Bridging Vision, Language, and Mathematics: Pictographic Character Reconstruction with Bézier Curves

Zihao Wan[1]*, **Pau Tong Lin Xu**[1]*, **Fuwen Luo**[1], **Ziyue Wang**[1], **Peng Li**[2] ✉, **Yang Liu**[1,2] ✉

[1]Dept. of Comp. Sci. & Tech., Institute for AI, Tsinghua University, Beijing, China
[2]Institute for AI Industry Research (AIR), Tsinghua University, Beijing, China

Abstract

While Vision-language Models (VLMs) have demonstrated strong semantic capabilities, their ability to interpret the underlying geometric structure of visual information is less explored. Pictographic characters, which combine visual form with symbolic structure, provide an ideal test case for this capability. We formulate this visual recognition challenge in the mathematical domain, where each character is represented by an executable program of geometric primitives. This is framed as a program synthesis task, training a VLM to decompile raster images into programs composed of Bézier curves. Our model, acting as a "visual decompiler", demonstrates performance superior to strong zero-shot baselines, including GPT-4o. The most significant finding is that when trained solely on modern Chinese characters, the model is able to reconstruct ancient Oracle Bone Script in a zero-shot context. This generalization provides strong evidence that the model acquires an abstract and transferable geometric grammar, moving beyond pixel-level pattern recognition to a more structured form of visual understanding.

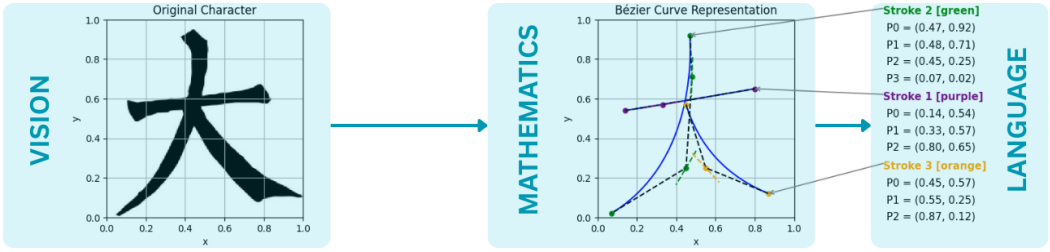

Figure 1: **Mathematics bridging vision and language.** A pictographic character can be represented using Bézier curves, each of which can then be written as a list of control points in text form.

1 Introduction

Vision-language Models (VLMs) have demonstrated strong capabilities in mapping images to high-level semantic descriptions [Bai et al., 2025, Team, 2024]. However, a more fundamental aspect of visual understanding — the ability to interpret and reconstruct the underlying structure of visual forms — has been less explored [Thrush et al., 2022]. Pictographic characters, which combine pictorial representation with a structured symbolic system, present an ideal test case for this capability, motivating our core question: *can VLMs' understanding generalize from semantic recognition to programmatic reconstruction?* We find that current models, including GPT-4o, are limited in this

*Equal contribution, ✉ Corresponding author

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: MATH-AI.

capacity (Figure 2). This difficulty suggests that current VLMs rely on surface-level correlations between pixel patterns and linguistic tokens, rather than parsing visual forms into geometric programs.

To address these limitations, we frame character reconstruction as a program synthesis problem, with the target output being a sequence of mathematical primitives — Bézier curves. This representation compels the model to move beyond pixel-level statistics and learn the geometric properties that define a character's form. Unlike localization methods such as bounding boxes [Girshick et al., 2014, Ren et al., 2015], a Bézier sequence provides a fine-grained, executable representation of an object's continuous vector shape. We train a VLM to act as a "visual decompiler", inferring the underlying generative program from its rendered, pixel-based output (Figure 1). This allows for substantial generalization: a model trained only on modern Chinese characters can reconstruct Oracle Bone Script in a zero-shot setting, suggesting an internalized "geometric grammar" — the compositional rules governing the arrangement, orientation, and form of the geometric primitives that constitute a character's structure.

Our contributions are as follows:

- We introduce and formalize the task of programmatic character reconstruction, where a model decompiles a raster image into an executable program of Bézier curves. To our knowledge, this is the first work to define character understanding as one of direct geometric program synthesis.
- We propose a framework that trains a VLM as a "visual decompiler", directly translating raster images into Bézier curve sequences with the spatial aid of an explicit visual coordinate system.
- We provide evidence that our model substantially outperforms strong zero-shot baselines such as GPT-4o. Crucially, it achieves zero-shot cross-script generalization from modern to ancient characters (Table 1), verifying its ability to learn an abstract and transferable geometric representation.

2 Methodology

In this section, we introduce how we construct paired "image-Bézier curve" training data and train VLM. We first introduce the data construction process of deriving Bézier curves from images of pictographic characters in Section 2.1, then introduce our training paradigm in Section 2.2.

2.1 Bézier Curves Extraction

To the best of our knowledge, few previous works have attempted to automatically construct mathematical representations, such as Bézier curves, from images of pictographic characters. Manual annotations, on the other hand, are costly and limited in scale, posing challenges for models to learn such representations and limiting their abilities to perform programmatic reconstruction. Thus, we develop an automated rule-based pipeline that converts character images to Bézier curve sequences. This approach allows us to create a large-scale "image-Bézier curve" dataset from any font file efficiently without requiring manual annotations. The pipeline operates in four steps:

1. **Preprocessing**: Input images are binarized to isolate the glyph.
2. **Skeletonization**: The glyph is skeletonized to a one-pixel-wide centerline, which preserves its topology.
3. **Graph Segmentation**: An 8-connected pixel graph [Noris et al., 2013] is built from the skeleton and segmented into paths at junctions and endpoints.
4. **Curve Fitting**: Each path is simplified using the Ramer-Douglas-Peucker algorithm [Douglas and Peucker, 1973], and adjacent segments are iteratively merged based on proximity and alignment to form smooth strokes.

This process yields a clean list of Bézier curves that faithfully represents the character's strokes. Further details are provided in Appendix B.

2.2 Training Paradigm

A primary challenge for a VLM to generate geometric programs is bridging the its textual output with the visual and spatial nature of target shapes. To address this, we first serialize the geometric information. We represent each glyph as a sequence of strokes encapsulated within "<bezierseq>" tags. Each stroke is defined by its control points inside "<bezier>" tags, e.g., `<bezier>(x1 y1)...</bezier>`.

Table 1: Main results comparing our trained model against zero-shot baselines. We report on three sub-metrics (D: Distance, A: Angle, L: Length) and the final geometric reconstruction score (G). Our model demonstrates superior performance and remarkable cross-script generalization.

Model / Method	Chinese STD (STD-zh)				Chinese Stylistic (Stylistic-zh)				OBS			
	D↑	A↑	L↑	G↑	D↑	A↑	L↑	G↑	D↑	A↑	L↑	G↑
Qwen2.5-VL-7B (Zero-Shot)	0.019	0.012	0.029	0.157	0.023	0.019	0.037	0.177	0.033	0.024	0.049	0.182
GPT-4o (Zero-Shot)	0.139	0.119	0.212	0.431	0.201	0.149	0.288	0.467	0.163	0.105	0.232	0.409
Ours (Qwen2.5-VL-7B + Warmup + RL)	0.508	0.454	0.654	0.678	0.561	0.463	0.687	0.663	0.359	0.292	**0.478**	0.548
Ours (Qwen2.5-VL-7B + SFT)	**0.770**	**0.686**	**0.854**	**0.821**	**0.603**	**0.509**	**0.696**	**0.723**	**0.362**	**0.293**	0.463	**0.568**

All coordinate values are normalized to a $[0, 1]$ range. This serialization transforms the visual reconstruction task into a program synthesis problem, compelling the model to learn a structured, symbolic representation of geometry rather than relying solely on pixel-level correlations.

However, serialization alone proves insufficient, as our analysis confirms that VLMs struggle to ground abstract coordinates to absolute positions within an image (see Table 2). To overcome this critical spatial reasoning deficit, we augment the input image by overlaying a normalized Cartesian coordinate system on the glyph. This visual aid, consisting of labeled x- and y-axes, provides explicit spatial context. This visual framework provides an explicit spatial anchor, enabling the model to directly map visual features to their precise numerical locations.

3 Experiments and Results

3.1 Experimental Settings

Models and Frameworks We use the Qwen2.5-VL-7B model [Bai et al., 2025] as our backbone, leveraging its strong vision-language capabilities. All experiments are conducted using the MS-Swift [Zhao et al., 2024] training framework. We compare our method against strong zero-shot baselines such as GPT-4o [Team, 2024] and the base Qwen2.5-VL-7B. More details on our baseline configurations are provided in Appendix C.

Training Data Our primary training dataset consists of 2,000 common Chinese characters. We use six font variations to create a dataset of approximately 12,000 samples for our main Supervised Fine-tuning (SFT) experiments. [1]

Evaluation Data We evaluate our models on three challenging, unseen datasets:

- **Chinese STD:** 1,000 unseen Chinese characters in a standard font (Source Han Sans SC Normal).
- **Chinese Stylistic:** 1,000 unseen Chinese characters in a stylistic, calligraphy-like font (JinNianYeYaoJiaYouYa) [2], which resembles handwritten style and helps assess the model's ability to capture expressive and diverse stroke patterns.
- **OBS:** 1,000 unseen Oracle Bone Script characters from the HWOBC [Li et al., 2020] dataset. This serves as our primary test set for cross-script generalization.

Experimental Details Our primary training method is Supervised Fine-Tuning (SFT) [Ouyang et al., 2022], where the main objective is to teach the VLM to function as a "visual decompiler" that translates character images into their corresponding Bézier curve programs. The model is trained on the axis-enhanced images and their ground-truth programs using a standard next-token prediction objective. For comparison, we also experiment with Reinforcement Learning (RL). This involves initializing a model from an SFT-warmed-up checkpoint and continuing training with a geometric reward signal using the GRPO algorithm [Shao et al., 2024]. This setup enables a direct comparison between purely supervised learning and reward-driven exploration. Further training details are available in Appendix C.

Evaluation Metrics We evaluate reconstruction quality using a comprehensive *Geometric Score* (G) as our primary metric, which also serves as the reward signal in our RL experiments. To provide

[1] Six fonts variations (Microsoft YaHei, SimHei, SimKai, SimLi, SimYou, SimSun), are standard system fonts included with Microsoft Windows.

[2] The modern Chinese fonts used for evaluation, Source Han Sans SC and JinNianYeYaoJiaYouYa, were sourced from the public font website `https://www.fonts.net.cn/`.

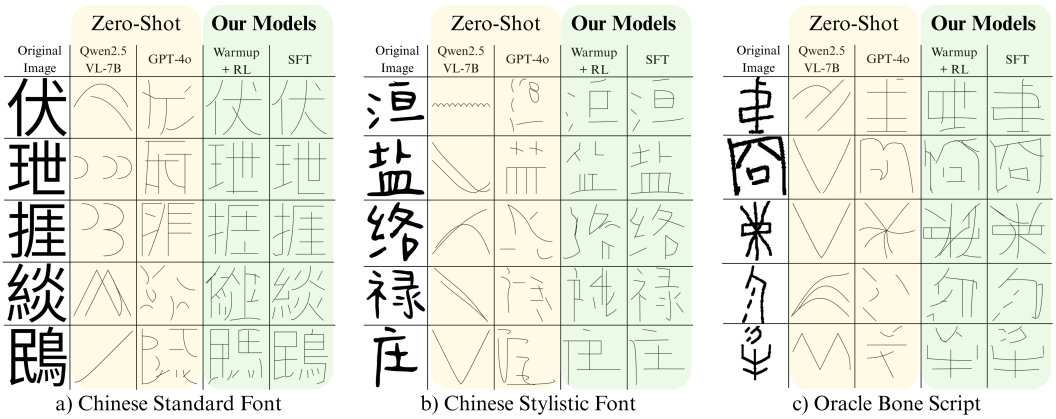

Figure 2: **Qualitative comparison of reconstructions across models. a, b, c** show reconstruction cases of Chinese standard font, Chinese stylistic font, and Oracle Bone Script, respectively.

a more granular analysis, we additionally report three sub-scores derived from the same matching framework: *Distance*, *Angle*, and *Length*. The primary *Geometric Score* calculation holistically compares the generated and ground-truth Bézier curve sequences by finding an optimal one-to-one stroke matching based on a composite similarity function. All final scores are normalized into the range $[0, 1]$, where 1 signifies a perfect match. A detailed breakdown is provided in Appendix D.

3.2 Main Results

Our findings, detailed in Table 1, confirm the effectiveness of our method across the overall reconstruction score and all three sub-metrics. Our model achieves the best results across all investigated datasets, demonstrating robust generalization to unseen standard and stylistic fonts.

Most strikingly, the model exhibits remarkable cross-script generalization. On the zero-shot Oracle Bone Script (OBS) evaluation, our SFT model scores 0.568, a **15.9%** increment over the powerful GPT-4o baseline. This result strongly supports our hypothesis that by learning programmatic construction, the model acquires a transferable geometric grammar rather than memorizing pixel patterns. To validate our Geometric Score's effectiveness, we conducted a human evaluation comparing our SFT model against GPT-4o on 150 samples in Appendix E. The results, showing a 91.6% average win rate for our model, confirm that our quantitative metric aligns well with human qualitative judgment.

4 Analysis

4.1 Case Study: Visualizing Reconstruction

In Figure 2, we present a qualitative comparison. We observe that our model successfully captures the core strokes and structure of the unseen characters, including Oracle Bone Script examples. In contrast, baseline models often produce visually plausible but structurally incorrect shapes, highlighting the limitations of purely pixel-based understanding.

4.2 Ablation Studies

To dissect our framework, we perform two key ablation studies in this section.

Table 2: Ablation results on the coordinate axis. The metric is the G↑.

Model / Method	STD-zh	Stylistic-zh	OBS
Zero-shot Baselines			
Qwen2.5-VL-7B (w/o Axis)	0.106	0.146	0.184
Qwen2.5-VL-7B (w/ Axis)	**0.157**	**0.177**	**0.182**
GPT-4o (w/o Axis)	0.347	0.361	0.381
GPT-4o (w/ Axis)	**0.431**	**0.467**	**0.409**
Our Fine-tuned Model			
Ours (SFT w/o Axis)	0.808	0.711	0.556
Ours (SFT w/ Axis)	**0.821**	**0.723**	**0.568**

The Critical Role of the Coordinate Axis. An ablation study (Table 2) shows that the coordinate system is a crucial component. Removing the axes degrades performance not only for our fine-tuned model but also for the zero-shot baselines (Qwen2.5-VL and GPT-4o). This consistently positive effect suggests that the axes serve as a fundamental anchor for spacial grounding, enhancing the innate geometric understanding capabilities of VLMs even without task-specific training.

A Deeper Look into Reinforcement Learning. We hypothesized that Reinforcement Learning (RL) would refine the SFT model's precision, but our results (Table 3) are more nuanced. While RL with an SFT warm-up improves over RL-only, it fails to surpass the performance of full SFT. This finding suggests a fundamental challenge in applying conventional RL algorithms to tasks involving fine-grained, continuous parameter generation. The action space for Bézier curve synthesis is vast, and the corresponding reward landscape is non-convex and sparse, where perturbations in a control point can cause discontinuous changes in the geometric score. This makes the credit assignment problem difficult for the exploration. In contrast, supervised learning provides a dense and globally coherent signal from ground-truth programs, which appears to be more effective in learning the complex structural dependencies required for this task. This highlights a key challenge for applying RL to structured visual generation and suggests that future investigations could benefit from exploring offline RL methodologies or developing more reward-shaping techniques tailored to geometric fidelity.

Table 3: Ablation results comparing different methods training out model. Qwen2.5-VL-7B is employed for this study. The metric is the G↑.

Model / Method	STD-zh	Stylistic-zh	OBS
Ours (RL)	0.539	0.497	0.432
Ours (Warmup + RL)	0.678	0.663	0.548
Ours (SFT)	**0.821**	**0.723**	**0.568**

5 Conclusion

This paper demonstrates a Vision-Language Model's capacity to perform "visual decompilation" by translating raster images into a precise mathematical representation as executable geometric programs. The model's ability to reconstruct Oracle Bone Script, despite being trained solely on modern Chinese characters, provides evidence that it acquires an abstract and transferable geometric grammar rather than memorizing pixel configurations. This study validates a mathematically grounded programmatic approach to visual understanding, highlighting two key findings: the critical role of an explicit coordinate system for spatial grounding and the efficacy of supervised learning to generate structured outputs. Ultimately, this work contributes to the development of advanced AI systems with a structural and generalizable capability for visual interpretation.

While our approach demonstrates strong generalization across scripts and styles, several challenges remain for future work. First, a deeper mathematical analysis of the Bézier representation and its parameter sensitivity is needed to better understand the underlying geometry encoding. Second, current geometric reasoning still lacks rigorous theoretical justification, which could further strengthen the interpretability of the model. Finally, the inference latency and tokenization bottleneck introduced by long Bézier sequences highlight an efficiency limitation that calls for more compact and adaptive vector representations.

Acknowledgement

This work is supported by the National Natural Science Foundation of China (No. 62276152, 62236011). We would like to thank Yaluo Liu for her assistance with the final revision of this paper.

References

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025. URL https://arxiv.org/abs/2502.13923.

Celeste Barnaby, Qiaochu Chen, Roopsha Samanta, and Işıl Dillig. Imageeye: Batch image processing using program synthesis. *Proc. ACM Program. Lang.*, 7(PLDI), June 2023. doi: 10.1145/3591248. URL https://doi.org/10.1145/3591248.

G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

Haichou Chen, Yishu Deng, Bin Li, Zeqin Li, Haohua Chen, Bingzhong Jing, and Chaofeng Li. Bézierseg: Parametric shape representation for fast object segmentation in medical images. *Life*, 13(3), 2023. ISSN 2075-1729. doi: 10.3390/life13030743. URL https://www.mdpi.com/2075-1729/13/3/743.

Song Chen, Xinyu Guo, Yadong Li, Tao Zhang, Mingan Lin, Dongdong Kuang, Youwei Zhang, Lingfeng Ming, Fengyu Zhang, Yuran Wang, Jianhua Xu, Zenan Zhou, and Weipeng Chen. Ocean-ocr: Towards general ocr application via a vision-language model, 2025. URL https://arxiv.org/abs/2501.15558.

Ayan Das, Yongxin Yang, Timothy Hospedales, Tao Xiang, and Yi-Zhe Song. Béziersketch: A generative model for scalable vector sketches. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVI*, page 632–647, Berlin, Heidelberg, 2020. Springer-Verlag. ISBN 978-3-030-58573-0. doi: 10.1007/978-3-030-58574-7_38. URL https://doi.org/10.1007/978-3-030-58574-7_38.

Xiaolei Diao, Rite Bo, Yanling Xiao, Lida Shi, Zhihan Zhou, Hao Xu, Chuntao Li, Xiongfeng Tang, Massimo Poesio, Cédric M. John, and Daqian Shi. Ancient script image recognition and processing: A review, 2025. URL https://arxiv.org/abs/2506.19208.

David H Douglas and Thomas K Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: the international journal for geographic information and geovisualization*, 10(2):112–122, 1973.

Kevin Ellis, Daniel Ritchie, Armando Solar-Lezama, and Joshua B. Tenenbaum. Learning to infer graphics programs from hand-drawn images. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 6062–6071, Red Hook, NY, USA, 2018. Curran Associates Inc.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

Aric Hagberg, Pieter Swart, and Daniel S Chult. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

Shreyas Kapur, Erik Jenner, and Stuart Russell. Diffusion on syntax trees for program synthesis. *arXiv preprint arXiv:2405.20519*, 2024.

Zaid Khan, Vijay Kumar BG, Samuel Schulter, Yun Fu, and Manmohan Chandraker. Self-training large language models for improved visual program synthesis with visual reinforcement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14344–14353, 2024.

H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955. doi: https://doi.org/10.1002/nav.3800020109. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.3800020109.

Bianca Lamm and Janis Keuper. Can visual language models replace ocr-based visual question answering pipelines in production? a case study in retail, 2024. URL https://arxiv.org/abs/2408.15626.

Bang Li, Qianwen Dai, Feng Gao, Weiye Zhu, Qiang Li, and Yongge Liu. Hwobc-a handwriting oracle bone character recognition database. *Journal of Physics: Conference Series*, 1651(1): 012050, nov 2020. doi: 10.1088/1742-6596/1651/1/012050. URL https://dx.doi.org/10.1088/1742-6596/1651/1/012050.

Caoshuo Li, Zengmao Ding, Xiaobin Hu, Bang Li, Donghao Luo, AndyPian Wu, Chaoyang Wang, Chengjie Wang, Taisong Jin, SevenShu, Yunsheng Wu, Yongge Liu, and Rongrong Ji. Oraclefusion: Assisting the decipherment of oracle bone script with structurally constrained semantic typography, 2025. URL https://arxiv.org/abs/2506.21101.

Yuliang Liu, Hao Chen, Chunhua Shen, Tong He, Lianwen Jin, and Liangwei Wang. Abcnet: Real-time scene text spotting with adaptive bezier-curve network. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Mor Shpigel Nacson, Aviad Aberdam, Roy Ganz, Elad Ben Avraham, Alona Golts, Yair Kittenplon, Shai Mazor, and Ron Litman. Docvlm: Make your vlm an efficient reader. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pages 29005–29015, June 2025.

Gioacchino Noris, Alexander Hornung, Robert W. Sumner, Maryann Simmons, and Markus Gross. Topology-driven vectorization of clean line drawings. *ACM Trans. Graph.*, 32(1), February 2013. ISSN 0730-0301. doi: 10.1145/2421636.2421640. URL https://doi.org/10.1145/2421636.2421640.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf.

Runqi Qiao, Qiuna Tan, Guanting Dong, MinhuiWu MinhuiWu, Jiapeng Wang, YiFan Zhang, Zhuoma GongQue, Chong Sun, Yida Xu, Yadong Xue, Ye Tian, Zhimin Bao, Lan Yang, Chen Li, and Honggang Zhang. V-oracle: Making progressive reasoning in deciphering oracle bones for you and me. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 20124–20150, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.986. URL https://aclanthology.org/2025.acl-long.986/.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Zecheng Tang, Chenfei Wu, Zekai Zhang, Minheng Ni, Shengming Yin, Yu Liu, Zhengyuan Yang, Lijuan Wang, Zicheng Liu, Juntao Li, and Nan Duan. Strokenuwa: tokenizing strokes for vector graphic synthesis. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

OpenAI Team. Gpt-4o system card, 2024. URL https://arxiv.org/abs/2410.21276.

Vikas Thamizharasan, Difan Liu, Shantanu Agarwal, Matthew Fisher, Michaël Gharbi, Oliver Wang, Alec Jacobson, and Evangelos Kalogerakis. Vecfusion: Vector font generation with diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7943–7952, 2024.

Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. Winoground: Probing vision and language models for visio-linguistic compositionality, 2022. URL https://arxiv.org/abs/2204.03162.

Stefan Van der Walt, Johannes L Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D Warner, Neil Yager, Emmanuelle Gouillart, and Tony Yu. scikit-image: image processing in python. *PeerJ*, 2:e453, 2014.

Yizhi Wang and Zhouhui Lian. Deepvecfont: synthesizing high-quality vector fonts via dual-modality learning. *ACM Transactions on Graphics (TOG)*, 40(6):1–15, 2021.

Ronghuan Wu, Wanchao Su, and Jing Liao. Chat2svg: Vector graphics generation with large language models and image diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pages 23690–23700, June 2025a.

Xiaofeng Wu, Karl Stratos, and Wei Xu. The impact of visual information in Chinese characters: Evaluating large models' ability to recognize and utilize radicals. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 331–350, Albuquerque, New Mexico, April 2025b. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.16. URL https://aclanthology.org/2025.naacl-long.16/.

Chongsheng Zhang, Ruixing Zong, Shuang Cao, Yi Men, and Bofeng Mo. Ai-powered oracle bone inscriptions recognition and fragments rejoining. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 5309–5311. International Joint Conferences on Artificial Intelligence Organization, 7 2020. doi: 10.24963/ijcai.2020/779. URL https://doi.org/10.24963/ijcai.2020/779. Demos.

Yuze Zhao, Jintao Huang, Jinghan Hu, Xingjun Wang, Yunlin Mao, Daoze Zhang, Zeyinzi Jiang, Zhikai Wu, Baole Ai, Ang Wang, Wenmeng Zhou, and Yingda Chen. Swift:a scalable lightweight infrastructure for fine-tuning, 2024. URL https://arxiv.org/abs/2408.05517.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/91f18a1287b398d378ef22505bf41832-Paper-Datasets_and_Benchmarks.pdf.

A Bézier Curve Fundamentals

A Bézier curve is a parametric curve fundamental to computer graphics, using Bernstein Polynomials as its basis. The curve, denoted as $c(t)$, is mathematically defined as a weighted sum of $n+1$ control points $\mathbf{b}_i$:

$$c(t) = \sum_{i=0}^{n} \mathbf{b}_i B_{i,n}(t), \quad 0 \leq t \leq 1, \tag{1}$$

where n is the degree of the curve and $B_{i,n}(t)$ are the Bernstein basis polynomials. Unlike raster images composed of a static pixel grid, a Bézier curve provides a continuous, mathematical definition of a shape, determined entirely by its set of control points.

In our work, we leverage Bézier curves to model the individual strokes of a character. This approach is advantageous because it provides a compact, flexible, and scalable vector representation. These properties have made Bézier curves a powerful tool in various deep learning applications. For instance, BézierSketch utilize sequences of curves to produce scalable vector sketches, shifting the modeling paradigm from dense pixel grids to sparse control points [Das et al., 2020]. More commonly, Bézier curves are used to parameterize and fit the external contours of objects. This is seen in domains such as real-time text spotting, where they define the boundaries of arbitrarily shaped text [Liu et al., 2020], and medical image analysis, where they ensure smooth and continuous segmentation masks [Chen et al., 2023]. In contrast to these approaches that focus on external boundaries, our work uses Bézier curves to reconstruct the internal topological structure of a character. By describing a character as a sequence of strokes, we transform the visual recognition problem into one of geometric program synthesis.

B Bézier Curve Extraction Details

Our pipeline uses OpenCV [Bradski, 2000] to load a grayscale raster and binarize it so strokes become foreground pixels. scikit-image's [Van der Walt et al., 2014] skeletonize function collapses thick strokes to one-pixel centerlines while preserving topology. NetworkX [Hagberg et al., 2008] builds an 8-connected pixel graph (nodes are skeleton pixels, edges are neighborhood adjacencies). svgwrite exports each simplified segment as SVG paths. svgpathtools parses the resulting SVG and exposes Line, QuadraticBezier, and CubicBezier primitives so the pipeline can collect control points, merge aligned segments, and normalize coordinates.

C Experimental Details

Baselines We compare our trained model against two powerful, zero-shot baselines: the base Qwen2.5-VL-7B model [Bai et al., 2025] and GPT-4o [Team, 2024]. For a robust and fair comparison, all baseline evaluations reported in our main results table are conducted using their stronger configuration, where the input image is enhanced with the same coordinate axis overlay provided to our models.

Implementation Details All models are trained on 8 NVIDIA A800 GPUs using DeepSpeed ZeRO Stage 3 and BF16 precision. For our main SFT model, we train for 5 epochs using a learning rate of 1×10^{-6} with a cosine scheduler and a warmup ratio of 0.01. We use a total batch size of 16. The vision transformer (ViT) backbone is kept frozen throughout training. For our comparative RL experiments, we initialize the model from a 5-epoch SFT warmup on $\frac{1}{6}$ of the data, followed by 1 epoch of training with the GRPO algorithm on the remaining $\frac{5}{6}$ of the data. The RL stage employs a distinct set of hyperparameters, including a global batch size of 48, a mini batch size of 24, rollout number per prompt of 6, and a generation temperature of 0.9, to explore the policy space.

D Detailed Evaluation Metrics

Our comprehensive scoring function, *Geometric Score*, is designed to quantitatively evaluate the quality of a reconstruction by reflecting its geometric fidelity. The calculation involves a three-step process: calculating pairwise similarity matrices between strokes, finding the optimal global assignment for each metric, and normalizing the results.

Pairwise Stroke Similarity Given a generated sequence of Bézier curves and a ground truth sequence, we compute similarity matrices where each element (i, j) represents a geometric similarity between the i-th ground truth stroke and the j-th generated stroke. For our analysis, we compute four distinct matrices: one for each sub-metric (Distance, Angle, Length), and a fourth composite matrix for the final *Geometric Score*. The core components are calculated as follows:

1. **Sampling:** We uniformly sample 10 points and their corresponding tangent vectors from both Bézier curves.
2. **Reward Calculation:** We compute three distinct geometric rewards, all normalized to $[0, 1]$, which serve as the values in the sub-metric similarity matrices:
   - **Distance Reward:** Calculated as $1/(1 + \bar{d})$, where $\bar{d}$ is the mean Euclidean distance between corresponding sampled points.
   - **Length Reward:** Calculated as $\min(L_1, L_2)/\max(L_1, L_2)$, rewarding similarity in the total arc length of the curves.
   - **Angle Reward:** The mean cosine similarity between corresponding tangent vectors, mapped from $[-1, 1]$ to $[0, 1]$. This measures directional alignment.
3. **Weighted Combination for Geometric Score:** For the composite similarity matrix used to calculate the final *Geometric Score*, the three rewards are combined using predefined weights: $0.6 \times$ Distance $+ 0.2 \times$ Length $+ 0.2 \times$ Angle. We also account for stroke directionality by calculating this composite score for both the original and reversed generated stroke, taking the maximum of the two.

Optimal Stroke Matching With the pairwise similarity matrices constructed, we treat the problem as a maximum weight bipartite matching task. This step is applied to each of the four matrices independently. We use the Hungarian algorithm [Kuhn, 1955] to find the optimal one-to-one assignment of generated strokes to ground truth strokes that maximizes the total similarity for that specific metric. This approach ensures that each reported score (Geometric Score, Distance, Angle, Length) is based on a globally optimal matching for that specific criterion. It is important to note that the final *Geometric Score* is derived from the matching on the composite matrix and is therefore not a simple weighted average of the three final sub-scores.

Final Score Normalization The sum of similarities from the optimal matching for each metric is normalized by the maximum number of strokes between the ground truth and the generated sequence. This yields a base score in the range $[0, 1]$. To enhance the signal for reinforcement learning, the score is then passed through a normalized sigmoid function centered at a threshold of 0.8, which amplifies differences in high-quality reconstructions.

E Human Evaluation Details

To validate that our proposed *Geometric Score* aligns with human perception of reconstruction quality, we conducted a formal human evaluation. This evaluation was designed to compare the qualitative performance of our best SFT model against the strong GPT-4o baseline.

Evaluation Setup We randomly sampled a total of 150 test cases, comprising 50 cases from each of our three distinct evaluation datasets: **Chinese STD**, **Chinese Stylistic**, and **OBS**. The evaluation was performed by five human experts, defined as individuals with proficiency in both Chinese calligraphy and digital vector graphics. The detailed breakdown of choices from each expert is shown in Table 4.

Table 4: Detailed breakdown of scores from the five human evaluators. "W" denotes cases where our model was preferred, "T" for ties, and "L" where the baseline was preferred.

Evaluator	W	T	L	Win Rate
Annotator 1	142	7	1	97.00%
Annotator 2	139	8	3	95.33%
Annotator 3	128	21	1	92.33%
Annotator 4	113	32	5	86.00%
Annotator 5	114	34	2	87.33%

Evaluation Protocol The evaluation followed a blind, side-by-side comparison protocol. For each of the 150 cases, the experts were presented with the ground-truth character image. Alongside it, the reconstructions from our SFT model and GPT-4o were displayed in a randomized order to prevent positional bias. The experts were not informed which model generated which image.

Following a standard methodology for evaluating generative models [Zheng et al., 2023], experts were asked to make a three-way choice for each pair of reconstructions:

- **Model A is Better**: If one reconstruction was clearly superior.
- **Model B is Better**: If the other reconstruction was clearly superior.
- **Tied**: If both reconstructions were of comparable quality (either equally good or equally bad).

The judgment criteria provided to the experts focused on (1) preservation of the character's topological structure, and (2) overall visual similarity to the source image.

Win Rate Calculation To aggregate the results, we calculated the overall win rate for our model against the baseline. The win rate is computed using a standard formula that gives partial credit for ties, providing a more nuanced performance measure than a simple win-loss metric. The formula is as follows:

$$\text{Win Rate} = \frac{\#(\text{Our Model Wins}) + 0.5 \times \#(\text{Ties})}{\text{Total Number of Comparisons}} \tag{2}$$

The final win rate, aggregated across all 150 samples and 6 experts, is reported in the main body of the paper.

F Related Work

Visual Program Synthesis. This field aims to translate visual inputs into executable programs, with diverse approaches tackling the challenge. Ellis et al. [2018] established the idea of factorizing the problem into a neural perception stage to identify drawing primitives and a symbolic synthesis stage to assemble them into a coherent program. More recent methods include reinforced self-training to overcome the need for expert-annotated datasets [Khan et al., 2024], while other novel paradigms explore neuro-symbolic languages [Barnaby et al., 2023] or diffusion models that operate directly on program syntax trees [Kapur et al., 2024]. In contrast, our work introduces a fine-grained "visual decompiler" task, training a model to translate images directly into a low-level geometric program of Bézier curves, emphasizing direct, structured generation over interpretative reasoning.

Generative Models for Vector Graphics. Generating vector graphics is a significant challenge, with key approaches including dual-modality learning [Wang and Lian, 2021] and cascaded diffusion models like VecFusion [Thamizharasan et al., 2024], which often rely on an intermediate raster stage to guide the final vector output. Other models like StrokeNUWA generate Scalable Vector Graphics (SVG) code directly, using "stroke tokens" that are inherently compatible with Large Language Models (LLMs) and mimic the sequential process of a human artist [Tang et al., 2024]. Hybrid frameworks such as Chat2SVG combine the semantic reasoning of LLMs to generate a basic template with the refinement power of image diffusion models to add geometric detail [Wu et al., 2025a]. Our framework differs by focusing on direct, end-to-end programmatic reconstruction from an image, bypassing creative generation and intermediate pixel-based stages to learn a canonical geometric grammar for a given character.

Vision-language Models for Text in Images. Historically, understanding text in images for tasks like Visual Question Answering (VQA) relied on multi-stage pipelines that sequentially performed detection, Optical Character Recognition (OCR), and reasoning. Modern Vision-language Models (VLMs) aim to replace this complex, error-prone process with a single end-to-end model [Lamm and Keuper, 2024]. However, this unified approach often struggles with fine-grained or dense text, where performance can be unreliable for tasks beyond simple transcription [Chen et al., 2025, Lamm and Keuper, 2024]. This has led to explorations into integrating specialized OCR encoders to improve detail preservation [Nacson et al., 2025]. A deeper level of understanding moves beyond mere transcription to parse the internal structure of characters; for complex logograms like Chinese, for instance, this involves recognizing constituent radicals and stroke arrangements [Wu et al., 2025b]. Our work advances this trajectory with a fundamentally different paradigm. Instead of asking the model to recognize or classify visual text, we frame the task as one of programmatic reconstruction, compelling the model to generate an executable geometric program. This shifts the goal from semantic identification to learning an underlying, generative grammar of form.

Historical Script Understanding and Generalization. Recognizing or understanding historical scripts is a difficult task due to data scarcity, degradation, and stylistic variation. This is particularly true for logographic systems like Oracle Bone Script (OBS), which have vast and complex character sets [Diao et al., 2025]. Traditional approaches often rely on script-specific, multi-stage pipelines for classification [Zhang et al., 2020] or general zero-shot recognizers like CLIP [Radford et al., 2021]. More recently, advanced frameworks have begun to leverage Vision-Language Models for this challenge. For instance, V-Oracle frames the task as a visual question-answering problem to interpret scripts through a multi-step reasoning chain [Qiao et al., 2025], while OracleFusion generates "semantically enriched vector fonts" for OBS, extending from analysis to creation [Li et al., 2025]. We build upon this direction but propose a novel paradigm of generalization through generation. By training on modern characters, our model learns a transferable, "universal geometric grammar." This allows it to demonstrate a deep, structural understanding by programmatically reconstructing an unseen script, rather than merely classifying or interpreting it, pushing the boundaries of cross-script generalization.