Towards Human-Like Robot Handwriting via Contour-Aware Generation

Yutao Qin[1]*, Gang Dai[2]*, Yifan Zhang[3,4], Youwei Han[1], Qisheng He[1], Shuangping Huang[1,5]†
[1]South China University of Technology [2]Guangdong University of Technology
[3]MiroMind AI [4]National University of Singapore [5]Pazhou Laboratory
{eeytqin.mail, eehsp}@scut.edu.cn, daigang@gdut.edu.cn, yifan.zhang@miromind.ai

Abstract

Empowering machines to simulate human handwriting is a promising research direction. Most existing methods, however, primarily focus on reproducing the writing trajectory to capture the overall character structure, while neglecting the critical aspect of stroke contour modeling. Consequently, these methods struggle to generate visually realistic, human-like handwriting, limiting their applicability in scenarios such as calligraphy robots. To address this issue, we propose a new task, called Contour-aware Handwriting Trajectory Reconstruction (CHTR). This task presents two major challenges: 1) Existing handwriting datasets lack stroke contour annotations, making supervised learning difficult; 2) Previous methods are unable to recover stroke contour and preserve the overall character structure jointly. To address the dataset limitation, we present CHTR-110K, a large-scale character dataset with refined stroke contour annotations. To tackle the technical challenge, we propose Graph-based Handwriting Trajectory Reconstruction (G-HTR), a novel method using contour-aware graphs to jointly model stroke contour and character structure. We use a Graph Neural Network to capture structural relationships among nodes and introduce a multi-scale graph learning strategy to encode both fine-grained stroke details and global character structure. Extensive experiments verify the effectiveness of G-HTR, outperforming previous state-of-the-art methods on both our CHTR-110K and the widely-used CASIA-OLHWDB dataset. G-HTR further shows strong real-world results when deployed on robots, confirming its practical value. Our source code and dataset is available at https://github.com/RittoQin/CHTR

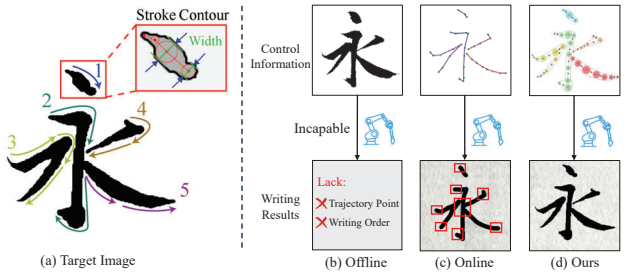


Figure 1. (a) Real human handwriting character, where the numbers indicate the writing order of each stroke, and the colored arrows show the writing trajectory of the strokes. Each stroke features a delicate contour with varying widths, represented by the diameter of the red circles. We employ various character generation methods to produce control information for the writing robot and present visual results in (b)-(d). Better zoom in 200%.

1. Introduction

Writing is one of humanity's most fundamental and essential skills, playing a crucial role in daily activities such as letter correspondence, note-taking, and signature authorization. Recently, enabling machines with the ability to write has emerged as a prominent research focus [41, 46], driven by applications in fields like handwriting robots [17, 44] and calligraphy education [18]. Our goal is to empower machines to produce realistic human-like handwriting, which remains a highly challenging task. As shown in Figure 1(a), achieving this requires careful attention to several key aspects: 1) Humans maintain accurate character structures, which consist of intricate topological connections formed by multiple strokes. 2) The strokes follow a precise order, each exhibiting delicate contours with varying widths.

Current character generation methods can be broadly categorized into two types: offline generation [33, 42] and online generation [7, 37]. Offline generation treats characters as static images, enabling the modeling of visual features such as character structures. However, as shown in Figure 1(b), these approaches fails to provide critical dynamic writing information (*e.g.*, stroke order and trajectory key points), making them unsuitable for controlling hand-

*Authors contributed equally. Work done by Gang Dai during his PhD period at South China University of Technology.
†Corresponding author.

writing robots. In contrast, online generation methods introduce dynamic writing information, enabling handwriting robots to perform writing tasks. However, these methods ignore the reconstruction of stroke contours, which hinders the visual realism. As shown in red boxes of Figure 1(c), the lack of detailed width information in the generated trajectories forces the robot to rely on a default fixed width. This limitation results in unreal strokes that fail to capture the natural width variations, leading to visual issues that lose the aesthetic appeal of calligraphy and instead resembles standard printed text. To address these limitations, we propose a new task called **C**ontour-aware **H**andwriting **T**rajectory **R**econstruction (**CHTR**). The primary goal of CHTR is to generate accurate trajectory sequences, encompassing trajectory points, stroke order and contours, thereby enabling human-like robot writing (cf. Figure 1(d)). Beyond this primary goal, CHTR offers substantial value to the broader computer vision community. Its precise contour-aware annotations serve as robust priors for downstream text analysis and synthesis tasks. Moreover, by emphasizing the dynamic writing process, CHTR facilitates immersive calligraphy education and drives embodied AI integration, where generated trajectory sequences directly guide robotic handwriting. Despite its significant potential, this task presents two major challenges: 1) **Dataset challenge:** Existing character datasets lack annotations that simultaneously provide trajectory points, stroke order and contours, making it difficult to train or evaluate CHTR models effectively. 2) **Technical challenge**: Accurately recovering both the overall character structure and the detailed stroke contours from static images remains a complex problem.

For the dataset limitations, existing widely-used character datasets like IAM [28], ICDAR [43], and CASIA [25] offer large-scale character images or contour-agnostic writing trajectory sequences. However, these datasets lack temporal information and trajectory key points in the images, while the trajectory sequences fail to capture stroke contours. This makes them inadequate for modeling dynamic and visually realistic human-like handwriting. To address this gap, we introduce CHTR-110K, the first large-scale dataset tailored for human-like robot handwriting. It comprises 110,540 samples spanning a broad lexicon and diverse writing styles. Each sample includes a contour-aware trajectory sequence annotated with trajectory key points, stroke order and contours. To ensure scalability and high-quality annotations, we develop a semi-automatic pipeline capable of accurately extracting contour-aware trajectories from character images.

As for the technical challenges, existing character generation methods typically rely on pixel-level image encoders to extract visual patterns from character images. These visual-based approaches face two main limitations: 1) They primarily focus on modeling local relationships between adjacent pixels, overlooking the intrinsic topological structure of characters. 2) They neglect low-level features from shallow network layers, which capture rich stroke details (*e.g.*, stroke curvature and joins) crucial for accurate handwriting reconstruction. Consequently, methods based solely on image encoding struggle to capture the complex structural relationships and fine-grained stroke details necessary for trajectory reconstruction.

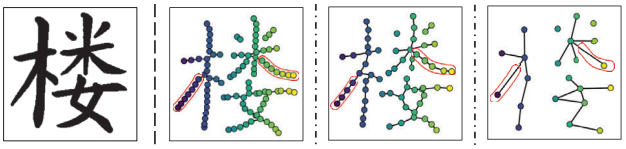


Figure 2. Multi-scale graphs are derived in different graph blocks. The red circle highlights the same stroke comparisons.

To tackle this challenge, we propose a novel Graph-based Handwriting Trajectory Reconstruction method (G-HTR). We begin by constructing a contour-aware character graph from the source character to explicitly model its topological structure. Specifically, trajectory key points with stroke widths are represented as nodes, while the topological connections between key points are represented as edges. This graph is then processed through a Graph Neural Network to facilitate structural representation learning. Following this, we introduce a multi-scale graph learning strategy to comprehensively capture both the overall structures and stroke details. Multi-scale graphs (cf. Figure 2) derived from the different graph blocks learn character features from fine-grained details to coarse structures. These features are then fed into a multi-scale aggregation decoder to autoregressively generate realistic trajectory sequences.

To sum, this work offers three key contributions:

- We propose a new task, Contour-aware Handwriting Trajectory Reconstruction, to empower machines to produce human-like handwriting. To support this task, we build CHTR-110K, the first large-scale dataset of 110,540 samples. This dataset includes high-quality trajectory key points, stroke order and contour annotations, providing a valuable foundation for future research.
- We propose a novel Graph-based Handwriting Trajectory Reconstruction method (G-HTR) that leverages multi-scale character graphs to explicitly model the topological relationships of characters and capture stroke details. This enables more accurate handwriting reconstruction.
- The proposed G-HTR not only achieves state-of-the-art performance on the CHTR-110K dataset and the publicly available handwriting trajectory reconstruction dataset CASIA-OLHWDB [25], but also demonstrates strong real-world performance when integrated into a calligraphy robot, marking a promising step toward practical, human-like handwriting.

2. Related Work

Handwriting Trajectory Recovery aims to restore static handwriting images to dynamic handwriting sequences. The early traditional handwriting trajectory recovery methods mainly depend on heuristic rules and image processing techniques [4, 22, 36], etc. These methods are suitable for simple characters or signatures, but they are difficult to generalize to multi-style, multi-stroke characters.

With the development of deep learning, some end-to-end trajectory recovery networks [3, 31] are proposed. For instance, TRACE [1] combines a Convolutional Recurrent Neural Network with a Dynamic Time Warping algorithm (DTW) to handle text-line trajectory recovery. To address complex glyphs and long trajectory recovery, PEN-Net [6] introduces a dual-path parsing encoder and a global tracking decoder architecture. Recently, some researchers proposed FINet [49], which integrates a spatial encoder and a temporal decoder to improve the accuracy of Chinese character trajectory recovery. Recently, TrajFormer [24] introduces Transformer [14, 39] to model the long-term sequence dependency of stroke trajectories.

However, none of the aforementioned methods explicitly model the character stroke contours, thus failing to directly recover realistic stroke details. In contrast, our method utilizes character graph to explicitly encode both stroke contours and structural relationships, thereby achieving authentic human-like handwriting. We discuss further related work on **Graph Neural Networks** in Appendix A.

Handwriting Generation is to generate handwritten images or trajectories with controllable styles and content. Popular methods [7–9, 32] adopt a style-content disentanglement pipeline. They extract style features from reference samples and combine them with textual content to generate the desired handwriting. Unlike these methods, which require both content and style inputs, our task only needs a single character image to reconstruct the handwriting trajectory. We provide more discussions in Appendix A.

Character datasets. Existing character datasets are typically divided into online and offline datasets. For instance, IAM-OnDB [26] is an online English handwritten dataset, containing approximately 86,000 word samples from 221 writers. For Japanese, Kondate [29] is an online handwriting dataset containing approximately 1,106 character categories from 100 writers. In the field of handwritten Chinese character, the CASIA-OLHWDB1.0-1.2 [25] is a large-scale online handwriting dataset, including 7,356 categories of Chinese characters from 1,020 writers. As for offline datasets, CASIA-HWDB1.0-1.2 [25] is an offline Chinese handwriting dataset, similar to the CASIA-OLHWDB1.0-1.2, containing 7,356 categories of Chinese characters from 1,020 writers. As an offline Arabic dataset, IFN/ENIT [34] contains approximately 26,000 characters from 946 writers. Additionally, there are several non-handwritten character datasets, including font and calligraphy collections. For example, SCUT-SPCC [48] includes 280 different styles of printed Chinese characters, covering 3,755 commonly used Chinese characters. In terms of calligraphy datasets, MCCD [47] comprises works from 142 calligraphers across different dynasties, containing 7,765 character categories.

Online and offline datasets have inherent limitations that make them unsuitable for contour-aware handwriting trajectory reconstruction task. To address this issue, we introduce the CHTR-110K dataset, containing 110,540 characters and simultaneously annotating trajectory sequences and stroke contour-related parameters, accelerating the development of efficient contour-aware models.

3. Problem Definition

To produce dynamic and visually realistic human-like handwriting, we propose a new task: Contour-aware Handwriting Trajectory Reconstruction (CHTR). The goal of CHTR is to generate accurate trajectory sequences comprising key points, stroke order and contours, thereby enabling human-like robot writing. Specifically, given a character image, the model aims to generate a trajectory sequence $P = \{p_i\}_{i=1}^{n}$ that not only follows a natural stroke order, but also maintains the overall character structure and preserves the fidelity of each stroke's contour, where n is the number of trajectory points. Despite its practical importance, CHTR remains largely unexplored due to the lack of contour-aware datasets and the inherent challenges of jointly reconstructing the overall character structure and detailed stroke contours from a single image.

4. CHTR-110K Dataset

To develop a character dataset that is more suitable for contour-aware handwriting trajectory reconstruction, we construct CHTR-110K, a large-scale Chinese character dataset comprising 110,540 paired samples, and each pair is composed of a contour-aware trajectory sequence with its corresponding source character image. In this section, we further delve into the specifics of data construction and provide a detailed analysis of our CHTR-110K dataset.

4.1. Data Construction

We construct the CHTR-110K dataset based on a combination of font character images collected from the Founder Font Library and handwritten character images from CASIA-HWDB [25]. To ensure both scalability and annotation quality, we design a semi-automatic annotation pipeline that extracts high-fidelity trajectory sequences with stroke contour annotations from these images.

As shown in Figure 3, given a character image I, we first utilize a stroke extractor $\mathcal{F}_{stroke}$, which employs a UNet-based detector named SDNet [23] to detect each stroke re-

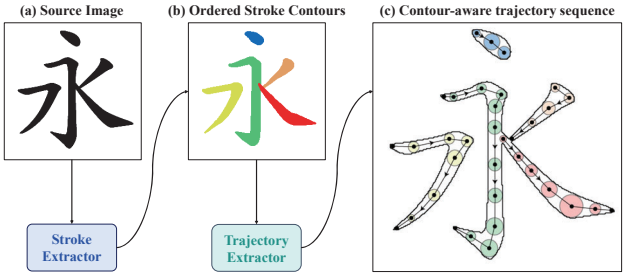


Figure 3. Pipeline for constructing the CHTR-110K dataset: (a) The source character image. (b) Ordered stroke contours: Each color represents a distinct stroke. (c) Contour-aware trajectory sequence: Circular points represent trajectory key points, arrows indicate the writing order within each stroke, and the diameter of each colored circle corresponds to the stroke width.

Figure 4. Visualization of source images and their corresponding contour-aware trajectory sequences. Better zoom in 200%.

gion and predict its stroke type and stroke order:

$$\{S_i, t_i, o_i\}_{i=1}^{M} = \mathcal{F}_{stroke}(I). \tag{1}$$

where S_i denotes the foreground region of the i-th stroke, t_i denotes the stroke type of the i-th stroke, o_i denotes stroke order of the i-th stroke. M denotes the number of the strokes. The Stroke type t_i is directly inferred from the class index (across 25 categories). The stroke order o_i is determined by matching the spatial position and predicted stroke type with the standard stroke composition of target character. Finally, we extract precise foreground regions for the ordered stroke contours.

For trajectory extractor $\mathcal{F}_{traj}$, we employ a CNN-LSTM based generator [30] to progressively derive contour-aware trajectory sequence P_i for i-th stroke from its corresponding region S_i:

$$P_i = \mathcal{F}_{traj}(S_i), \tag{2}$$

where P_i=$[p_1, p_2, \ldots, p_n]$ is the trajectory sequence of the i-th stroke. Each trajectory point $p = (x, y, w, s^1, s^2, s^3)$ includes 2D coordinates (x, y), stroke width w, and three mutually exclusive (*i.e.*, one-hot) pen states: s^1 (down), s^2 (up), and s^3 (end). After obtaining the trajectory sequence of all strokes, we sequentially concatenate them to obtain the trajectory sequence of the entire character P=$[P_1, P_2, \ldots, P_M]$. This pipeline constructs a contour-aware trajectory sequence that not only accurately reconstructs the character structure but also preserves detailed contours of each stroke.

CHTR-110K	Sample size			Styles	Classes
	Train	Test	Total		
Font	7,997	2,122	10,119	60	9,516
Handwriting	80,172	20,249	100,421	1,020	7,184
All	88,169	22,371	110,540	1,080	9,837

Table 1. Statistics of our CHTR-110K dataset. "Styles" refers to the number of calligraphy styles. "Classes" refers to the number of character classes.

Character Dataset	Contour	Order	Styles	Classes
MCCD [47]	✓	×	142	7765
SCUT-SPCC [48]	✓	×	280	3755
CVL [21]	✓	×	311	83
ICDAR2013 [43]	×	✓	60	3,755
Kondate [29]	×	✓	100	1,106
IAM-OnDB [26]	×	✓	221	81
CASIA-OLHWDB [25]	×	✓	1,020	7,356
CHTR-110K (Ours)	✓	✓	**1,080**	**9,837**

Table 2. Comparisons between CTHR-110K and existing datasets. "Contour" refers to stroke contours, and "Order" refers to stroke order annotations. "×" indicates the annotation absence.

To ensure the quality of the dataset, we employ post-processing steps that involve manually correcting errors, filtering out trajectory sequences that do not reproduce the stroke contour. We recruit 5 volunteers with undergraduate degrees for post-processing steps, which totals approximately *1,500 person-hours*. Furthermore, to quantitatively evaluate our annotation quality, we calculate the intersection-over-union (IOU) between $\hat{S}_i$ and S_i, where $\hat{S}_i$ is rendered by P_i. The resulting CHTR-110K dataset achieves a mean IOU of *0.972*, demonstrating that our annotations maintain high consistency with the ground truth. More details are put in Appendix C.

4.2. Data Analysis

As shown in Table 1, we construct trajectory sequences with authentic stroke contours from a total of 10,119 font character images and 100,421 handwritten character images. We randomly selected 88,169 samples to form our CHTR-110K training set. The CHTR-110K test set is sourced from the remaining trajectory sequences. Note that certain writing styles are exclusively allocated to the testing set to assess the generalization and robustness.

As shown in Table 2, compared to existing character datasets, CHTR-110K offers significant advantages in both diversity and annotation richness. It encompasses a broad range of character classes and writing styles, and critically provides authentic stroke contour annotations, which are absent in prior datasets. This unique property enables the training of contour-aware handwriting reconstruction mod-

els and supports applications such as handwriting generation and calligraphy robots.

We present several examples in Figure 4, which consistently maintain the character structure across various styles and preserve the detailed contour of each stroke. More example visualizations are provided in Appendix L.

5. Method

5.1. Overall Scheme

To achieve dynamic and visually realistic human-like handwriting, we introduce a new Graph-based Handwriting Trajectory Reconstruction method (G-HTR). As shown in Figure 5, our G-HTR consists of an image encoder, a multi-scale graph encoder, and a multi-scale aggregation decoder.

Starting from a source character image I, we initially construct a contour-aware character graph G, which is then input into the multi-scale graph encoder to extract multi-scale graph features F_g. Next, each scale of graph feature f_g along with the character image feature f_i is fed into a multi-scale aggregation decoder to generate the contour-aware trajectory sequence in an autoregressive manner.

The model is supervised using a stroke prediction loss $\mathcal{L}_{pre}$ and a pen state classification loss $\mathcal{L}_{cls}$ by comparing the reconstructed trajectory against the ground-truth annotations:

$$\mathcal{L} = \lambda \mathcal{L}_{pre} + \mathcal{L}_{cls}, \tag{3}$$

where λ serves as a trade-off factor, and we empirically set it to 0.5. Lastly, achieving human-like robotic writing requires considering multiple elements, including stroke order, pressure, anisotropic strokes, dynamics, and dwell time. While our contour-aware trajectory sequences inherently capture stroke order and pressure, we model the remaining factors by converting these trajectories into refined control sequences using the brush model and reinforcement learning pipeline proposed in [27]. Finally, these control sequences are executed by the robot to accurately reproduce the characters.

5.2. Contour-aware Character Graph Construction

Previous visual-based handwriting trajectory reconstruction methods [24, 49] primarily focus on modeling local relationships between adjacent pixels, overlooking the intrinsic topological structure of characters. In contrast, we represent the character image as a Contour-aware character graph to explicitly model the topological structure of the character.

We first apply a thinning algorithm [45] to the input image I to extract its skeleton I_s, and then cluster [10] the dense skeleton points to obtain a simplified set of key points $\{p_i\}_{i=1}^{N}$. The stroke width w_i at each p_i is estimated as its shortest distance to the character contour. Based on these attributes, we construct a contour-aware graph $G = (\mathbf{V}, \mathbf{E})$. Each node $v_i \in \mathbf{V}$ represents a key point, associated with a feature vector $f_g(v_i) = [x_i, y_i, w_i]$ that captures its 2D spatial coordinates (x_i, y_i) and stroke width w_i. The edge set $\mathbf{E}$ explicitly models the topological connectivity between adjacent key points. Finally, we add self-loops to all nodes and formulate G as an undirected graph.

5.3. Multi-scale Graph Encoder

We propose a multi-scale graph encoder to grasp comprehensive structural representations from the constructed contour-aware graph G. It is composed of several stacked graph blocks, each comprising a multi-head Graph Attention layer [40] and a Graph Convolutional layer [20]. The Graph Attention layer focuses on capturing global structural relationships, while the Graph Convolutional layer targets local topological features. Building on this, we develop a multi-scale graph learning strategy that extracts graph features $f_g \in \mathbb{R}^{N \times D}$ at various scales from multiple graph blocks, where N is the number of graph nodes and D is the dimension of node features. This design effectively captures the overall character structure and stroke details, guiding the precise reconstruction of trajectory sequences.

Graph Attention Layer. The graph attention layer is designed to capture global structural relationships by enabling interactions among all nodes in the graph. Given input graph features f_g, the query Q, key K, and value V are obtained by applying linear projections to f_g. The multi-head graph attention output Y is then calculated as:

$$Y = Softmax(\frac{QK}{\sqrt{D}})V. \tag{4}$$

Graph Convolution Layer. The graph convolution layer concentrates on capturing local topology structures of graphs by aggregating neighbourhoods of adjacent nodes, which can be defined as:

$$\tilde{f}\left(v_i\right) = f\left(v_i\right) + \frac{1}{\left|\mathcal{N}\left(v_i\right)\right|} \sum_{v_j \in \mathcal{N}(v_i)} w_{ij} f\left(v_j\right), \tag{5}$$

where $\mathcal{N}\left(v_i\right)$ denotes neighbourhoods of node v_i, and w_{ij} denotes the aggregation weight between node v_i and node v_j. After that, we employ graph coarsening [11, 12] to create a smaller-scale graph.

5.4. Multi-scale Aggregation Decoder

The goal of the multi-scale aggregation decoder is to autoregressively generate realistic trajectory sequences, denoted as $\hat{P} = \{\hat{p}_j\}_{j=1}^{L}$, with L being the total number of points, conditioned on the extracted character image feature f_i and the multi-scale graph features F_g.

At any decoding step t, we obtain the query vector Q_t by concatenating character image feature f_i and previous points $\{p_j\}_{j=1}^{t-1}$. Subsequently, Q_t attends to the multi-scale graph features F_g over subsequent aggregation modules with cross-attention mechanisms [5, 13] to adaptively

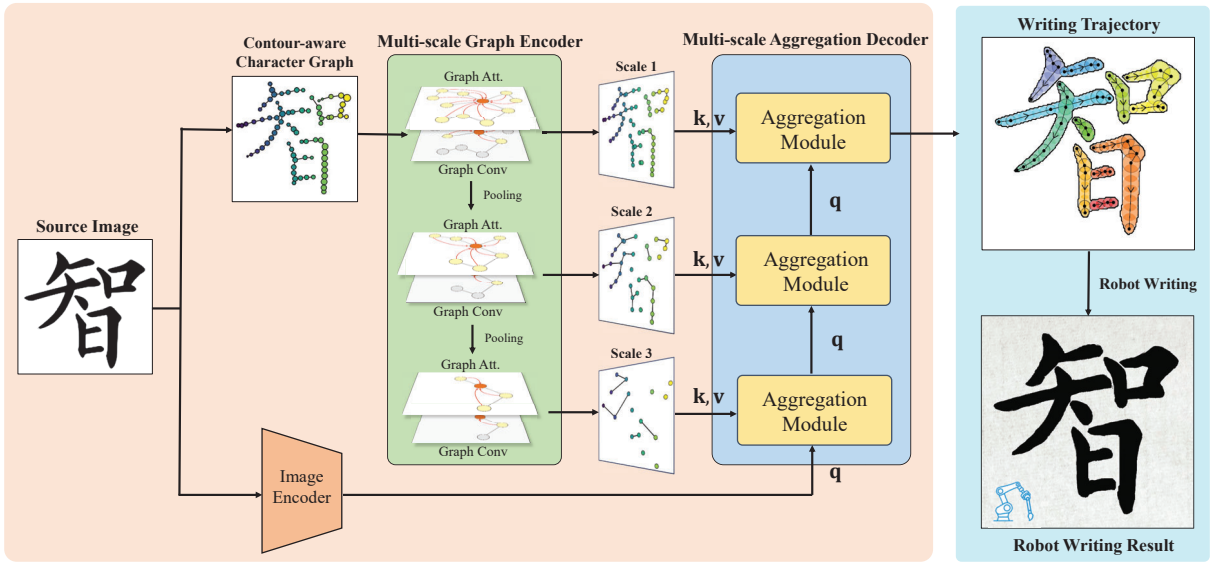


Figure 5. Overview of the proposed method. We begin by constructing a contour-aware character graph from the source character image, which is processed by a multi-scale graph encoder to extract multi-scale graph features. These features, along with character image features from an image encoder, are fed into a multi-scale aggregation decoder. The decoder autoregressively reconstructs the contour-aware trajectory sequence. This reconstructed trajectory sequence is then converted into a control sequence, enabling the robot to execute realistic human-like writing.

aggregate multi-scale structure information, ultimately generating the output $O_t \in \mathbb{R}^6$ (*i.e.,* the stroke parameters $(\hat{x}_t, \hat{y}_t, \hat{w}_t)$ and the pen state $(\hat{s}_t^1, \hat{s}_t^2, \hat{s}_t^3)$). The training loss for supervising the model comprises two parts: the stroke prediction loss $\mathcal{L}_{pre}$ and the pen state classification loss $\mathcal{L}_{cls}$:

$$\mathcal{L}_{pre} = L1\left(\hat{x}_t - x_t\right) + L1\left(\hat{y}_t - y_t\right) + L1\left(\hat{w}_t - w_t\right), \quad (6)$$

$$\mathcal{L}_{cls} = -\sum_{i=1}^{3} s_i \log \hat{s}_i, \quad (7)$$

where $p_t = \left(x_t, y_t, w_t, s_t^1, s_t^2, s_t^3\right)$ denotes the ground-truth point, and $L1\left(\cdot\right)$ denotes the L1 regression loss.

6. Experiment

6.1. Experimental Settings

Evaluation Dataset. We conduct experiments on our CHTR-110K dataset and popular CASIA-OLHWDB [25] dataset. To thoroughly explore the method's reconstruction performance across font and handwriting samples, we establish three testing scenarios on our CHTR-110K: a test set containing only font character samples, a test set containing only handwriting character samples, and a test set comprising both types. Notably, we use both font and handwriting samples for training. For the CASIA-OLHWDB dataset, we follow its standard splits.

We use mIOU [23] to measure the fidelity of stroke contours. Besides, we use Dynamic Time Warping (DTW) [2] to measure the distance between the generated and real trajectory sequence, and adopt LPIPS [6], to quantify the glyph fidelity of the generated characters. FID [16] and HWD [35] are employed to evaluate the quality of the generated results in terms of visual quality.

Implementation details. Implementation details. Images are resized to 256×256 for CHTR-110K and 64×64 for CASIA-OLHWDB [6, 24]. Our image encoder comprises a ResNet18 and 3 self-attention layers. The multi-scale graph encoder contains 4 graph blocks (each with 2 graph attention layers, $c = 512$, 8 heads), with the final three blocks outputting multi-scale features. We train the model for 300k iterations on an RTX 4090 GPU using Adam [19] (batch size 48, learning rate 10^{-4}, gradient clip 2.0). Following [15], variable-length trajectories are padded to the maximum dataset length N_{max} by setting $p_i = (0, 0, 0, 0, 0, 1)$ for $i > L$.

Compared Methods. We compare our method with state-of-the-art trajectory reconstruction approaches, including Cross-VAE [38], DED-Net [3], PEN-Net [6], and TrajFormer [24]. Since these methods inherently generate contour-agnostic trajectory sequences, we modify their official implementations by adjusting the input embedding and output projection layers to incorporate stroke width modeling. To ensure fair comparisons, all baselines are carefully tuned. As shown in Table 4, these contour-aware adaptations consistently outperform their original versions, confirming their optimal performance.

Method	Font					Handwriting					All				
	mIOU ↑	DTW ↓	FID ↓	LPIPS ↓	HWD ↓	mIOU ↑	DTW ↓	FID ↓	LPIPS ↓	HWD ↓	mIOU ↑	DTW ↓	FID ↓	LPIPS ↓	HWD ↓
Cross-VAE [38]	0.139	54.991	66.953	0.292	3.477	0.087	41.856	22.573	0.255	4.268	0.092	43.102	22.760	0.259	4.193
DED-Net [3]	0.177	64.317	24.251	0.289	2.055	0.162	31.106	3.864	0.223	1.856	0.163	34.243	3.964	0.229	1.875
PEN-Net [6]	0.180	58.387	26.615	0.287	2.069	0.140	33.288	4.794	0.232	1.877	0.144	35.666	4.893	0.237	1.896
TrajFormer [24]	0.643	19.273	7.548	0.106	1.626	0.542	17.657	1.472	0.083	1.513	0.552	18.277	1.475	0.092	1.525
Ours	**0.750**	**13.318**	**5.639**	**0.074**	**1.247**	**0.633**	**12.877**	**1.177**	**0.065**	**1.206**	**0.641**	**12.765**	**1.228**	**0.066**	**1.218**

Table 3. Comparisons with state-of-the-arts on CHTR-110K test set. "Font" and "Handwriting" represent the test set containing only font or handwriting character samples, respectively. "All" represents the complete test set that includes both types.

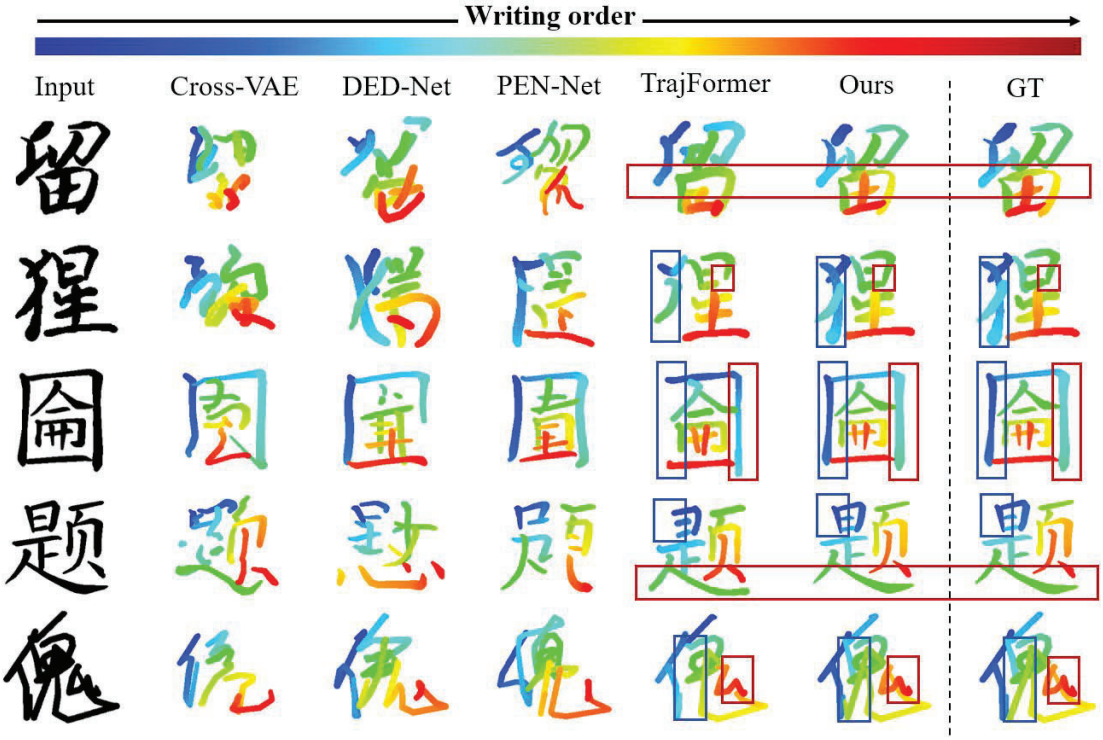


Figure 6. Qualitative comparison between our method and state-of-the-art approaches for contour-aware handwriting trajectory reconstruction on our CHTR-110K test set. The orders of handwriting trajectory are depicted in a gradient from blue to red. The blue boxes highlight the character structure integrity. The red boxes highlight comparisons between stroke details in the real and generated trajectories.

6.2. Main Results

Contour-aware handwriting trajectory reconstruction. Quantitative results are provided in Table 3, demonstrating that our G-HTR outperforms other methods across all evaluation metrics under different scenarios. Specifically, it improves mIOU by 16.64%(0.643 → 0.750) on the Font subset, 16.79%(0.542 → 0.633) on the Handwriting subset, and 16.12%(0.552 → 0.641) on the complete test set compared to the second-best. Moreover, our method achieves improvements of 27.69% (18.277 → 12.765) and 22.83% (0.092 → 0.066) over the second-best method in DTW and LPIPS on the complete test set. In terms of FID and HWD, our method achieves an improvement of approximately 16.75% for FID (1.475 → 1.228) and 20.13% for HWD (1.525 → 1.218) on the test set. These results highlight the superiority of our method in visual quality.

We provide qualitative results to explain the benefit of our G-HTR in Figure 6. Cross-VAE, DED-Net, and PEN-Net fail to maintain overall structure. TrajFormer struggles to maintain character structure and accurately reconstruct stroke details (cf. blue boxes and red boxes in Figure 6). In contrast, our G-HTR preserves both overall structure and fine stroke contours.

Conventional handwriting trajectory reconstruction. We evaluate our G-HTR's ability to reconstruct trajectory sequences, independent of stroke contours. We conduct experiments on CASIA-OLHWDB and use mIOU, DTW, and FID to assess the reconstructed trajectory sequence. Quantitative results are provided in Table 5. We observe that our G-HTR outperforms the compared methods in terms of mIOU, DTW, and FID, demonstrating its strong performance in conventional handwriting trajectory recovery.

Method	mIOU↑	DTW↓	FID↓
PEN-Net (original)	0.048	43.477	16.539
PEN-Net (modified)	**0.144**	**35.666**	**4.893**
TrajFormer (original)	0.104	33.869	7.879
TrajFormer (modified)	**0.552**	**18.277**	**1.475**

Table 4. Baseline modification on the CHTR-110K dataset. "original" denotes the official implementations, and "modified" denotes the modified implementations of input embedding and the output projection layers.

Method	mIOU ↑	DTW ↓	FID ↓
Cross-VAE [38]	0.049	57.125	23.660
DED-Net [3]	0.272	25.864	2.692
PEN-Net [6]	0.267	25.080	2.585
TrajFormer [24]	0.445	23.892	1.121
Ours	**0.530**	**16.264**	**1.050**

Table 5. Comparison with the state-of-the-art methods for conventional handwriting trajectory reconstruction on the public CASIA-OLHWDB [25] dataset.

Method	mIOU↑	DTW↓	FID↓
Base	0.532	19.228	1.642
Base+ε_G	0.596	15.512	1.346
Base+ε_G+MGL	**0.641**	**12.765**	**1.228**

Table 6. Ablation studies of the graph encoder (ε_G), and the multi-scale graph learning (MGL) strategy.

6.3. Analysis

In this section, we conduct ablation studies to analyze our G-HTR. We provide more analyses in Appendix, including discussions about the multi-scale graph learning, human-like robot writing, ethics, and more visualization results.

Ablation studies of graph encoder and multi-scale graph learning. We perform ablation experiments on our CHTR-110K dataset to validate the effect of the graph encoder and the multi-scale graph learning strategy. We provide quantitative results in Table 6. We can find that: (1) Introducing the graph encoder ε_{graph} significantly enhances all metrics, improving mIOU by 12.03% (0.532 → 0.596), DTW by 19.33% (19.228 → 15.512), and FID by 18.02% (1.642 → 1.346). This reveals that ε_{graph} contributes to improving reconstruction abilities. (2) Adding the multi-scale graph learning strategy further enhances reconstruction performance, with improvements of 7.55% in mIOU (0.596 → 0.641), 17.70% in DTW (15.512 → 12.765), and 8.76% in FID (1.346 → 1.228). Qualitative ablation analysis of the graph encoder and multi-scale graph learning is provided in Appendix F.

Failure cases analysis. While G-HTR accurately reconstructs character structures and stroke details, it occasionally predicts incorrect writing orders for rare characters outside the GB2312-80 set. This issue primarily stems from the scarcity of such complex topological structures in the training data, as illustrated in Figure 7. To resolve this, future work will involve expanding the dataset with diverse rare characters (*e.g.,* rare characters in the GB18030 character set) to improve model robustness.

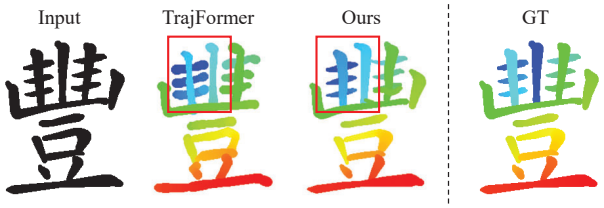
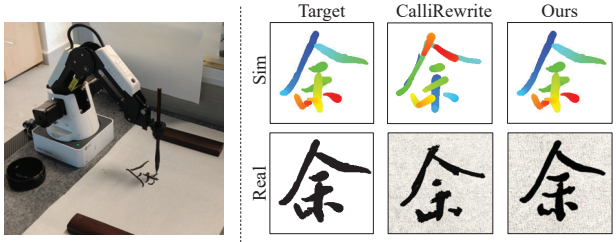


Figure 7. Failure cases analysis. The red boxes highlight the writing order errors.



(a) Calligraphy Robot (b) Comparisons of the Robot Rewriting

Figure 8. (a) A calligraphy robot is in the process of reproducing a character. (b) The first row represents the human writing order and the reconstructed contour-aware trajectory. The second row represents the real character image and the robot rewriting results. Please refer to the supplementary for "Robot Rewriting" video demos. More qualitative comparisons are provided in Appendix N.

Application to realistic calligraphy robot. We explore the practical applications of our G-HTR for calligraphy robots. As shown in Figure 8, the trajectory sequences from CalliRewrite [27] lead to unsatisfactory writing results due to their failure to follow the natural stroke order. In contrast, our method generates trajectory sequences that follow the natural writing order and preserve the character structure and stroke details, enabling the calligraphy robot to perform human-like writing and accurately reproduce the characters. Further discussions about human-like robot writing, along with implementation details, are provided in Appendix I.

7. Conclusion

We introduce Contour-aware Handwriting Trajectory Reconstruction (CHTR) to generate dynamic, visually realistic human-like handwriting. To advance this task, we construct CHTR-110K, a large-scale dataset comprising 110,540 annotated trajectory sequences. Technically, we propose G-HTR, a novel graph-based method leveraging multi-scale graphs to explicitly model character topologies and intricate stroke details. Extensive experiments coupled with real-world robotic deployments validate the superiority and practical value of our approach.

Acknowledgments This work was supported by the National Key Research and Development Program of China (No.2023YFC3502900), the National Natural Science Foundation of China (Nos.62176093, 61673182 and 62576139), and the Guangdong Emergency Management Science and Technology Program (No.2025YJKY001).

References

[1] Taylor Archibald, Mason Poggemann, Aaron Chan, and Tony Martinez. Trace: a differentiable approach to line-level stroke recovery for offline handwritten text. In *International Conference on Document Analysis and Recognition*, pages 414–429. Springer, 2021. 3

[2] Donald J Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *Proceedings of the 3rd international conference on knowledge discovery and data mining*, pages 359–370, 1994. 6

[3] Ayan Kumar Bhunia, Abir Bhowmick, Ankan Kumar Bhunia, Aishik Konwer, Prithaj Banerjee, Partha Pratim Roy, and Umapada Pal. Handwriting trajectory recovery using end-to-end deep encoder-decoder network. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 3639–3644. IEEE, 2018. 3, 6, 7, 8

[4] Giuseppe Boccignone, Angelo Chianese, Luigi P Cordella, and Angelo Marcelli. Recovering dynamic information from static handwriting. *Pattern recognition*, 26(3):409–418, 1993. 3

[5] Tianshui Chen, Jianman Lin, Zhijing Yang, Chumei Qing, Yukai Shi, and Liang Lin. Contrastive decoupled representation learning and regularization for speech-preserving facial expression manipulation. *International Journal of Computer Vision*, 133(7):3822–3838, 2025. 5

[6] Zhounan Chen, Daihui Yang, Jinglin Liang, Xinwu Liu, Yuyi Wang, Zhenghua Peng, and Shuangping Huang. Complex handwriting trajectory recovery: Evaluation metrics and algorithm. In *Proceedings of the asian conference on computer vision*, pages 1060–1076, 2022. 3, 6, 7, 8

[7] Gang Dai, Yifan Zhang, Qingfeng Wang, Qing Du, Zhuliang Yu, Zhuoman Liu, and Shuangping Huang. Disentangling writer and character styles for handwriting generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5977–5986, 2023. 1, 3

[8] Gang Dai, Yifan Zhang, Quhui Ke, Qiangya Guo, and Shuangping Huang. One-dm: One-shot diffusion mimicker for handwritten text generation. In *European Conference on Computer Vision*, pages 410–427. Springer, 2024.

[9] Gang Dai, Yifan Zhang, Yutao Qin, Qiangya Guo, Shuangping Huang, and Shuicheng Yan. Beyond isolated words: Diffusion brush for handwritten text-line generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19054–19064, 2025. 3

[10] Inderjit S Dhillon, Yuqiang Guan, and Brian Kulis. Weighted graph cuts without eigenvectors a multilevel approach. *IEEE transactions on pattern analysis and machine intelligence*, 29(11):1944–1957, 2007. 5

[11] Ji Gan, Yuyan Chen, Bo Hu, Jiaxu Leng, Weiqiang Wang, and Xinbo Gao. Characters as graphs: Interpretable handwritten chinese character recognition via pyramid graph transformer. *Pattern Recognition*, 137:109317, 2023. 5

[12] Shengjie Gong, Wenjie Peng, Hongyuan Chen, Gangyu Zhang, Yunqing Hu, Huiyuan Zhang, Shuangping Huang, and Tianshui Chen. Learning hierarchical and geometry-aware graph representations for text-to-cad. In *The Fourteenth International Conference on Learning Representations*. 5

[13] Shengjie Gong, Haojie Li, Jiapeng Tang, Dongming Hu, Shuangping Huang, Hao Chen, Tianshui Chen, and Zhuoman Liu. Monocular and generalizable gaussian talking head animation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5523–5534, 2025. 5

[14] Qiangya Guo, Gang Dai, Zhuoman Liu, Shuangping Huang, Yunqing Hu, Huiyuan Zhang, and Tianshui Chen. Plan then act: Bi-level cad command sequence generation. In *The Fourteenth International Conference on Learning Representations*. 3

[15] David Ha and Douglas Eck. A neural representation of sketch drawings. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. 6

[16] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 6

[17] Shuming Hu, Na Wang, Kaixing Zhao, Yao Jing, Ying Zhang, Zhu Wang, Bin Guo, and Zhiwen Yu. Masterpiece creation: An ai-powered robotic calligraphy creation system. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 9(3):1–34, 2025. 1

[18] Muwei Jian, Junyu Dong, Maoguo Gong, Hui Yu, Liqiang Nie, Yilong Yin, and Kin-Man Lam. Learning the traditional art of chinese calligraphy via three-dimensional reconstruction and assessment. *IEEE Transactions on Multimedia*, 22 (4):970–979, 2019. 1

[19] Diederik Kinga, Jimmy Ba Adam, et al. A method for stochastic optimization. In *International conference on learning representations (ICLR)*. California;, 2015. 6

[20] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 5

[21] Florian Kleber, Stefan Fiel, Markus Diem, and Robert Sablatnig. Cvl-database: An off-line database for writer retrieval, writer identification and word spotting. In *2013 12th international conference on document analysis and recognition*, pages 560–564. IEEE, 2013. 4

[22] Kai Kwong Lau, Pong C Yuen, and Yuan Y Tang. Directed connection measurement for evaluating reconstructed stroke sequence in handwriting images. *Pattern Recognition*, 38(3): 323–339, 2005. 3

[23] Meng Li, Yahan Yu, Yi Yang, Guanghao Ren, and Jian Wang. Stroke extraction of chinese character based on deep

structure deformable image registration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1360–1367, 2023. 3, 6

[24] Junxiang Lin, Zhounan Chen, Lingyu Liang, Wenjie Peng, and Shuangping Huang. Handwriting trajectory recovery via trajectory transformer with global radical context-aware module. In *International Conference on Pattern Recognition*, pages 182–195. Springer, 2024. 3, 5, 6, 7, 8

[25] Cheng-Lin Liu, Fei Yin, Da-Han Wang, and Qiu-Feng Wang. Casia online and offline chinese handwriting databases. In *2011 international conference on document analysis and recognition*, pages 37–41. IEEE, 2011. 2, 3, 4, 6, 8

[26] Marcus Liwicki and Horst Bunke. 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)*, pages 956–961. IEEE, 2005. 3, 4

[27] Yuxuan Luo, Zekun Wu, and Zhouhui Lian. Callirewrite: Recovering handwriting behaviors from calligraphy images without supervision. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8671–8678. IEEE, 2024. 5, 8

[28] U-V Marti and Horst Bunke. The iam-database: an english sentence database for offline handwriting recognition. *International journal on document analysis and recognition*, 5(1): 39–46, 2002. 2

[29] Tomohisa Matsushita and Masaki Nakagawa. A database of on-line handwritten mixed objects named" kondate". In *2014 14th International Conference on Frontiers in Handwriting Recognition*, pages 369–374. IEEE, 2014. 3, 4

[30] Haoran Mo, Edgar Simo-Serra, Chengying Gao, Changqing Zou, and Ruomei Wang. General virtual sketching framework for vector line art. *ACM Transactions on Graphics (TOG)*, 40(4):1–14, 2021. 4

[31] Hung Tuan Nguyen, Tsubasa Nakamura, Cuong Tuan Nguyen, and Masaki Nakawaga. Online trajectory recovery from offline handwritten japanese kanji characters of multiple strokes. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 8320–8327. IEEE, 2021. 3

[32] Konstantina Nikolaidou, George Retsinas, Giorgos Sfikas, and Marcus Liwicki. Diffusionpen: Towards controlling the style of handwritten text generation. In *European Conference on Computer Vision*, pages 417–434. Springer, 2024. 3

[33] Wei Pan, Anna Zhu, Xinyu Zhou, Brian Kenji Iwana, and Shilin Li. 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*, pages 19506–19516, 2023. 1

[34] Mario Pechwitz and V Margner. Baseline estimation for arabic handwritten words. In *Proceedings eighth international workshop on frontiers in handwriting recognition*, pages 479–484. IEEE, 2002. 3

[35] Vittorio Pippi, Fabio Quattrini, Silvia Cascianelli, and Rita Cucchiara. Hwd: A novel evaluation score for styled handwritten text generation. *arXiv preprint arXiv:2310.20316*, 2023. 6

[36] Yu Qiao and Makoto Yasuhara. Recover writing trajectory from multiple stroked image using bidirectional dynamic search. In *18th International Conference on Pattern Recognition (ICPR'06)*, pages 970–973. IEEE, 2006. 3

[37] Min-Si Ren, Yan-Ming Zhang, Qiu-Feng Wang, Fei Yin, and Cheng-Lin Liu. Diff-writer: a diffusion model-based stylized online handwritten chinese character generator. In *International Conference on Neural Information Processing*, pages 86–100. Springer, 2023. 1

[38] Taichi Sumi, Brian Kenji Iwana, Hideaki Hayashi, and Seiichi Uchida. Modality conversion of handwritten patterns by cross variational autoencoders. In *2019 international conference on document analysis and recognition (ICDAR)*, pages 407–412. IEEE, 2019. 6, 7, 8

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 3

[40] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, Yoshua Bengio, et al. Graph attention networks. *stat*, 1050(20):10–48550, 2017. 5

[41] Lei Wang, Yu Liu, Mohd Yunus Sharum, Razali Bin Yaako, Khairul Azhar Kasmiranm, and Cunrui Wang. Deep learning for chinese font generation: A survey. *Expert Systems with Applications*, page 127105, 2025. 1

[42] Zhenhua Yang, Dezhi Peng, Yuxin Kong, Yuyi Zhang, Cong Yao, and Lianwen Jin. Fontdiffuser: One-shot font generation via denoising diffusion with multi-scale content aggregation and style contrastive learning. In *Proceedings of the AAAI conference on artificial intelligence*, pages 6603–6611, 2024. 1

[43] Fei Yin, Qiu-Feng Wang, Xu-Yao Zhang, and Cheng-Lin Liu. Icdar 2013 chinese handwriting recognition competition. In *2013 12th international conference on document analysis and recognition*, pages 1464–1470. IEEE, 2013. 2, 4

[44] Hang Yin, Patrícia Alves-Oliveira, Francisco S Melo, Aude Billard, and Ana Paiva. Synthesizing robotic handwriting motion by learning from human demonstrations. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, pages 3530–3537, 2016. 1

[45] Tongjie Y Zhang and Ching Y. Suen. A fast parallel algorithm for thinning digital patterns. *Communications of the ACM*, 27(3):236–239, 1984. 5

[46] Xu-Yao Zhang, Fei Yin, Yan-Ming Zhang, Cheng-Lin Liu, and Yoshua Bengio. Drawing and recognizing chinese characters with recurrent neural network. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):849–862, 2017. 1

[47] Yixin Zhao, Yuyi Zhang, and Lianwen Jin. Mccd: A multi-attribute chinese calligraphy character dataset annotated with script styles, dynasties, and calligraphers. *arXiv preprint arXiv:2507.06948*, 2025. 3, 4

[48] Zhuoyao Zhong, Lianwen Jin, and Ziyong Feng. Multi-font printed chinese character recognition using multi-pooling convolutional neural network. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 96–100. IEEE, 2015. 3, 4

[49] Yuanping Zhu, Shengnan Li, Hui Wang, and Feilong Wei. Finet: Handwriting trajectory reconstruction of chinese characters based on the font imitate network. *Pattern Recognition*, 157:110949, 2025. 3, 5