio, D., Leymarie, F.F., Asente, P., Echevarria, J.: Strokestyles: Stroke-based segmentation and stylization of fonts. ACM Transactions on Graphics (TOG) (2022)
6. Bhunia, A.K., Khan, S., Cholakkal, H., Anwer, R.M., Khan, F.S., Shah, M.: Handwriting transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021)

7. Chen, Z., Kim, B., Ito, D., Wang, H.: Wetbrush: Gpu-based 3d painting simulation at the bristle level. ACM Transactions on Graphics (TOG) (2015)
8. Chu, N.H., Tai, C.L.: An efficient brush model for physically-based 3d painting. In: 10th Pacific Conference on Computer Graphics and Applications, 2002. Proceedings. (2002)
9. Dai, G., Zhang, Y., Ke, Q., Guo, Q., Huang, S.: One-dm: One-shot diffusion mimicker for handwritten text generation. In: Proceedings of the European Conference on Computer Vision (ECCV) (2024)
10. Dai, G., Zhang, Y., Wang, Q., Du, Q., Yu, Z., Liu, Z., Huang, S.: Disentangling writer and character styles for handwriting generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
11. Gan, J., Wang, W., Leng, J., Gao, X.: Higan+: handwriting imitation gan with disentangled representations. ACM Transactions on Graphics (TOG) (2022)
12. Graves, A.: Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850 (2013)
13. Hou, Z., Yu, C., Hsu, D., Yu, H.: Teachingbot: Robot teacher for human handwriting. IEEE Robotics and Automation Letters (RA-L) (2026)
14. Kang, L., Riba, P., Wang, Y., Rusinol, M., Fornés, A., Villegas, M.: Ganwriting: content-conditioned generation of styled handwritten word images. In: Proceedings of the European Conference on Computer Vision (ECCV) (2020)
15. Kleber, F., Fiel, S., Diem, M., Sablatnig, R.: Cvl-database: An off-line database for writer retrieval, writer identification and word spotting. In: 2013 12th international conference on document analysis and recognition (2013)
16. Lee, J.: Simulating oriental black-ink painting. IEEE Computer Graphics and Applications (1999)
17. Liu, C.L., Yin, F., Wang, D.H., Wang, Q.F.: Casia online and offline chinese handwriting databases. In: International Conference on Document Analysis and Recognition. pp. 37–41. IEEE (2011)
18. Liu, Y., Khalid, F.B., Wang, L., Zhang, Y., Wang, C.: Elegantly written: Disentangling writer and character styles for enhancing online chinese handwriting. In: Proceedings of the European Conference on Computer Vision (ECCV) (2024)
19. Liwicki, M., Bunke, H.: Iam-ondb-an on-line english sentence database acquired from handwritten text on a whiteboard. In: Eighth International Conference on Document Analysis and Recognition (ICDAR'05) (2005)
20. Luo, Y., Wu, Z., Lian, Z.: Callirewrite: Recovering handwriting behaviors from calligraphy images without supervision. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) (2024)
21. Marti, U.V., Bunke, H.: The iam-database: an english sentence database for offline handwriting recognition. International journal on document analysis and recognition (2002)
22. Nikolaidou, K., Retsinas, G., Sfikas, G., Liwicki, M.: Diffusionpen: towards controlling the style of handwritten text generation. In: Proceedings of the European Conference on Computer Vision (ECCV) (2024)
23. Pippi, V., Cascianelli, S., Cucchiara, R.: Handwritten text generation from visual archetypes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
24. Pippi, V., Quattrini, F., Cascianelli, S., Cucchiara, R.: Hwd: A novel evaluation score for styled handwritten text generation. arXiv preprint arXiv:2310.20316 (2023)

25. Pippi, V., Quattrini, F., Cascianelli, S., Tonioni, A., Cucchiara, R.: Zero-shot styled text image generation, but make it autoregressive. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2025)
26. Ren, M.S., Zhang, Y.M., Chen, Y.: Decoupling layout from glyph in online chinese handwriting generation. Proceedings of the International Conference on Learning Representations (ICLR) (2025)
27. Siméoni, O., Vo, H.V., Seitzer, M., Baldassarre, F., Oquab, M., Jose, C., Khalidov, V., Szafraniec, M., Yi, S., Ramamonjisoa, M., Massa, F., Haziza, D., Wehrstedt, L., Wang, J., Darcet, T., Moutakanni, T., Sentana, L., Roberts, C., Vedaldi, A., Tolan, J., Brandt, J., Couprie, C., Mairal, J., Jégou, H., Labatut, P., Bojanowski, P.: DINOv3 (2025), https://arxiv.org/abs/2508.10104
28. Strassmann, S.: Hairy brushes. ACM Siggraph Computer Graphics (1986)
29. Tang, S., Lian, Z.: Write like you: Synthesizing your cursive online chinese handwriting via metric-based meta learning. In: Computer Graphics Forum (CGF) (2021)
30. Xu, Z., Cai, S., Varma T, M., Venugopalan, S., Zhai, S.: Skipwriter: Llm-powered abbreviated writing on tablets. In: Proceedings of the ACM Symposium on User Interface Software and Technology (UIST) (2024)
31. Yao, Z., Sun, Q., Liu, B., Lu, Y., Liu, G., Yang, X.D., Mi, H.: Inkbrush: A sketching tool for 3d ink painting. In: Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI) (2024)
32. Zhu, Y., Li, Z., Wang, T., He, M., Yao, C.: Conditional text image generation with diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023)