

Towards Human-Like Robot Handwriting via Contour-Aware Generation

Supplementary Material

We organize our supplementary material as follows.

- In Section A, we discuss the more related works, including handwriting generation and graph neural network.
- In Section B, we describe more implementation details of the method.
- In Section C, we provide more details about the dataset construction process.
- In Section D, we provide more details of evaluation metrics.
- In Section E, we describe more details of ablation studies.
- In Section F, we provide the qualitative evaluation of the proposed graph encoder and the multi-scale graph learning strategy.
- In Section G, we analyze the proposed multi-scale graph learning strategy.
- In Section H, we study the effect of the order of the multi-scale graphs input.
- In Section I, we discuss the related elements of human-like robot handwriting.
- In Section J, we provide quantitative evaluations of robot writing.
- In Section K, we discuss the ethical implications of this technology.
- In Section L, we provide more example visualizations.
- In Section M, we show more qualitative comparisons of contour-aware handwriting trajectory reconstruction.
- In Section N, we show more visual comparisons of robot rewriting results.

A. More related work

Handwriting Generation Early generation methods primarily use Recurrent Neural Networks (RNNs) [12, 23, 26] and Generative Adversarial Networks (GANs) [2, 11, 19]. These methods condition the generation process on the style and content conditions. In recent years, some methods adopt Transformer [6, 15, 24] or diffusion models [7, 18, 20] to generate realistic handwritings. Specifically, SDT [6] disentangles writer-wise and character-wise style features and combines them with content conditions to generate handwriting through the Transformer decoder. Some diffusion-based methods, like One-DM [7], DiffBrush [8], and DiffusionPen [18], guide the denoising process by the merged style and content features.

However, these handwriting generation methods struggle to generalize to our trajectory reconstruction task because they require both style and content conditions as inputs. In contrast, our task requires only a character image input to achieve accurate trajectory reconstruction without provid-

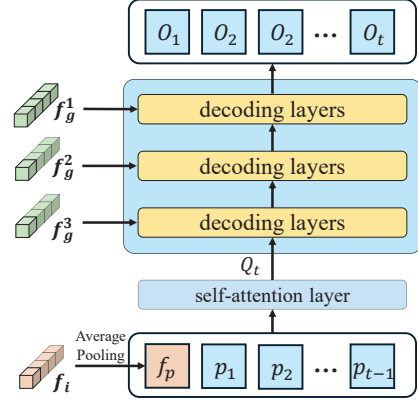


Figure 1. Illustration of the multi-scale aggregation decoder. At each time step t , the query vector Q_t is encoded from the pooled character image feature f_p and previous points $\{p_j\}_{j=1}^{t-1}$. Then, it successively attends to multi-scale graph features $F_g = \{f_g^1, f_g^2, f_g^3\}$ for predicting the current-step output.

ing separated style and content conditions.

Graph Neural Network Recently, graph neural networks (GNNs) have been applied in many fields [5, 10, 27]. In terms of image generation, they include scene graph generation [3], 3D point cloud generation [21], medical image generation [9], etc. Different from them, we are the first to apply GNNs to trajectory reconstruction to capture the structural relationships among stroke segments.

B. More details of multi-scale decoder

To accurately reconstruct contour-aware trajectory sequences of the character, we integrate the character image feature f_i and the multi-scale graph features $F_g = \{f_g^1, f_g^2, f_g^3\}$ using multi-head attention layers in our multi-scale aggregation decoder.

As shown in Figure 1, we take the pooled character image feature $f_p \in \mathbb{R}^c$ as the initial point, where c is the channel dimension. Then, we employ a self-attention layer to the character context $[f_p, p_1, p_2, \dots, p_{t-1}]$ and obtain the query vector Q_t . Subsequently, Q_t serves as the query vector that successively attends to $f_g^1 \in \mathbb{R}^{N_1 \times c_1}$, $f_g^2 \in \mathbb{R}^{N_2 \times c_2}$, and $f_g^3 \in \mathbb{R}^{N_3 \times c_3}$ to 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 $(\hat{s}_t^1, \hat{s}_t^2, \hat{s}_t^3)$), where N_1, N_2, N_3 is the number of nodes, the dimensional state c_1 is 128, c_2 is 256, c_3 is 512.

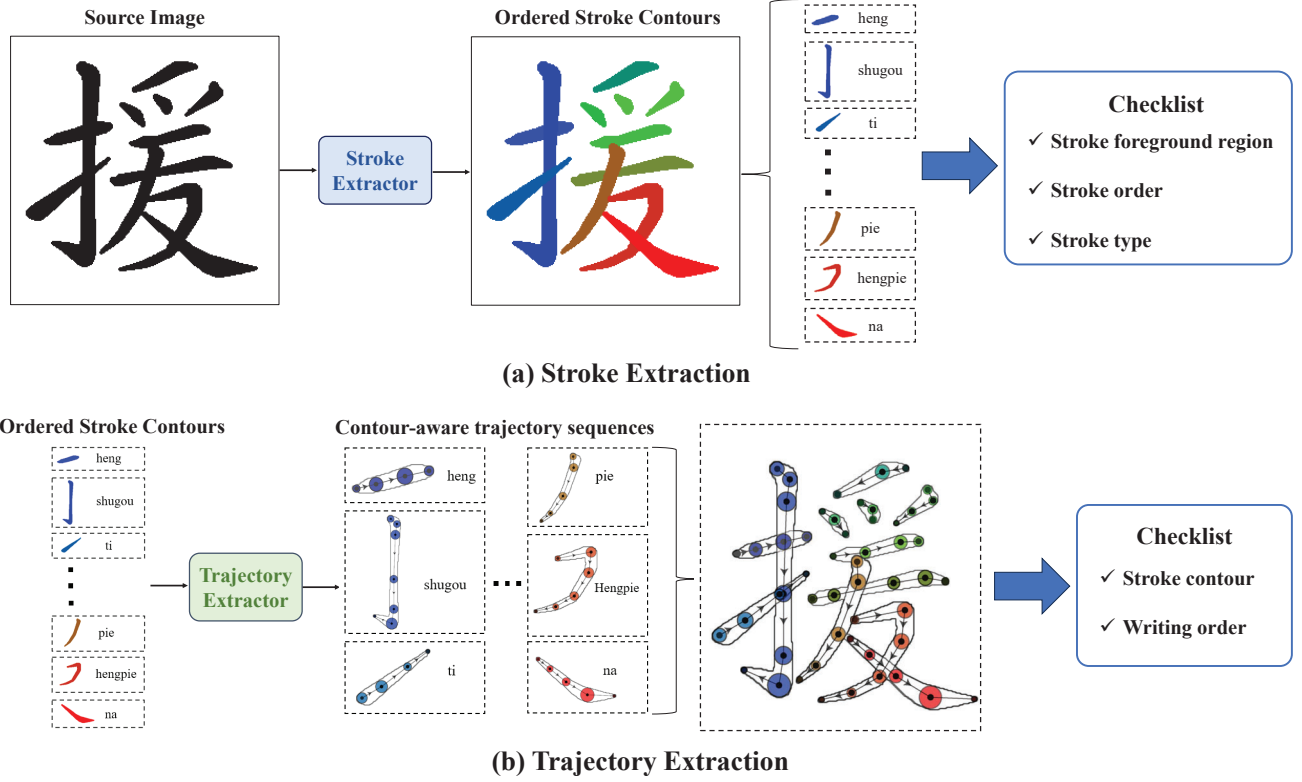


Figure 2. Two-stage semi-automatic annotation pipeline for constructing the CHTR-110K dataset.

C. More details of data construction

We provide more implementation details of the dataset construction pipeline to illustrate how to achieve effective annotation and ensure the quality of the dataset. The proposed dataset construction pipeline is a two-stage semi-automatic annotation process. It involves stroke extraction and trajectory extraction as distinct stages.

C.1. Stroke extraction

As shown in Figure 2(a), given a character image, stroke extraction aims to extract the stroke foreground region along with its stroke order and stroke type. Specifically, the stroke type is directly inferred from the class index predicted by SDNet [13] (across 25 categories). The stroke order is determined by matching the spatial positions and types of predicted regions with the standard stroke composition of the target character. Finally, we extract precise foreground regions for the ordered stroke contours. Here, we denote all extracted stroke foreground regions as $S = \{S_i\}_{i=1}^n$, where n represents the total number of strokes.

After acquiring all extracted stroke regions, we perform manual inspection and correction to ensure: (1) accurate segmentation of each stroke foreground region; (2) correct prediction of stroke order; (3) correct prediction of stroke type. For the first criterion, we adjust the extracted stroke

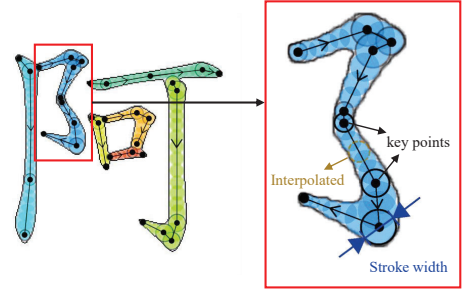


Figure 3. Illustration of the stroke render process. The diameter and center of the black circles represent the stroke widths and key points in the trajectory sequence, respectively. We further interpolate the key points and draw additional circles to better preserve the authenticity of the rendered stroke.

regions to ensure they align with a single stroke. For the second and third criteria, we refer to the standard writing rules of the target character to correct any errors in the stroke order and type.

C.2. Trajectory extraction

The goal of Trajectory extraction is to derive contour-aware trajectory sequences from the stroke foreground region. As shown in Figure 2(b), given the foreground region and type of the i -th stroke, we employ a trajectory extractor [17] to extract the contour-aware trajectory sequence P_i .

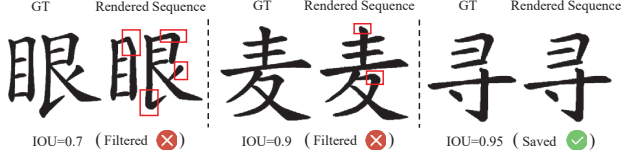


Figure 4. Sample filtering based on its IOU.

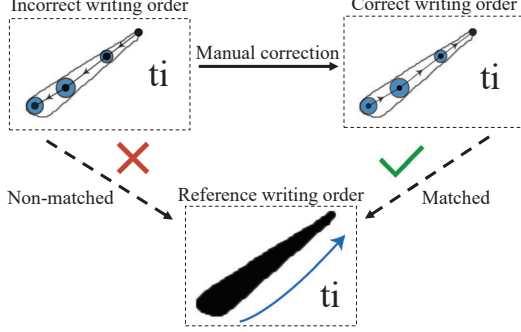


Figure 5. Illustration of verifying and adjusting the order of key points within the trajectory sequence.

After acquiring all trajectory sequences, we need to ensure: (1) the trajectory sequence accurately reproduces the stroke contour; (2) based on the corresponding stroke type, the order of key points within the trajectory sequence follows the natural writing order.

For the first criterion, we calculate the intersection-over-union (IOU) between \hat{S}_i and S_i , where \hat{S}_i is rendered by P_i (cf. Figure 3). Then we filtered out low-quality samples with $IOU(\hat{S}_i, S_i) < 0.95$ to achieve the required accuracy in reproducing the stroke contour (cf. Figure 4). For the second criterion, we manually verify and adjust the order of key points within the trajectory sequence to ensure consistency with the natural writing order (cf. Figure 5).

Carrying out the inspections and corrections is a meticulous and challenging process. We recruit 5 volunteers with undergraduate degrees who possess the necessary expertise for post-processing steps. The entire process totals approximately 1,500 person-hours.

D. Implementation details of metrics

mIOU. We use mIOU [13] to measure the fidelity and the order correctness at stroke-level. Specifically, we extract trajectories of individual strokes from online characters using the pen state (s^1, s^2, s^3) and render them as single-stroke images, and sequentially calculate the similarity between the generated and target single-stroke images. This calculation can be formulated as follows:

$$mIOU = \sum_{i=0}^n IOU(rt_s^i, t_s^i), \quad (1)$$

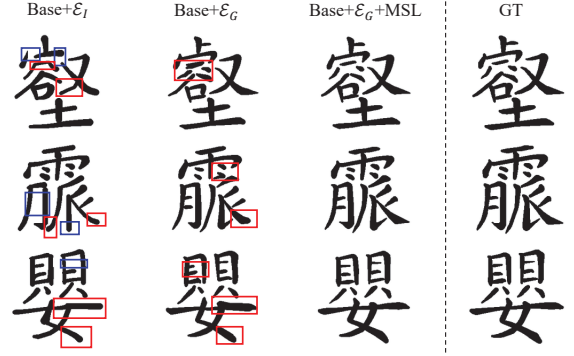


Figure 6. Qualitative ablation studies. ε_G denotes the graph encoder. MGL denotes the multi-scale graph learning strategy. The blue boxes highlight overall character structure errors, while the red boxes highlight failures in preserving stroke details.

where n denotes the number of strokes in the target character. rt_s denotes all generated single-stroke images. t_s denotes all target single-stroke images. rt_s^i denotes the i -th generated single-stroke image. t_s^i denotes the i -th target single-stroke image. We set $IOU(rt_s^i, t_s^i)$ to 0 if the number of rt_s is less than n .

DTW. For a more robust evaluation metric, we follow [25] to normalize the absolute 2D coordinates of trajectory points into a standard interval to eliminate the effects of different lengths.

LPIPS, FID and HWD. We render both the generated and target trajectory sequences into character images and then evaluate the glyph fidelity and visual quality between the two sets of characters.

E. Implementation details of ablation studies

Base. In this setting, we utilize the character image features f_i as key/value vectors and feed them into a decoder comprising three Transformer layers to obtain the contour-aware trajectory sequence.

Base+ ε_G . We replace the character image feature f_i with the single-scale graph feature f_g^3 as key/value vectors and feed them into a decoder comprising three Transformer layers to obtain the contour-aware trajectory sequence.

Base+ ε_{graph} +MGL. As shown in Figure 1, we obtain multi-scale graph features from the multi-scale graph encoder, and feed them to the multi-scale aggregation decoder.

F. Qualitative ablation analysis

To further analyze each module in our G-HTR, we conduct visual ablation experiments. As shown in Figure 6, we observe that the introduction of the graph encoder allows for precise reconstruction of the overall character structure. However, using only the single-scale graph feature still presents challenges in preserving stroke details. Finally, the multi-scale graph learning strategy further helps the learning of fine-grained stroke details.

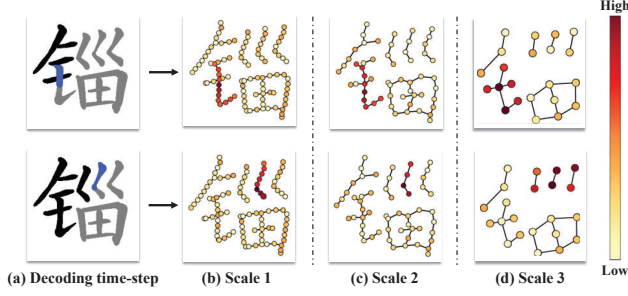


Figure 7. Visualization of the multi-scale graph attention under different decoding time-steps. (a) The blue segments in the image represent the input stroke at the current time step. (b)-(d) The node color transition from light to dark in the multi-scale graphs indicates the attention weights that the model pays to each node.

Order	mIOU \uparrow	DTW \downarrow	FID \downarrow
$f_g^1 - f_g^2 - f_g^3$	0.626	14.144	1.322
$f_g^3 - f_g^2 - f_g^1$	0.641	12.765	1.228

Table 1. Effect of multi-scale graph features (f_g^1, f_g^2, f_g^3) input order on CHTR-110K. f_g^1 represents low-level features extracted from the shallow graph block, while f_g^3 represents high-level features extracted from the deep graph block.

G. Analysis of multi-scale graph learning

To intuitively demonstrate the effectiveness of the proposed multi-scale graph learning strategy, we visualize the attention weights across multi-scale graphs at different decoding time steps. In Scale 1 of Figure 7(b), the model focuses on adjacent nodes within the input stroke to capture intricate stroke details. While in Scale 3 of Figure 7(d), it attends to the input stroke along with its neighboring strokes to capture coarse structural features. These findings underscore the advantages of exploiting multi-scale graph features, enabling the model to effectively learn character features from fine-grained details to coarse structures, facilitating accurate and consistent trajectory reconstruction.

H. Multi-scale graph features input order

To investigate the effect of the input order of multi-scale graph features, we conduct two independent experiments on CHTR-110K. As shown in Table 1, we can observe that prioritizing high-level features (i.e., $f_g^3 - f_g^2 - f_g^1$) outperforms prioritizing low-level features (i.e., $f_g^1 - f_g^2 - f_g^3$). This is because high-level features, which capture coarse-grained character structures, help the model establish a global understanding of the character. Subsequently, incorporating low-level features that emphasize fine-grained stroke information enables the model to further refine local character details. Therefore, this global-to-local fusion order leads to superior character reconstruction performance.

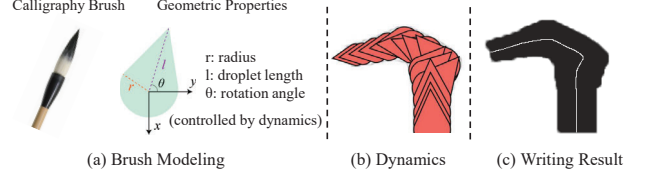


Figure 8. Brush modeling and Dynamics modeling.

Method	IOU \uparrow	CD \downarrow	LPIPS \downarrow
CalliRewrite	0.389	3.650	0.349
TrajFormer	0.419	2.304	0.336
Ours	0.540	1.310	0.327

Table 2. Comparison with the state-of-the-art methods on robot rewriting.

I. Discussions about human-like calligraphy

To achieve human-like calligraphy, some elements like stroke order, pressure, anisotropic strokes, dynamics, dwell time, and ink effects must be considered. In our work, we explicitly consider most of the dimensions: 1) Our G-HTR inherently models *stroke order* and *pressure*. 2) To physically draw the predicted trajectory points in the real world, *anisotropic strokes* and *brush dynamics* should be introduced. To bridge this gap, we adopt a droplet-shaped brush model [22] to produce *anisotropic strokes* and an RL pipeline [22] to model *brush dynamics*. Each anisotropic stroke contains three geometric parameters (cf. Figure 8(a)) to control stroke direction and stroke shape. The RL pipeline then optimizes the evolution of these parameters, thereby achieving realistic dynamics (cf. Figure 8(b)). These are finally translated into control sequences for robot handwriting (cf. Figure 8(c)). 4) For *dwell time*, our robot adopts a density-aware control mechanism, allowing a higher point density to naturally mimic human-like pauses at critical stroke junctions. Overall, with the predicted contour-aware trajectories, we advance structural and dynamic writing realism compared to prior works. While *ink effects* (e.g., bleeding) remain a limitation due to intricate fluid-paper interactions, we leave them for future work.

J. Quantitative evaluation of robot writing

We first generate writing contour-aware trajectory sequences and convert them into control sequences for robotic writing. To quantitatively evaluate their performance, we scan the physical writing outputs and process them into character images with a white background. Subsequently, we compute the Intersection Over Union (IOU), Chamfer Distance (CD), and LPIPS between these processed images and the ground-truth character images. The results in Table 2 indicate that our G-HTR achieves the best performance, confirming its superiority.

K. Ethics

Realistic imitation of human handwriting introduces potential security and ethical concerns, particularly regarding forgery and unauthorized signature generation. To prevent the improper exploitation of our method and dataset, we establish strict behavioral guidelines for their release and application. Specifically, we use restrictive licenses, such as CC BY-NC-SA, to prohibit commercial misuse. Furthermore, we apply digital watermarking to the generated trajectories and rendered images, which ensures traceability and helps distinguish machine-generated handwriting from authentic human scripts. These proactive measures will mitigate potential risks.

L. More dataset example demonstrations

We present more example visualizations in Figure 9, our CHTR-110K dataset contains 110,540 contour-aware trajectory sequences annotated with trajectory key points, stroke order and contours. Each trajectory sequence consistently maintains the integrity of the character structure across various styles while preserving the detailed stroke contours. The proposed dataset is of significant importance for the development of technologies related to contour-aware handwriting trajectory reconstruction.

M. More qualitative comparisons of CHTR

We provide more qualitative comparisons on contour-aware handwriting trajectory reconstruction in Figure 10 between our proposed G-HTR and the previous state-of-the-art works, i.e., Cross-VAE [22], DED-Net [1], PEN-Net [4], and TrajFormer [14]. Figure 10 shows that Cross-VAE, DED-Net, and PEN-Net fail to keep the character structure integrity. TrajFormer struggles to maintain character structure and the precise reconstruction of stroke details (cf. blue and red boxes in Figure 10). Conversely, our method is capable of capturing the fine-grained stroke details while preserving the correct writing order and overall character structure, resulting in more accurate and consistent contour-aware trajectory reconstruction.

N. More comparisons of robot writing

From Figure 11 and Figure 12, we provide more robot rewriting comparisons between our proposed G-HTR with CalliRewrite [16] and TrajFormer [14]. From these results, we can observe that CalliRewrite generates discontinuous strokes to reproduce the character images, which do not align with the natural human writing order and lead to unsatisfactory results in robot rewriting. TrajFormer can produce a relatively complete character structure but fails to accurately reconstruct stroke details. In contrast, our G-HTR precisely reconstructs contour-aware handwriting tra-

jectories, maintaining the correct writing order and complete character structure while preserving stroke details.

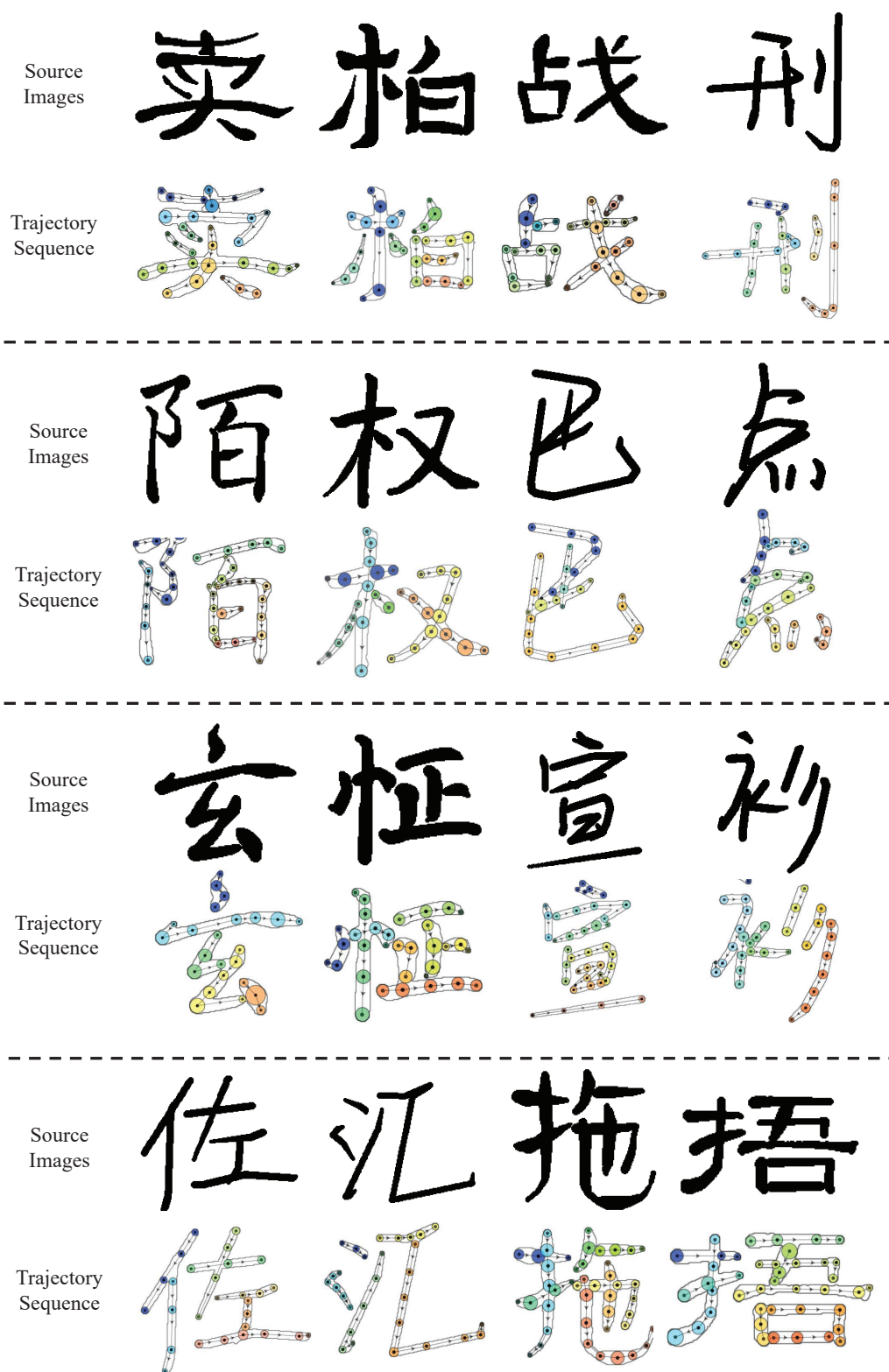


Figure 9. Sample visualizations of our CHTR-110K dataset. The points within the strokes are denoted as the key points of the trajectory sequences, with arrows indicating the writing order and the diameter of each circle denoting the stroke width. Better zoom in 200%.

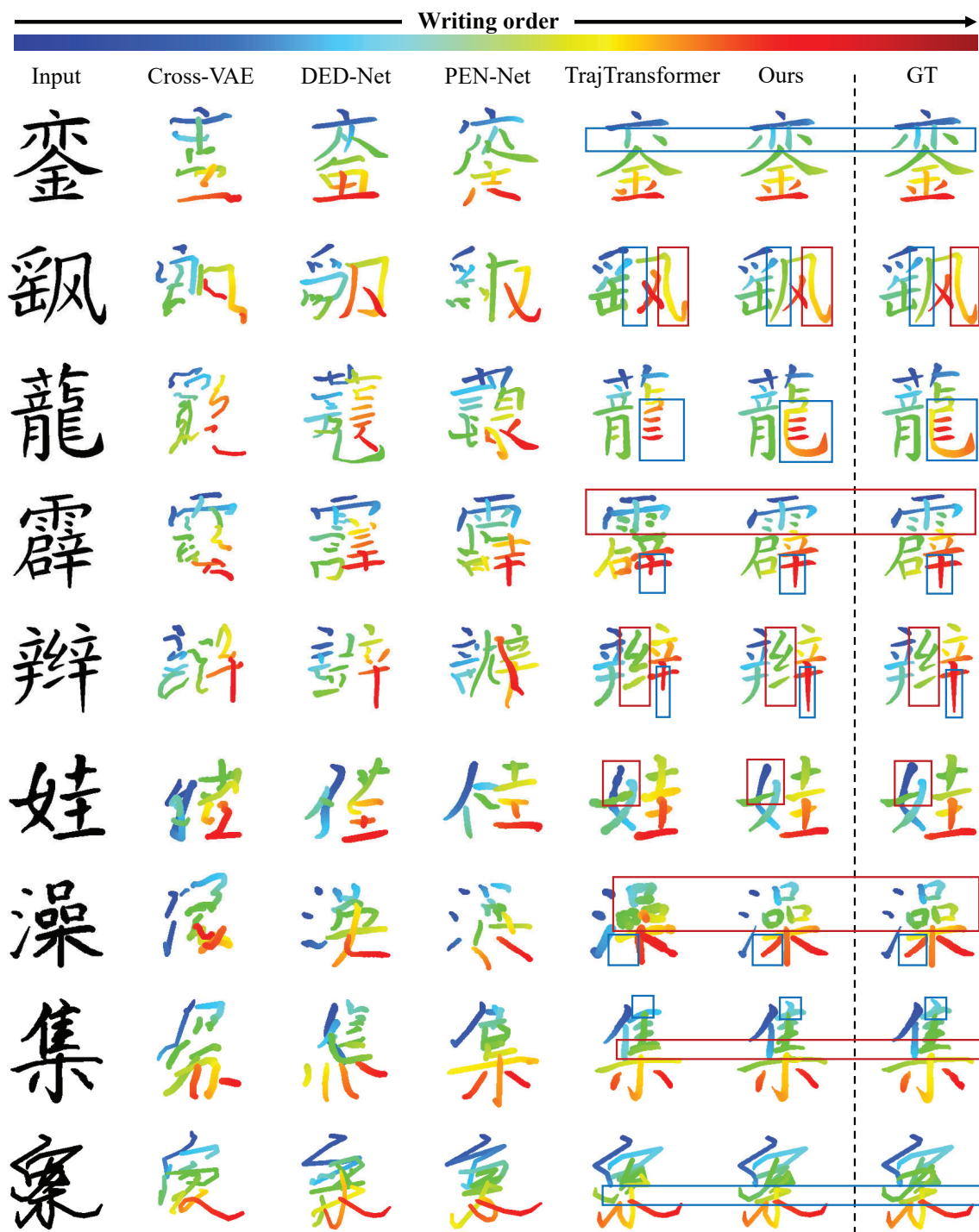


Figure 10. Additional comparisons between our method with state-of-the-art methods on contour-aware handwriting trajectory reconstruction. The orders of writing trajectory are depicted in a gradient from blue to red. The blue boxes highlight the character structure integrity. The red boxes highlight comparisons between stroke details in the real and generated trajectories.



Figure 11. Comparisons on robotic demonstration. Each row of “Sim” showcases the trajectories of the target and the generated character, and “Real” represents the real character image and the robot rewriting results on different methods.



Figure 12. Comparisons on robotic demonstration. Each row of “Sim” showcases the trajectories of the target and the generated character, and “Real” represents the real character image and the robot rewriting results on different methods.

References

- [1] 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. [5](#)
- [2] Ankan Kumar Bhunia, Salman Khan, Hisham Cholakkal, Rao Muhammad Anwer, Fahad Shahbaz Khan, and Mubarak Shah. Handwriting transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1086–1094, 2021. [1](#)
- [3] Xiaojun Chang, Pengzhen Ren, Pengfei Xu, Zhihui Li, Xiaojiang Chen, and Alex Hauptmann. A comprehensive survey of scene graphs: Generation and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1): 1–26, 2021. [1](#)
- [4] 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. [5](#)
- [5] Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, and Yanwen Guo. Multi-label image recognition with graph convolutional networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5177–5186, 2019. [1](#)
- [6] 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](#)
- [7] 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. [1](#)
- [8] 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. [1](#)
- [9] Kexin Ding, Mu Zhou, Zichen Wang, Qiao Liu, Corey W Arnold, Shaoting Zhang, and Dimitri N Metaxas. Graph convolutional networks for multi-modality medical imaging: Methods, architectures, and clinical applications. *arXiv preprint arXiv:2202.08916*, 2022. [1](#)
- [10] Danfeng Hong, Lianru Gao, Jing Yao, Bing Zhang, Antonio Plaza, and Jocelyn Chanussot. Graph convolutional networks for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 59(7):5966–5978, 2020. [1](#)
- [11] Lei Kang, Pau Riba, Yaxing Wang, Marçal Rusinol, Alicia Fornés, and Mauricio Villegas. Ganwriting: content-conditioned generation of styled handwritten word images. In *European conference on computer vision*, pages 273–289. Springer, 2020. [1](#)
- [12] Atsunobu Kotani, Stefanie Tellex, and James Tompkin. Generating handwriting via decoupled style descriptors. In *European Conference on Computer Vision*, pages 764–780. Springer, 2020. [1](#)
- [13] 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. [2](#), [3](#)
- [14] 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. [5](#)
- [15] Yu Liu, Fatimah Binti Khalid, Lei Wang, Youxi Zhang, and Cunrui Wang. Elegantly written: Disentangling writer and character styles for enhancing online chinese handwriting. In *European Conference on Computer Vision*, pages 409–425. Springer, 2024. [1](#)
- [16] 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](#)
- [17] 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. [2](#)
- [18] 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. [1](#)
- [19] Vittorio Pippi, Silvia Cascianelli, and Rita Cucchiara. Handwritten text generation from visual archetypes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22458–22467, 2023. [1](#)
- [20] 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](#)
- [21] Dong Wook Shu, Sung Woo Park, and Junseok Kwon. 3d point cloud generative adversarial network based on tree structured graph convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3859–3868, 2019. [1](#)
- [22] Taichi Sumi, Brian Kenji Iwana, Hideaki Hayashi, and Seichi 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. [5](#)
- [23] Shusen Tang and Zhouhui Lian. Write like you: Synthesizing your cursive online chinese handwriting via metric-based meta learning. In *Computer Graphics Forum*, pages 141–151. Wiley Online Library, 2021. [1](#)
- [24] Carmine Zaccagnino, Fabio Quattrini, Vittorio Pippi, Silvia Cascianelli, Alessio Tonioni, and Rita Cucchiara. Autoregressive styled text image generation, but make it reliable. *arXiv preprint arXiv:2510.23240*, 2025. [1](#)

- [25] 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. [3](#)
- [26] Bocheng Zhao, Jianhua Tao, Minghao Yang, Zhengkun Tian, Cunhang Fan, and Ye Bai. Deep imitator: Handwriting calligraphy imitation via deep attention networks. *Pattern Recognition*, 104:107080, 2020. [1](#)
- [27] Ling Zhao, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng, and Haifeng Li. T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE transactions on intelligent transportation systems*, 21(9):3848–3858, 2019. [1](#)