Gracefully Air-Written: Enhancing the Legibility and Style Consistency of In-Air Handwriting

Yu Liu[1,2], Cunrui Wang[1*], Lin Feng[1], Jianxin Zhang[1], Bo Lu[1]

[1]Dalian Chinese Font Design Technology Innovation Center, Dalin Minzu University, 116600, Dalian, China.
[2]Faculty of Computer Science and Information Technology, University Putra Malaysia , 43400 UPM Serdang,Malaysia.
ethanliuyu@foxmail.com, wcr@dlnu.edu.cn, fenglin@dlut.edu.cn, jxzhang@dlnu.edu.cn, lubo@dlnu.edu.cn

Abstract

Space computing devices expand handwritten input from two-dimensional screens into three-dimensional space, providing an unrestricted interactive experience. Due to the high degree of freedom and lack of tactile feedback in in-air handwriting, handwritten characters not only become less legible but also lose the writer's personal style. This paper proposes a method for reconstructing discrete in-air handwriting using continuous diffusion models, capturing the writing process and style from a small number of user-provided handwritten tracks and images, to restore the legibility of characters and mimics the writer's style. We represent handwritten track data in binary form and model it with continuous diffusion models, recovering discrete handwritten track data through threshold processing. Our approach reconstructs in-air handwritten characters in two stages. During the content preservation phase, we propose a partial noise injection strategy based on reference sequence modeling, using the content information of the original character as a guiding condition to maintain content consistency in handwritten character. In the style aggregation phase, we adaptively fuse the visual style of the handwritten in the image modality with the dynamic writing process in the sequence modality, overcoming issues of insufficient style capture due to noise interference in the backward process. Qualitative and quantitative experiments demonstrate the superiority of our method.

Code —
https://github.com/ethanliuyu/GracefullyAirWritten

Introduction

Handwritten are a unique means of conveying information and personal expression. With the growing prevalence of virtual reality technologies, handwriting is no longer limited to paper or screens and has expanded into three-dimensional space. Unlike two-dimensional approaches, in-air handwriting-with its high degree of freedom and lack of tactile feedback-results in characters that are not only less legible but also lack the writer's personal style. Therefore, optimizing in-air handwriting characters is essential for advanced human-computer interaction in the future.

*Corresponding Author

By integrating various somatosensory devices, researchers have developed numerous in-air handwriting character recognition systems, focusing on algorithmic recognition of handwritten characters. Recent popular approaches (Gan, Wang, and Lu 2019, 2020; Gan et al. 2023; Wang and Du 2021) either use two-dimensional convolutional neural networks (2D-CNNs) on handwritten stroke images or employ recurrent neural networks (RNNs) or one-dimensional convolutional neural networks (1D-CNNs) on time-track data. Although these methods perform well in recognizing the content of in-air handwriting characters, they fail to restore the legibility of in-air handwriting characters or convey the user's handwriting style.

Figure 1: In-air handwritten characters have continuous strokes and irregular jitter. Our method restores character content and emulates the writer's handwriting style.

The task of font generation enables the model to learn font styles from a limited set of samples, achieving a high degree of imitation and reproduction. Some approaches generate raster font images for 9,169 characters in an end-to-end manner (Zeng and Pan 2022; Zeng et al. 2021; Liu et al. 2024b). Additionally, certain studies use unsupervised learning to generate raster font images for any combination of style and content (Xie et al. 2021; Wang et al. 2023; Pan et al. 2023; Liu et al. 2022; Zhu et al. 2020; Liu et al. 2026). However, these approaches treat fonts as static images, which differs from the dynamic process of human handwriting. Humans draw characters sequentially, stroke by stroke, rather than "instantly generating a complete image." The raster representation of fonts not only overlooks

personal style embedded in the writing process but also lacks editability.

With advancements in sequence models like RNN, LSTM, and Transformer, some methods have started modeling handwritten characters, representing strokes as continuous sequences of writing tracks (Zhang et al. 2017; Tang et al. 2019; Dai et al. 2023; Liu et al. 2024a, 2025). However, due to the instability of hand movement and lack of stroke pauses in in-air handwriting, characters tend to connect with irregular jitter, posing challenges in modeling long sequences for in-air handwriting strokes using autoregressive approaches. Diffusion models, which gradually denoise target data iteratively in a non-autoregressive (NAR) manner, show unique advantages in natural language processing (Li et al. 2022; Gong et al. 2023). Unlike the context-dependent approach in natural language, however, handwritten characters require explicit guidance on content and style, and the discrete nature of handwriting tracks presents limitations for directly applying continuous space models.

The goal is to optimize in-air handwriting characters with continuous lines and irregular jitter, restoring character content and mimicking the original style. We separate character content and style from the handwritten track, combining any style with content and reconstructing the in-air handwriting track in a NAR manner. Due to the continuity of in-air handwriting strokes and the lack of gaps between characters, effective segmentation is challenging. We employ an overlapping sliding window to obtain variable-length handwritten strokes. We represent the in-air handwriting track as a discrete sequence of SVG drawing parameters, converting these parameters into binary sequences mapped to real number space, and reconstruct it using continuous diffusion models. The reconstruction process is divided into two phases: in the content preservation phase, we use a partial noise injection strategy with reference sequence modeling to fully leverage the content information of the original stroke, maintaining consistency of handwritten stroke content. Since the early-stage content features contain substantial noise, making style extraction difficult, but the reference sequence includes the necessary strokes forming the target character, we apply an adaptive fusion parameter during the style aggregation phase to adaptively merge the bimodal style features extracted from both content and reference sequences. The contributions of this paper are summarized as follows:

- A diffusion model-based method was proposed to optimize in-air handwriting characters. With only a few character samples, the model could optimize handwritten traces by imitating the user's writing process and style.
- We represented discrete handwritten trace sequences as binary sequences, using continuous-state diffusion models for modeling. Discrete handwritten traces were generated through threshold quantization, avoiding the non-smooth nature of directly generating discrete sequences.
- A partial noise injection strategy with reference sequence modeling was proposed, utilizing the content information of the original character as a conditional guide to maintain consistency of handwritten character content.
- The limitation of noise interference during the backward process that restricts effective style extraction from content features was overcame. Through adaptive fusion, we merged the visual style in the image modality with the dynamic writing process in the sequence modality of handwritten traces at different stages of sampling.
- To address the scarcity of paired "perfect/imperfect handwriting" data, we designed a zero-annotation simulation method that generates low-quality handwriting samples for model training, thereby pioneering exploration in this domain under data-constrained conditions.

Related Work

In-Air Handwriting Systems

In-air handwriting represents an innovative form of character input that transcends traditional paper and screen limitations by expanding into three-dimensional space. In recent years, numerous in-air handwriting systems have been developed. For example, Amma *et al*. (2012) introduced a 3D in-air handwriting system using glove sensors mounted on the back of the hand, enabling users to write in mid-air. Xu *et al*. (2015) designed an in-air handwriting Chinese character recognition system based on Leap Motion. Additionally, Gan *et al*. (2019) developed an in-air handwriting system employing an LSTM-based sequence-to-sequence model. These systems primarily focus on character or word recognition. Moreover, with handwritten characters, lines tend to connect, and the lack of spacing between characters makes irregularities in the characters sequence more pronounced. To address these issues, we treat in-air handwriting traces as a series of variable-length segments and apply an overlapping sliding window approach, eliminating the need for pre-alignment or character segmentation.

Handwriting Font Generation

Unlike printed fonts, however, handwritten fonts are characterized by curved lines and varying character sizes, adding complexity to the generation of handwritten fonts. Some methods treat handwriting as images, capturing the visual features of handwritten text to emulate individual handwriting styles to some degree (Gan and Wang 2021; Pippi, Cascianelli, and Cucchiara 2023). However, these methods do not consider the dynamic information of the writing process. To address this limitation, several studies employ sequence models to process handwritten characters (Zhao et al. 2020; Tang and Lian 2021; Aksan, Pece, and Hilliges 2018; Dai et al. 2023; Wang, Wang, and Liu 2025), incorporating both the visual characteristics of characters and the dynamic information of the writing process. While these methods can generate stylistically consistent font images, they lack the capability to optimize handwritten traces. Additionally, these methods predict each subsequent mark in sequence, which can be inefficient when processing longer sequences of handwritten characters. In our approach, we treat in-air handwriting characters as continuous writing track sequences, learning writing styles from these sequences and employing an iterative NAR parallel optimization for in-air handwriting characters.

Diffusion Model

Diffusion models (Sohl-Dickstein et al. 2015; Ho, Jain, and Abbeel 2020) generate high-quality outputs by iteratively removing noise. Recent studies (Hoogeboom et al. 2021; Austin et al. 2021) have adapted diffusion models for text in discrete space, based on unconditional language modeling. Compared to autoregressive models, which predict each token sequentially via causal attention masking, diffusion models iteratively refine samples in a highly parallel manner, requiring far fewer sampling steps than the data dimensionality (Chen, Zhang, and Hinton 2023). Diffusion-LM (Li et al. 2022) for constrained text generation and DiffuSeq (Gong et al. 2023) for sequence-to-sequence text generation were among the first to apply diffusion models to sequence modeling, and this was followed by applications in machine translation (Yuan et al. 2022; Gao et al. 2022) and summarization (Zhou et al. 2023; Zhang, Liu, and Zhang 2023). In contrast to natural language generation, generating discrete drawing parameters remains a challenge due to the inherent discreteness of SVG drawing parameters. We convert the coordinates of handwritten tracks into binary sequences and model them using continuous diffusion models. This approach simplifies the generation of discrete data without the need to introduce discrete state spaces or modify the diffusion process.

Method Description

Method Overview

For in-air handwriting characters, we represent the dynamic handwritten track through SVG vector drawing parameters (cf. Sec. In-Air Handwriting Data Structure). An overlapping sliding window approach is used to capture variable-length handwritten tracks (cf. Sec. Sliding Window). We convert the drawing commands and coordinates to binary sequences, which are then mapped to a real-number space through a linear transformation (cf. Sec. Binarized Encoding). Leveraging the advantages of continuous diffusion models, we apply simple thresholding to the model's predictions to reconstruct the SVG drawing parameters (cf. Sec. Forward Process and Reverse Process).We divide the reconstruction process of in-air handwriting characters into two stages: Content Preservation Stage (cf. Sec. Content Preservation) and Style Aggregation Stage (cf. Sec. Style Aggregation).

In-Air Handwriting Data Structure

As shown in Figure 2(a) and (b), to capture detailed writing features and maintain the editability of in-air handwriting strokes, we represent them as vector images drawn with straight lines using SVG drawing parameters. Handwritten strokes are composed of the drawing command parameters L and M along with coordinates. Figure 2(c) provides a structured example of drawing parameters for handwritten characters.

Simulates In-Air Handwritten

Since there is not yet any paired in-air handwriting data of "stroke discontinuity" and "stroke continuity", we add

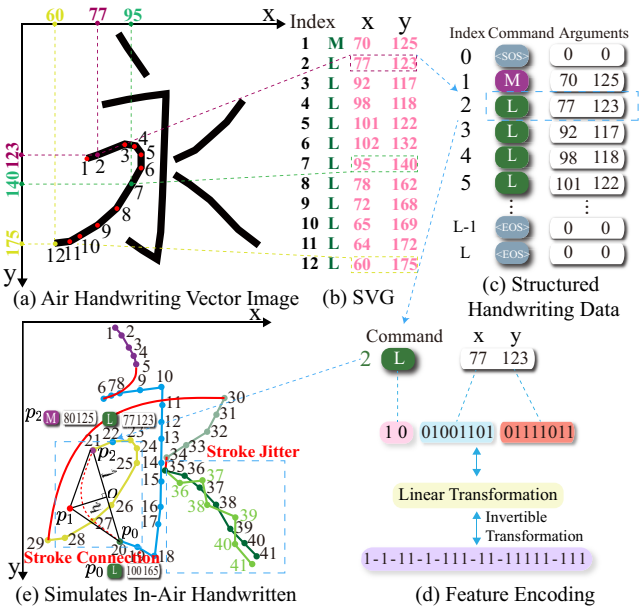

Figure 2: (a) Example of handwritten character. (b) SVG drawing parameters for handwritten character. (c) Visualization of in-air handwriting character data structure. The drawing command begins with SOS, command M denotes the starting point, and L denotes the drawn track sequence, ending with EOS. Zeroes indicate padding. (d) Binary feature encoding, with drawing coordinates converted to binary and mapped to a continuous real-number space using linear transformation.(e) Using quadratic Bézier curves (red line) to connect the end and start points of lines, simulating the continuity of in-air handwritten strokes. Add coordinate offsets to simulate jitter in in-air handwriting.

stroke connections and jitter to the CASIA-OLHWDB dataset to simulate in-air writing. **Stroke Connection:** As shown in Figure 2(e), the end point of one line is denoted as the start point p_0 of a quadratic Bézier curve, and the start point of the next line as the end point p_2. The control point p_1 is positioned at the midpoint of the line connecting p_0 and p_2, at a distance h from this line. After obtaining the quadratic Bézier curve, it is fitted with a linear Bézier curve. **Stroke Jitter:** To simulate the jitter caused by lack of hand stability, an offset $\Delta_{x,y} \in [-5, 5]$ is added to each coordinate.

Data Augmentation

Data augmentation uses two methods: **Scaling:** The lines of the handwritten track are represented using a quadratic Bézier curve formula $B(t) = (1-t)p_0 + tp_1$, where p_0 and p_1 are the Bézier curve control points. To vary the endpoint of the lines, we keep the starting control point p_0 fixed and calculate $B(t)$ instead of p_1 by randomly selecting tt from $t \in \{0.8, 0.9, 1.0, 1.1, 1.2\}$. This method produces a slightly altered writing track while retaining the main characteristics of the track. **Translation:** We add a random offset Δ_x to all x-coordinates, with $\Delta_x \in [-5, 5]$, and similarly add

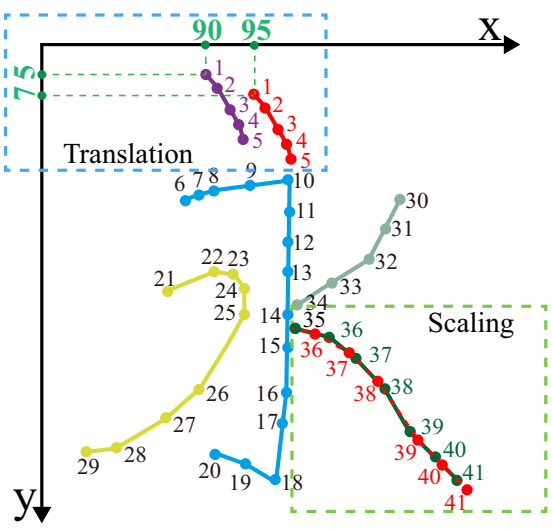

Figure 3: In-air handwritten character data augmentation.

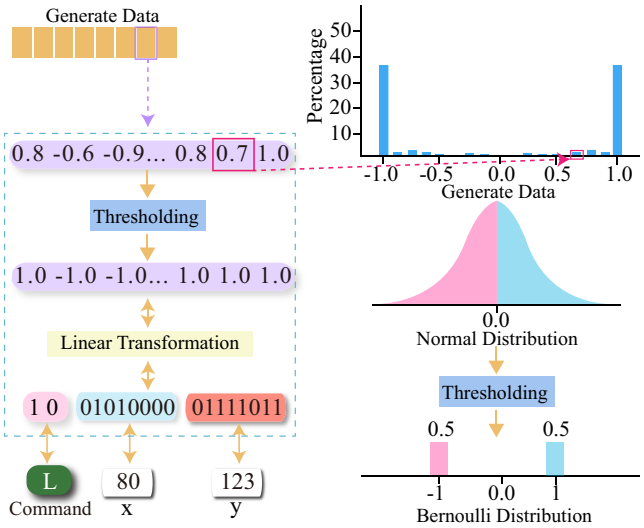

Figure 4: Invertible binary encoding of discrete coordinate parameters. Bernoulli-partition-based discretization of the noise space in a continuous diffusion model.

an offset Δ_y to the y-coordinates. This method horizontally and vertically shifts the handwritten characters, thereby augmenting the data. In-air handwritten character data augmentation is illustrated in Figure 3.

Binarized Encoding

Since the drawing parameters are discrete, a straightforward approach would be to re-encode the drawing parameters using a discrete data space and state space(Liu et al. 2024c; Ren et al. 2023; Kong, Liu, and Yao 2025). However, this would require defining an independent state or category for each discrete value, increasing complexity during model generation and inference. We convert the coordinates of the drawing parameters to binary, then apply a linear transformation to map the binary $\{0, 1\}$ to the continuous real number space $\{-1.0, 1.0\} \in \mathbb{R}$. This avoids a complex discrete state space, allowing direct use of continuous diffusion models. Specifically, for the i-th drawing parameter $v_i = (h_i, p_i)$, the drawing command h_i has four states, represented with $\log_2 4$ bits. The coordinate range is set to $[0, 255]$, with each coordinate represented using $\log_2 256$ bits. A linear transformation then maps it to $\{-1.0, 1.0\}$, providing a reversible transformation without any training parameters. The feature encoding of the i-th drawing parameter is shown in Figure 2(d).

As shown in Figure 4, the data generated by the continuous diffusion model can directly partition the noise space into two regions, each corresponding to a possible output of a Bernoulli distribution, ensuring that the generated results strictly follow the target distribution. This method allows the model to remain differentiable during training and ultimately achieve a discrete effect through simple thresholding or region partitioning.

Sliding Window

The size of the sliding window is defined as $p/2 + p$. Each window overlaps with the previous one by $p/2$ at each shift. This overlap retains part of the stroke information at the window boundaries to provide contextual information. We include the coordinates from the drawing parameters that fall within the current window into the selected sequence, while truncating those outside.

Forward Process

To maintain the consistency of handwritten stroke content during reconstruction, we propose a partial noise injection strategy that references the source sequence for modeling. First, we concatenate the content handwritten strokes $x \in \mathbb{R}^{(L/2+L)\times 18}$ and target handwritten strokes $y \in \mathbb{R}^{L\times 18}$ as $z = x_0 \oplus y_0$. The input is mapped via a linear layer to $z_0 \in \mathbb{R}^{(N+L)\times d}$, where $N = L/2 + L$. Sine-cosine position encoding is then added to z_0. This transformation allows us to convert discrete drawing parameters into a standard Markov forward process.

During the forward process, z_0 is perturbed. With each step $q(z_t \mid z_{t-1})$, noise is injected only into y_{t-1}, preserving the overall integrity of z_t. This modification enables diffusion models to use x as a content reference in modeling. After T steps of forward random perturbations, z_0 is ultimately converted to partial Gaussian noise $y_T \sim \mathcal{N}(0, I)$.

$$q(z_t \mid z_{t-1}) = \mathcal{N}\left(z_t; \sqrt{1-\beta_t} z_{t-1}, \beta_t I\right), \quad (1)$$

where $t = 1, 2, \ldots, T$ and $\{\beta_t \in (0, 1)\}_{t=1}^{T}$ represent the variance schedule. We define $\alpha_t = 1 - \beta_t$ and $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$. At any time step z_t,

$$z_t = \sqrt{\bar{\alpha}_t} z_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \quad (2)$$

where ϵ stands for Gaussian noises. Thus sampling z_t at arbitrary timestep has a closed form:

$$q(z_t \mid z_0) = \mathcal{N}\left(z_t; \sqrt{\bar{\alpha}_t} z_0, (1-\bar{\alpha}_t) I\right), \quad (3)$$

where $\bar{\alpha}_t = 1 - \sqrt{t/T + s}$ represents the sqrt noise schedule, with s being a very small constant.

Reverse Process

The reverse process generates data from isotropic Gaussian noise z_T , and gradually recovers z_0 via the reverse distribution $p_\theta\left(z_{t-1} \mid z_t\right)$.

$$p_\theta\left(z_{0:T}\right) := p\left(z_T\right) \prod_{t=1}^{T} p_\theta\left(z_{t-1} \mid z_t, z_0\right). \tag{4}$$

We can sample z_{t-1} using this formula, implementing the reverse generation process.

$$\hat{x}_0(z_t, \epsilon_t) = \frac{z_t - \sqrt{1-\overline{\alpha}_t}\,\epsilon_t}{\sqrt{\overline{\alpha}_t}}, \tag{5}$$

$$z_{t-1} = \sqrt{\overline{\alpha}_{t-1}}\,\hat{x}_0 + \sqrt{1-\overline{\alpha}_{t-1}-\sigma^2}\,\epsilon_t + \sigma^2\epsilon. \tag{6}$$

We compute the variational lower bound following the original diffusion process.

$$\mathcal{L}_{\mathrm{VLB}} = \min_\theta \left[\sum_{t=1}^{T} \left\|\mathbf{y}_0 - \tilde{f}_\theta\left(\mathbf{z}_t, t\right)\right\|^2 + \mathcal{R}\left(\left\|\mathbf{z}_0\right\|^2\right)\right], \tag{7}$$

where the regularization term $\mathcal{R}\left(\left\|\mathbf{z}_0\right\|^2\right)$ maintains the stability of embedded features. Here, $\tilde{f}_\theta\left(\mathbf{z}_t, t\right)$ denotes the reconstructed features of the model, represented as $\hat{y}_0 \in \mathbb{R}^{(L\times 18)}$. During model optimization, strict constraints enforcing exact binary outputs are unnecessary, allowing optimization in a continuous space, which avoids the non-smoothness associated with generating exact binary values. A simple thresholding of the model's reconstruction restores the original discrete plotting parameters, with each model output corresponding to a unique and meaningful plotting parameter.

Content Preservation

Unlike natural language processing (NLP), which models target sentences based on contextual information, the context of handwritten characters is only weakly associated with the noise y_t (i.e., previous and subsequent handwritten characters are independent). We propose reconstructing only the noise y_t while using unperturbed x_t as a conditional guide for handwritten characters content. The model iteratively approximates the target data distribution over T steps without relying on a separate content encoder or classifier.

As show in Figure 5(d), we split the feature z_t to obtain x_t and y_t. Treating y_t as query Q and key K_y, we compute self-attention $A_y = \mathrm{softmax}\left(\frac{QK_y^\top}{\sqrt{d}}\right) \in \mathbb{R}^{L\times L}$. Then, considering x_t as key K_x and value V, we calculate cross-attention $A_x = \mathrm{softmax}\left(\frac{QK_x^\top}{\sqrt{d}}\right) \in \mathbb{R}^{L\times N}$. The aggregated handwritten stroke content features $V_y = A_x V \in \mathbb{R}^{L\times d}$ are then assigned to each query token. Subsequently, using V_y as the value, we broadcast the global information to y_t, resulting in the final output $z_c = A_y V_y \in \mathbb{R}^{L\times d}$, equivalent to:

$$z_c = \sigma\left(QK_y^T\right)\sigma\left(QK_x^T\right)V, \tag{8}$$

where $\sigma(\cdot)$ represents the softmax function.Finally,by combiningz_t and z_c through residual connections, we obtain the updated feature representation $z_c' = [x_t; y_t + z_c]$. This approach avoids self-attention calculations for z_t, reducing computational complexity while facilitating information exchange between x_t and y_t.

During iterations, the model optimizes for the current diffusion step. We use Adaptive Layer Normalization (AdaLN) (Peebles and Xie 2023) to incorporate the diffusion timestep t into the model, where an MLP learns modulation parameters γ_c and β_c to adjust normalized features with timestep information.

$$\mathrm{AdaLN}(z_c', t) = \gamma_c(t) \odot \left(\frac{z_c' - \mu_c}{\sigma_c}\right) + \beta_c(t), \tag{9}$$

where μ_c and σ_c represent the mean and standard deviation.

Style Encoder

Sequential Style Encoder. In each iteration, for content handwritten strokes, K handwritten character sequences are randomly selected as style references. These are then passed through a style encoder, which consists of a six-layer multi-head self-attention transformer, to extract style features $f_s \in \mathbb{R}^{(K\times L)\times d}$.

Image Style Encoder. For K in-air handwriting character sequences, binary rasterized images are created. The features $f_c \in \mathbb{R}^{K\times 1024}$ are then extracted using a six-layer Conv-BN-ReLU network. We employ contrastive learning to pre-train the image style encoder. After pre-training, the gradients of the image style encoder are frozen.

Style Aggregation

In the early sampling stages, the content feature z_y consists of substantial noise and lacks effective content information. This prevents the cross-attention mechanism, when using z_y as the query to aggregate style features, from focusing on relevant style features. On the other hand, z_x contains the content features of the input character; despite the continuous lines and irregular jitter present in z_x, it still encompasses the essential lines forming the target character. Therefore, we propose an adaptive style aggregation module.

Specifically, we divide the features from the previous layer into two parts: $z_c' = [z_x; z_y]$, where z_x represents content features with distortions and continuous lines, and z_y represents content features containing noise. Then, we apply a linear projection to z_x of shape $\left(\frac{1}{2}L + L\right) \times d$, mapping it to a tensor of shape $L \times d$. We then use z_y as query Q_y and z_x as query Q_x, performing dot product operations with the Key K_s of the style features, resulting in $U_y = Q_y K_s^\top$ and $U_x = Q_x K_s^\top$.

In each fusion module, after embedding the feature at time t, it is fed into a linear layer to predict an adaptive fusion parameter α, which is used to integrate U_y and U_x to generate

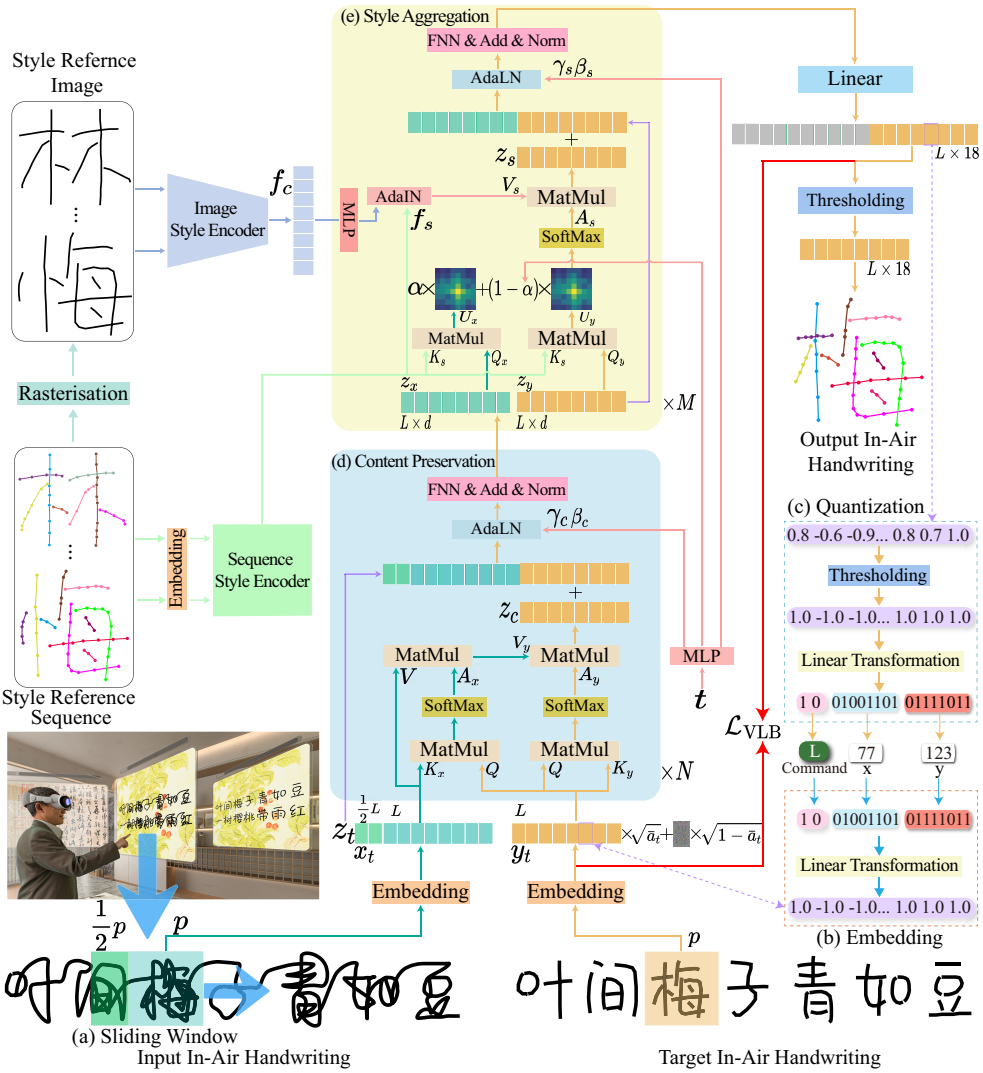

Figure 5: Overview of the proposed method. (a) Overlapping windows keep context for continuous strokes. (b) Strokes are encoded as SVG parameters for diffusion and (c) quantized back to SVG after sampling. In the forward process, content x and target y are modeled jointly, but noise is added only to y; the reverse iteratively predicts y_0. Reconstruction separates (d) content preservation and (e) style aggregation via adaptive fusion of content feature z_x and stage-wise target features z_y.

a fused attention matrix at different sampling stages:

$$A_s = \sigma\left(\frac{\alpha U_x + (1-\alpha)\, U_y}{\sqrt{d}}\right), \tag{10}$$

where $\sigma(\cdot)$ represents the Softmax function.

For the style feature f_c extracted by the image style encoder, an MLP layer learns a $\gamma \in \mathbb{R}^{1\times d}$ and a $\beta \in \mathbb{R}^{1\times d}$. Applying AdaIN to the style features f_s extracted by the sequence encoder, we obtain the Value V_s.

$$\mathrm{AdaIN}(f_s, f_c) = \gamma\left(\frac{f_s - \mu(f_s)}{\sigma(f_s)}\right) + \beta, \tag{11}$$

where $\mu(f_s)$ and $\sigma(f_s)$ are the normalization operations for feature f_s.

The fused style features of the sequence and image modalities V_s are aggregated with the queried style features using the attention matrix $z_s = A_s V_s$. Finally, a residual connection is applied to obtain the fused style feature representation $z'_s = [z_x; z_y + z_s]$.

Experiments

Handwriting Dataset. The CASIA-OLHWDB (1.0-1.2) dataset (Liu et al. 2011) includes approximately 3.7 million online handwritten Chinese characters from 1,020 writers as the training set. For testing, 60 writers each provide 3,755 characters. Additionally, we tested on the IAHCT-UCAS 2018 (Gan, Wang, and Lu 2020) real in-air handwritten dataset .

Evaluation Metrics. We use Dynamic Time Warping (DTW) (Chen et al. 2022) elastic matching technology to calculate the distance between generated and real handwrit-

Figure 6: Qualitative comparison with state-of-the-art online Chinese stroke generation methods.

Method	Simulate In-Air Handwritten Characters (CASIA-OLHWDB)				In-Air Handwritten Characters (IAHCT-UCAS 2018)			
	Style ↑	Content ↑	DTW ↓	User ↑	Style ↑	Content↑	DTW ↓	User ↑
Drawing (Zhang et al. 2017)	25.57	52.42	2.1331	3.6	20.46	43.42	2.4331	3.5
FontRNN (Tang et al. 2019)	33.04	58.71	2.0881	6.7	25.21	48.28	2.1125	6.2
Diff-Writer (Ren et al. 2023)	39.31	62.03	1.9122	8.1	29.32	54.33	2.0932	6.9
DeepImitator (Zhao et al. 2020)	45.31	68.03	1.7322	9.6	39.32	62.21	2.0675	7.6
WriteLikeYou (Tang and Lian 2021)	70.35	80.48	1.4232	40.6	64.32	72.23	1.8941	32.5
SDT (Dai et al. 2023)	80.46	86.21	1.2021	56.7	72.34	81.67	1.5365	46.4
ElegantlyWritten(Liu et al. 2024c)	83.50	90.54	1.1114	64.2	80.50	87.94	1.3387	51.1
Ours	88.41	93.87	0.9214	69.5	87.85	92.16	1.0989	68.3

Table 1: Comparisons with state-of-the-art methods on CASIA-OLHWDB and IAHCT-UCAS 2018.

ing trajectory sequences , allowing for nonlinear alignment. The content-and-style scores and the user-preference study were conducted exactly as in ElegantlyWritten (Liu et al. 2024c).

Comparison with Other Methods

Qualitative Comparison. The visual results of simulated and real in-air handwritten characters are shown in Figure 6. In contrast, our method progressively iterates on the original stroke as a conditioning guide, reconstructing the handwritten character step-by-step, consistently extracting stable style features across image and sequence modalities, which enhances stability and style coherence.

Quantitative Comparison. Table 1 presents the quantitative analysis results of simulated and real in-air handwritten data. In contrast, our method exhibits a smaller decline in performance on in-air handwriting characters, maintaining a substantial advantage across various metrics. However, human perception is sensitive to subtle differences, and testers can still detect minor discrepancies between synthetic and real strokes.

Ablation Study

Analysis of Feature Encoding Effectiveness. We compared the L2 loss, Logistic loss and Cross-Entropy loss with two feature encoding methods, with experimental results shown in Table 2. Our method bypasses the complex discrete state space of traditional models and leverages the advantages of diffusion models by using the L2 loss function to achieve smooth, continuous gradients.

Effectiveness of Partial Noise Injection Strategies. As shown in Table 3, when the partial noise injection strategy is removed, the lack of conditional guidance on handwritten character content leads to a significant drop in generation quality, with the content score being most noticeably affected.

Feature Encoding	Loss Function	Style ↑	Content ↑	DTW ↓
One-Hot	Cross-Entropy	78.28	88.64	1.4357
One-Hot	L2	56.43	68.55	1.9786
Binarized	Logistic	82.28	89.73	1.2836
Binarized	L2	87.85	92.16	1.0989

Table 2: Comparison of different feature encoding methods and loss functions.

Partial Noise Injection Strategy	Style ↑	Content ↑	DTW ↓
w/o	73.55	76.61	1.5142
w/	87.85	92.16	1.0989

Table 3: Ablation study of the partial noise injection strategy.

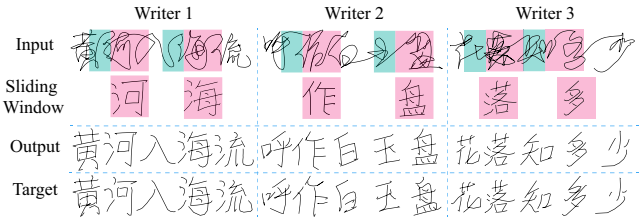

Figure 7: Optimized results of capturing incomplete in-air handwritten characters using overlapping sliding windows.

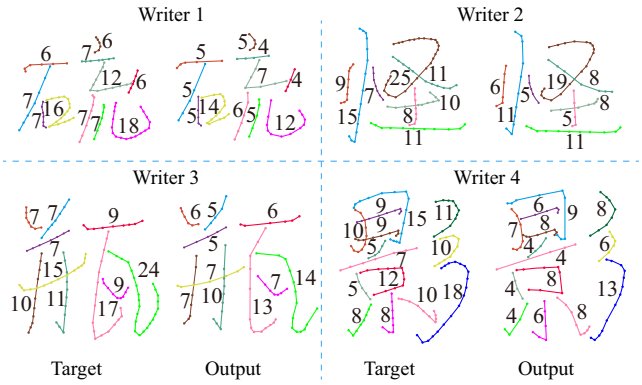

Figure 8: Simplicity analysis of drawing commands. Black numbers indicate the order in which each stroke is written.

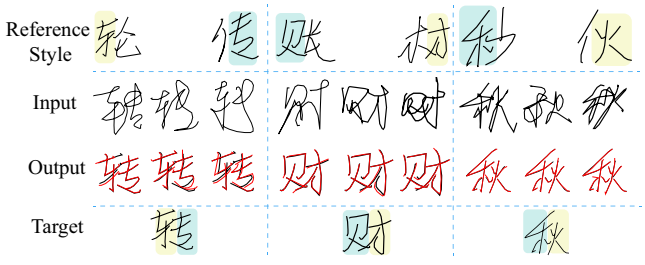

Figure 9: Optimization results for characters with different degrees of distortion.

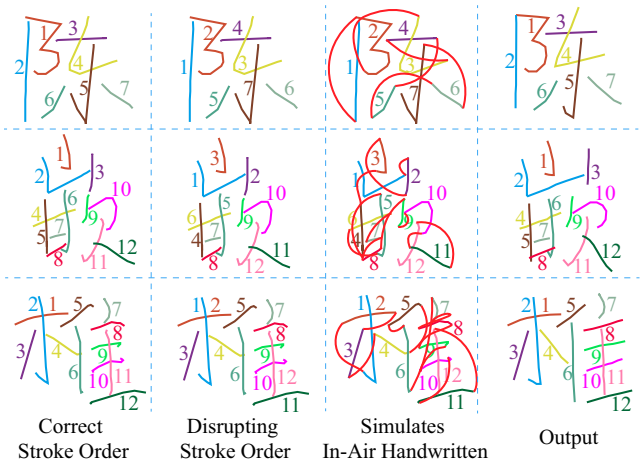

Figure 10: Impact of incorrect stroke order on stroke optimization.

Analysis

Optimization of Incomplete Characters. As shown in Figure 7, we use overlapping sliding windows, allowing a character that is incomplete in one window to be fully captured in the next. Overlapping windows provide the model multiple observations of the same data position, enhancing its ability to capture incomplete characters.

Simplicity Analysis of Drawing Commands. The number of vector plotting parameters for each line indicates the simplicity of its representation. As show in Figure 8, our method optimizes handwritten strokes to achieve simplicity in stroke representation.

Handwriting Correction. We introduced three levels of distortion to handwritten characters to test our method's ability to restore character readability. As shown in Figure 9, character readability improves as our method eliminates irregular distortions in the original handwritten characters while preserving the user's unique handwriting style.

Effect of Incorrect Stroke Order. Despite the strict stroke order required for Chinese characters, variations in individual writing habits can result in deviations from the standard sequence. We input characters with a randomized stroke order into the model. As shown in Figure 10, our method effectively adjusts the strokes, restoring high readability in the characters. This finding demonstrates that our approach can accommodate variations in writing errors.

Conclusion

This paper proposes a method to improve the readability of in-air handwritten characters while reproducing the user's writing style. By modeling the discrete trajectory parameters of in-air handwriting with continuous diffusion models and reconstructing them through a two-stage process. The promising experimental results verify the effectiveness of our proposed method. **Limitation:** The model was trained on simulated in-air handwriting datasets. Real-time is not considered. **Negative Impact:** This technology could potentially be misused to mimic a user's signatures.

Acknowledgements

This study was supported in part by Liaoning Provincial Science and Technology Plan Joint Program (Technology R&D Program Project) under Grants 2024JH2/102600108, the Science and Technology Innovation Fundation of Dalian under Grant 2023JJ12GX026, and in part by the Foundation of Key Laboratory of Education Informatization for Nationalities (Yunnan Normal University, Ministry of Education.) under Grant EIN2024B002.

References

Aksan, E.; Pece, F.; and Hilliges, O. 2018. Deepwriting: Making digital ink editable via deep generative modeling. In *Proceedings of the 2018 CHI conference on human factors in computing systems*, 1–14. ACM.

Amma, C.; Georgi, M.; and Schultz, T. 2012. Airwriting: Hands-free mobile text input by spotting and continuous recognition of 3D-space handwriting with inertial sensors. In *International Symposium on Wearable Computers*, 52–59.

Austin, J.; Johnson, D. D.; Ho, J.; Tarlow, D.; and Van Den Berg, R. 2021. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34: 17981–17993.

Chen, T.; Zhang, R.; and Hinton, G. 2023. Analog bits: Generating discrete data using diffusion models with self-conditioning. In *The Eleventh International Conference on Learning Representations*, 1–13.

Chen, Z.; Yang, D.; Liang, J.; Liu, X.; Wang, Y.; Peng, Z.; and Huang, S. 2022. Complex handwriting trajectory recovery: Evaluation metrics and algorithm. In *Proceedings of the Asian Conference on Computer Vision*, 1060–1076. Springer.

Dai, G.; Zhang, Y.; Wang, Q.; Du, Q.; Yu, Z.; Liu, Z.; and Huang, S. 2023. Disentangling Writer and Character Styles for Handwriting Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5977–5986. IEEE.

Gan, J.; Chen, Y.; Hu, B.; Leng, J.; Wang, W.; and Gao, X. 2023. Characters as graphs: Interpretable handwritten Chinese character recognition via Pyramid Graph Transformer. *Pattern Recognition*, 137: 109317.

Gan, J.; and Wang, W. 2019. In-air handwritten English word recognition using attention recurrent translator. *Neural Computing and Applications*, 31: 3155–3172.

Gan, J.; and Wang, W. 2021. HiGAN: Handwriting Imitation Conditioned on Arbitrary-Length Texts and Disentangled Styles. In *Proceedings of the AAAI conference on artificial intelligence*, 7484–7492. AAAI Press.

Gan, J.; Wang, W.; and Lu, K. 2019. A new perspective: Recognizing online handwritten Chinese characters via 1-dimensional CNN. *Information Sciences*, 478: 375–390.

Gan, J.; Wang, W.; and Lu, K. 2020. In-air handwritten Chinese text recognition with temporal convolutional recurrent network. *Pattern Recognition*, 97: 107025.

Gao, Z.; Guo, J.; Tan, X.; Zhu, Y.; Zhang, F.; Bian, J.; and Xu, L. D. 2022. Empowering diffusion model on embedding space for text generation. *arXiv preprint arXiv:2212.09412*.

Gong, S.; Li, M.; Feng, J.; Wu, Z.; and Kong, L. 2023. DiffuSeq: Sequence to sequence text generation with diffusion models. In *International Conference on Learning Representations*, 1–13.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, 6840–6851.

Hoogeboom, E.; Nielsen, D.; Jaini, P.; Forré, P.; and Welling, M. 2021. Argmax flows and multinomial diffusion: Learning categorical distributions. In *Advances in Neural Information Processing Systems*, volume 34, 12454–12465.

Kong, Y.; Liu, J.; and Yao, C. 2025. Elegantly Written V2: Next-scale prediction for enhancing online Chinese handwriting. In *Chinese Conference on Pattern Recognition and Computer Vision*, 1–14. Springer.

Li, X.; Thickstun, J.; Gulrajani, I.; Liang, P. S.; and Hashimoto, T. B. 2022. Diffusion-lm improves controllable text generation. In *Advances in Neural Information Processing Systems*, 4328–4343.

Liu, C.-L.; Yin, F.; Wang, D.-H.; and Wang, Q.-F. 2011. CASIA online and offline Chinese handwriting databases. In *Proceedings of International Conference on Document Analysis and Recognition*, 37–41. IEEE.

Liu, X.; Meng, G.; Chang, J.; Hu, R.; Xiang, S.; and Pan, C. 2022. Decoupled representation learning for character glyph synthesis. *IEEE Transactions on Multimedia*, 24: 1787–1799.

Liu, Y.; binti Khalid, F.; binti Mustaffa, M. R.; and bin Azman, A. 2024a. Dual-modality learning and transformer-based approach for high-quality vector font generation. *Expert Systems with Applications*, 240: 122405.

Liu, Y.; binti Khalid, F.; Wang, C.; binti Mustaffa, M. R.; and bin Azman, A. 2024b. An end-to-end chinese font generation network with stroke semantics and deformable attention skip-connection. *Expert Systems with Applications*, 237: 121407.

Liu, Y.; binti Khalid, F.; Wang, L.; Zhang, Y.; and Wang, C. 2024c. Elegantly Written: Disentangling writer and character styles for enhancing online Chinese handwriting. In *European Conference on Computer Vision*, 409–425.

Liu, Y.; Ding, Y.; Khalid, F. B.; Wang, C.; and Wang, L. 2026. Few-shot font generation via denoising diffusion and component-level fine-grained style. *Expert Systems with Applications*, 296: 128987.

Liu, Y.; Khalid, F. B.; Wang, C.; Mustaffa, M. R. B.; and Azman, A. B. 2025. DiffVecFont: Fusing Dual-Mode Reconstruction Vector Fonts via Masked Diffusion Transformers. In *International Conference on Computational Visual Media*, 339–363. Springer.

Pan, W.; Zhu, A.; Zhou, X.; Iwana, B. K.; and Li, S. 2023. Few shot font generation via transferring similarity guided global style and quantization local style. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 19506–19516.

Peebles, W.; and Xie, S. 2023. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4195–4205.

Pippi, V.; Cascianelli, S.; and Cucchiara, R. 2023. Handwritten text generation from visual archetypes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 22458–22467. IEEE.

Ren, M.-S.; Zhang, Y.-M.; Wang, Q.-F.; Yin, F.; and Liu, C.-L. 2023. Diff-writer: a diffusion model-based stylized online handwritten Chinese character generator. In *International Conference on Neural Information Processing*, 86–100. Springer.

Sohl-Dickstein, J.; Weiss, E.; Maheswaranathan, N.; and Ganguli, S. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, 2256–2265. PMLR.

Tang, S.; and Lian, Z. 2021. Write Like You: Synthesizing your cursive online chinese handwriting via metric-based meta learning. *Computer Graphics Forum*, 40(2): 141–151.

Tang, S.; Xia, Z.; Lian, Z.; Tang, Y.; and Xiao, J. 2019. FontRNN: Generating large-scale Chinese fonts via recurrent neural network. *Computer Graphics Forum*, 38(7): 567–577.

Wang, C.; Zhou, M.; Ge, T.; Jiang, Y.; Bao, H.; and Xu, W. 2023. Cf-font: Content fusion for few-shot font generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1858–1867.

Wang, L.; Wang, C.; and Liu, Y. 2025. EHW-Font: A handwriting enhancement approach mimicking human writing processes. *Expert Systems with Applications*, 278: 127278.

Wang, Z.-R.; and Du, J. 2021. Joint architecture and knowledge distillation in CNN for Chinese text recognition. *Pattern Recognition*, 111: 107722.

Xie, Y.; Chen, X.; Sun, L.; and Lu, Y. 2021. DG-Font: Deformable generative networks for unsupervised font generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 735–751. IEEE.

Xu, N.; Wang, W.; and Qu, X. 2015. Recognition of in-air handwritten Chinese character based on leap motion controller. In *Image and Graphics: 8th International Conference*, 160–168. Springer.

Yuan, H.; Yuan, Z.; Tan, C.; Huang, F.; and Huang, S. 2022. Seqdiffuseq: Text diffusion with encoder-decoder transformers. *arXiv preprint arXiv:2212.10325*.

Zeng, J.; Chen, Q.; Liu, Y.; Wang, M.; and Yao, Y. 2021. Strokegan: Reducing mode collapse in chinese font generation via stroke encoding. In *AAAI*, 3270–3277.

Zeng, S.; and Pan, Z. 2022. An unsupervised font style transfer model based on generative adversarial networks. *Multimedia Tools and Applications*, 81(4): 5305–5324.

Zhang, H.; Liu, X.; and Zhang, J. 2023. DiffuSum: Generation enhanced extractive summarization with diffusion. *Association for Computational Linguistics*.

Zhang, X.-Y.; Yin, F.; Zhang, Y.-M.; Liu, C.-L.; and Bengio, Y. 2017. Drawing and recognizing chinese characters with recurrent neural network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4): 849–862.

Zhao, B.; Tao, J.; Yang, M.; Tian, Z.; Fan, C.; and Bai, Y. 2020. Deep imitator: Handwriting calligraphy imitation via deep attention networks. *Pattern Recognition*, 104: 107080.

Zhou, K.; Li, Y.; Zhao, W. X.; and Wen, J.-R. 2023. Diffusion-nat: Self-prompting discrete diffusion for non-autoregressive text generation. *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics*.

Zhu, A.; Lu, X.; Bai, X.; Uchida, S.; Iwana, B. K.; and Xiong, S. 2020. Few-shot text style transfer via deep feature similarity. *IEEE Transactions on Image Processing*, 29: 6932–6946.