

Appendix

Due to space limitations in the main paper, we provide additional results and discussions in this appendix, organized as follows:

- Sec. A: More Details about VLLM-based Captioning.
- Sec. B: Detailed Comparison with Existing Work.
- Sec. C: Detailed Categories of Simple Parts and Complex Parts.
- Sec. D: Details about Metrics for Evaluating Text-to-CAD consistency.
- Sec. E: LLMs of Different Scales.
- Sec. F: Sensitivity Analysis of Key Hyper-parameters in Sampling.
- Sec. G: Criteria for Dataset Selection.
- Sec. H: Failure Cases, Limitations and Future Work.
- Sec. I: Five-shot Prompt Example.
- Sec. J: Additional Qualitative Results.

A More Details about VLLM-based Captioning

The prompt used for VLLM-based captioning is as follows:

"Given a loop in a CAD sketch, provide a brief description of its geometric shape starting with 'a' or 'an' if identifiable; otherwise, state 'None'."

Using this prompt, we randomly caption 1k complex local parts with GPT-4o [1] and Qwen2.5-VL-72B-Instruct [2], respectively. Regardless of whether these models output a specific shape or 'None', we manually evaluate each result by judging its correctness as either "Yes" or "No". The overall captioning accuracy across these 1,000 parts is 91.3% for Qwen2.5-VL-72B-Instruct and 86.5% for GPT-4o. These results indicate that Qwen2.5-VL-72B-Instruct outperforms GPT-4o in this captioning task, which is consistent with the latest multimodal model leaderboard rankings. Furthermore, given the lower cost of Qwen2.5-VL-72B-Instruct, we use it to caption the remaining complex parts.

B Detailed Comparison with Existing Work

As mentioned in lines 36-48 in our main paper, existing work struggles to achieve local geometry-controllable CAD generation. Here, we further highlight the differences between CAD-Editor [11], FlexCAD [12] and our GeoCAD. CAD-Editor has difficulty focusing on local generation for two main reasons: 1) It may unintentionally modify the remaining parts, resulting in outputs that do not align with user requirements (as illustrated in the last example of Fig. 1 in the original CAD-Editor paper). 2) CAD-Editor fails to accurately obtain angle and length information, making it incapable of generating even simple parts, such as a right triangle, let alone an isosceles right triangle, as mentioned in line 45 of our main paper. FlexCAD, on the other hand, can focus on local parts but incorporates minimal geometric constraints, thereby struggling to follow geometric instructions. In particular, FlexCAD is unable to understand, let alone follow, simple or complex geometric instructions. This limitation is clearly demonstrated in Fig. 1 of our main paper.

C Detailed Categories of Simple Parts and Complex Parts

The categories of simple parts include acute triangle, right triangle, obtuse triangle, isosceles triangle, isosceles right triangle (Notably, equilateral triangles do not occur in the DeepCAD [8] dataset), quadrilateral, trapezoid, isosceles trapezoid, kite (Two pairs of adjacent sides equal), parallelogram, rectangle, rhombus, square, circle, semicircle, quarter-circle, three-quarter circle, major-arc loop (defined as containing an arc longer than a semicircle), minor-arc loop (defined as containing an arc shorter than a semicircle), and so on. The remaining local parts are classified as complex, exhibiting more intricate and diverse visual patterns.

Table 1: Ablation studies on fine-tuning LLMs with different scales. Llama-3-8B is the model used in our main paper to enable a fair comparison with FlexCAD [12]. Transformer-4M is a small Transformer-based [6] language model, with a total number of trainable parameters comparable to that of our model in the main paper when using LoRA. Llama-3-8B-Full denotes full-parameter fine-tuning. Llama-3-8B, Qwen2.5-3B-Instruct, and Qwen2.5-7B-Instruct are all fine-tuned using LoRA. The best results are shown in **bold**, and the second-best results are marked with *.

Model	COV \uparrow	MMD \downarrow	JSD \downarrow	PV \uparrow	Ver-score \uparrow	VLLM-score \uparrow
Transformer-4M	59.1%	1.32	1.26	85.5%	69.3%	51.2%
Llama-3-8B-Full	67.5%*	1.02*	1.06	89.7%	78.9%*	64.2%
Llama-3-8B	64.9%	1.13	0.98*	90.5%	76.4%	65.7%*
Qwen2.5-3B-Instruct	65.8%	1.01	1.10	87.4%	74.2%	64.9%
Qwen2.5-7B-Instruct	68.7%	1.05	0.86	90.1%*	79.8%	70.2%

D Details about Metrics for Evaluating Text-to-CAD consistency

As mentioned in lines 213–217 of our main paper, we employ *Ver-score*, *VLLM-score*, and *Realism* to comprehensively evaluate model performance in terms of text-to-CAD consistency. Specifically, to compute *Ver-score*, we extract vertex coordinates from the generated local parts and analyze their geometric attributes to determine whether they align with the given geometric instructions. To obtain *VLLM-score*, we first render the local parts into images and then prompt two of the most powerful VLLMs, GPT-4o [1] and Qwen2.5-VL-72B-Instruct [2], to judge whether the rendered images match the corresponding instructions, assigning a binary label: "Yes" or "No." We report the average of their scores in Table 1 of our main paper, where both models significantly outperform the baselines. To evaluate *Realism*, we randomly render 500 newly generated CAD models into images, with the modified local parts clearly marked. Five crowd workers are then asked to assess whether the generated local parts align with the geometric instructions and do not conflict with the remaining parts. If both criteria are satisfied, they assign a binary label: "Yes"; otherwise, "No." The average score from these workers is reported in Table 1 of our main paper.

E LLMs of Different Scales

As shown in Table 1, Transformer-4M achieves the lowest performance, confirming that LLMs play a key role in enhancing local CAD generation. Llama-3-8B-Full performs comparably to Llama-3-8B, demonstrating the effectiveness of the LoRA strategy [3]. As two of the most popular open-source LLMs, Qwen2.5-7B-Instruct slightly outperforms Llama-3-8B.

F Sensitivity Analysis of Key Hyper-parameters in Sampling

Table 2: Effectiveness analysis of key hyper-parameters, including the sampling temperature τ and Top-p. Best performances are in **bold** and the second-bests are marked by *.

Model	COV \uparrow	MMD \downarrow	JSD \downarrow	PV \uparrow	Ver-score \uparrow	VLLM-score \uparrow
$\tau = 0.7$	63.4%	1.18	1.03	91.2%	75.9%	63.2%
$\tau = 0.9$	64.9%*	1.13	0.98*	90.5%*	76.4%*	65.7%
$\tau = 1.1$	65.6%	1.16*	0.95	89.1%	77.5%	65.1%*
Top-p = 0.8	64.1%	1.21	1.09	91.0%	75.3%	64.4%
Top-p = 0.9	64.9%*	1.13	0.98*	90.5%*	76.4%*	65.7%*
Top-p = 1.0	65.2%	1.18*	0.92	88.3%	76.9%	66.8%

As shown in Table 2, we conduct a sensitivity analysis on key hyperparameters, including the sampling temperature τ and Top-p. All other experimental settings follow those described in Section 4.2 of our main paper. In general, increasing τ or Top-p results in more diverse and stochastic predictions. However, this comes at the cost of reduced PV, while other metrics tend to improve, consistent with

findings in [12]. In our experiments, we balance this trade-off by selecting τ and Top-p values that ensure the PV remains above 90%.

G Criteria for Dataset Selection

DeepCAD [8] is a suitable dataset for evaluation, and the reasons are detailed below: 1) Scale: DeepCAD is a large-scale 3D CAD dataset, comprising over 178k samples. 2) Relevance to Controllability: Compared to 2D sketch datasets, DeepCAD better reflects the requirements of controllable generation, as aligning local parts within 3D CAD models is both more challenging and more practical. 3) Design Process Alignment: In contrast to other 3D CAD datasets, such as the ABC dataset [5], DeepCAD includes sketch-and-extrusion sequences that closely mirror the design workflows of commercial CAD tools like SolidWorks and AutoCAD. 4) Community Adoption: Due to its characteristics, DeepCAD is also the only choice for prior studies, including SkexGen [10], HNC-CAD [9], CAD-Editor [11], CADFusion [7], Text2CAD [4], and FlexCAD [12].

H Failure Cases, Limitations and Future Work

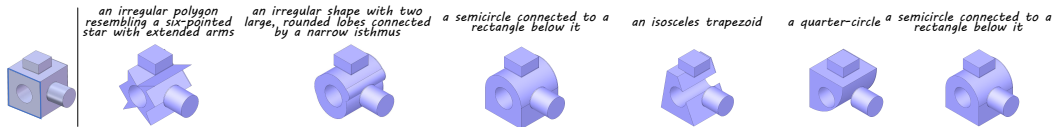


Figure A1: Failure cases. The generated local parts align well with the user’s geometric instructions but do not integrate smoothly with the remaining parts of the original CAD model.

Failure cases. Despite notable advancements, our GeoCAD sometimes results in failure cases. As shown in Fig. A1, given a CAD model, when only the special part is modified (*i.e.*, the part upon which the remaining parts are constructed and strictly aligned in size), the unchanged remaining parts may lead to structural conflicts with it. To mitigate this issue, when modifying the special parts, users should provide geometric instructions that account for the constraints imposed by the remaining parts, since the DeepCAD dataset does not annotate the relationships between different parts.

Limitations and future work. In this paper, we fine-tune LLMs to enable local geometry-controllable CAD generation, primarily guided by textual instructions. However, in practice, certain complex local parts may be difficult or even impossible to describe using text alone. Thus, in the future, if users can complement textual inputs with hand-drawn images for local geometry-controllable CAD generation, they may be able to convey their design intent more effectively. Given the strong capabilities of VLLMs in both CAD generation and text understanding, our future work aims to develop a more advanced multimodal LLM tailored for controllable CAD generation from both text and image inputs.

I Five-shot Prompt Example

To better illustrate the implementation details of the baselines and our GeoCAD in Table 1 of our main paper, we present a five-shot prompt example, as shown in Fig. A2.

J Additional Qualitative Results

We provide additional qualitative results in Fig. A3.

References

- [1] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

- 104 [2] J. Bai, S. Bai, S. Yang, S. Wang, S. Tan, P. Wang, J. Lin, C. Zhou, and J. Zhou. Qwen-vl: A
105 versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv*
106 *preprint arXiv:2308.12966*, 2023.
- 107 [3] E. J. Hu, yelong shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. LoRA:
108 Low-rank adaptation of large language models. In *International Conference on Learning*
109 *Representations*, 2022.
- 110 [4] M. S. Khan, S. Sinha, T. Uddin, D. Stricker, S. A. Ali, and M. Z. Afzal. Text2cad: Generat-
111 ing sequential cad designs from beginner-to-expert level text prompts. *Advances in Neural*
112 *Information Processing Systems*, 37:7552–7579, 2024.
- 113 [5] S. Koch, A. Matveev, Z. Jiang, F. Williams, A. Artemov, E. Burnaev, M. Alexa, D. Zorin, and
114 D. Panozzo. Abc: A big cad model dataset for geometric deep learning. In *Proceedings of the*
115 *IEEE/CVF conference on computer vision and pattern recognition*, pages 9601–9611, 2019.
- 116 [6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and
117 I. Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference*
118 *on Neural Information Processing Systems*, NIPS’17, page 6000–6010, Red Hook, NY, USA,
119 2017. Curran Associates Inc.
- 120 [7] S. Wang, C. Chen, X. Le, Q. Xu, L. Xu, Y. Zhang, and J. Yang. Cad-gpt: Synthesising cad
121 construction sequence with spatial reasoning-enhanced multimodal llms. In *Proceedings of the*
122 *AAAI Conference on Artificial Intelligence*, volume 39, pages 7880–7888, 2025.
- 123 [8] R. Wu, C. Xiao, and C. Zheng. Deepcad: A deep generative network for computer-aided design
124 models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages
125 6772–6782, 2021.
- 126 [9] X. Xu, P. K. Jayaraman, J. G. Lambourne, K. D. Willis, and Y. Furukawa. Hierarchical
127 neural coding for controllable CAD model generation. In A. Krause, E. Brunskill, K. Cho,
128 B. Engelhardt, S. Sabato, and J. Scarlett, editors, *ICML*, volume 202 of *Proceedings of Machine*
129 *Learning Research*, pages 38443–38461. PMLR, 23–29 Jul 2023.
- 130 [10] X. Xu, K. D. Willis, J. G. Lambourne, C.-Y. Cheng, P. K. Jayaraman, and Y. Furukawa. SkexGen:
131 Autoregressive generation of CAD construction sequences with disentangled codebooks. In
132 K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, editors, *ICML*, volume
133 162 of *Proceedings of Machine Learning Research*, pages 24698–24724, 17–23 Jul 2022.
- 134 [11] Y. Yuan, S. Sun, Q. Liu, and J. Bian. Cad-editor: A locate-then-infill framework with automated
135 training data synthesis for text-based cad editing. *ICML*, 2025.
- 136 [12] Z. Zhang, S. Sun, W. Wang, D. Cai, and J. Bian. FlexCAD: Unified and versatile controllable
137 CAD generation with fine-tuned large language models. In *The Thirteenth International*
138 *Conference on Learning Representations*, 2025.

You answer questions about controllable CAD generation. When answering user questions, please follow these examples:

Example 1

Instruction:

Below is a partial description of a CAD sequence where one command has been replaced with the string "[loop mask]":

line,0,26 <curve_end> line,1,26 <curve_end> line,1,28 <curve_end> line,0,28 <curve_end> <loop_end> <face_end> [loop mask] <face_end>
<sketch_end> add,31,45,31,31,1,0,0,0,1,0,0,0,1,44,31,36 <extrusion_end>

Generate an string that could replace "[loop mask]" in the CAD sequence. Notably, the string denotes an isosceles right triangle.

Answer:

line,0,28 <curve_end> line,1,29 <curve_end> line,1,28 <curve_end> <loop_end>

Example 2

Instruction:

Below is a partial description of a CAD sequence where one command has been replaced with the string "[loop mask]":

line,42,47 <curve_end> line,43,46 <curve_end> line,43,47 <curve_end> <loop_end> <face_end> line,43,15 <curve_end> line,44,15
<curve_end> line,44,16 <curve_end> <loop_end> <face_end> [loop mask] <face_end> <sketch_end> add,28,34,31,31,1,0,0,0,1,0,-
1,0,38,46,31 <extrusion_end>

Generate an string that could replace "[loop mask]" in the CAD sequence. Notably, the string denotes an isosceles right triangle.

Answer:

line,43,47 <curve_end> line,44,46 <curve_end> line,44,47 <curve_end> <loop_end>

Example 3

Instruction:

Below is a partial description of a CAD sequence where one command has been replaced with the string "[loop mask]":

line,14,57 <curve_end> line,16,55 <curve_end> line,16,10 <curve_end> line,21,4 <curve_end> line,47,4 <curve_end> line,48,8 <curve_end>
line,48,8 <curve_end> line,47,5 <curve_end> line,21,5 <curve_end> line,16,10 <curve_end> line,16,55 <curve_end> line,14,58 <curve_end>
<loop_end> <face_end> <sketch_end> add,31,63,31,31,1,0,0,0,1,0,-1,0,21,39,45 <extrusion_end> line,26,0 <curve_end> line,29,0
<curve_end> line,36,7 <curve_end> line,36,62 <curve_end> line,36,62 <curve_end> line,36,7 <curve_end> line,29,0 <curve_end> line,26,0
<curve_end> <loop_end> <face_end> line,26,0 <curve_end> line,29,0 <curve_end> line,36,7 <curve_end> line,36,62 <curve_end> line,26,62
<curve_end> <loop_end> <face_end> [loop mask] <face_end> <sketch_end> cut,27,31,31,55,-1,0,0,0,1,0,1,0,16,29,17 <extrusion_end>

Generate an string that could replace "[loop mask]" in the CAD sequence. Notably, the string denotes an isosceles right triangle.

Answer:

line,29,0 <curve_end> line,36,0 <curve_end> line,36,7 <curve_end> <loop_end>

Example 4

Instruction:

Below is a partial description of a CAD sequence where one command has been replaced with the string "[loop mask]":

[loop mask] <face_end> <sketch_end> add,31,62,31,31,1,0,0,0,1,0,-1,0,36,31,44 <extrusion_end> line,5,14 <curve_end> line,5,31
<curve_end> line,22,48 <curve_end> line,40,48 <curve_end> line,57,31 <curve_end> line,57,14 <curve_end> line,31,40 <curve_end>
<loop_end> <face_end> <sketch_end> add,31,39,31,31,1,0,0,0,1,0,-1,0,39,31,48 <extrusion_end>

Generate an string that could replace "[loop mask]" in the CAD sequence. Notably, the string denotes an isosceles right triangle.

Answer:

line,31,17 <curve_end> line,59,17 <curve_end> line,31,45 <curve_end> <loop_end>

Example 5

Instruction:

Below is a partial description of a CAD sequence where one command has been replaced with the string "[loop mask]":

line,6,12 <curve_end> line,6,50 <curve_end> line,43,50 <curve_end> line,56,36 <curve_end> line,56,33 <curve_end> line,32,33 <curve_end>
line,32,29 <curve_end> line,56,29 <curve_end> line,56,26 <curve_end> line,43,12 <curve_end> <loop_end> <face_end> line,43,12
<curve_end> line,56,12 <curve_end> line,56,26 <curve_end> <loop_end> <face_end> line,43,50 <curve_end> line,56,36 <curve_end>
line,56,50 <curve_end> <loop_end> <face_end> <sketch_end> add,31,39,31,31,1,0,0,0,1,0,0,1,44,24,34 <extrusion_end> [loop mask]
<face_end> line,20,61 <curve_end> line,42,39 <curve_end> line,42,61 <curve_end> <loop_end> <face_end> <sketch_end>
cut,31,57,31,31,1,0,0,0,1,0,0,0,1,27,45,34 <extrusion_end> line,22,1 <curve_end> line,40,1 <curve_end> line,40,61 <curve_end> line,22,61
<curve_end> <loop_end> <face_end> <sketch_end> add,31,41,30,33,39,1,0,0,0,1,0,0,0,1,27,3,31 <extrusion_end> line,0,25 <curve_end>
line,6,37 <curve_end> line,56,37 <curve_end> line,62,25 <curve_end> <loop_end> <face_end> <sketch_end>
cut,5,31,11,33,44,0,1,0,0,0,1,1,0,0,19,31,34 <extrusion_end>

Generate an string that could replace "[loop mask]" in the CAD sequence. Notably, the string denotes an isosceles right triangle.

Answer:

line,20,1 <curve_end> line,42,1 <curve_end> line,42,23 <curve_end> <loop_end>

Instruction:

Below is a partial description of a CAD sequence where one command has been replaced with the string "[loop mask]":

line,4,20 <curve_end> line,22,14 <curve_end> line,47,14 <curve_end> line,58,25 <curve_end> line,47,25 <curve_end> line,22,25 <curve_end>
line,4,25 <curve_end> <loop_end> <face_end> line,4,37 <curve_end> line,22,37 <curve_end> line,22,48 <curve_end> line,4,42 <curve_end>
<loop_end> <face_end> line,22,37 <curve_end> line,47,37 <curve_end> line,47,48 <curve_end> line,22,48 <curve_end> <loop_end>
<face_end> arc,47,25,51,27 <curve_end> line,53,31 <curve_end> line,58,31 <curve_end> line,58,25 <curve_end> <loop_end> <face_end>
arc,47,37,51,35 <curve_end> line,53,31 <curve_end> line,58,31 <curve_end> line,58,37 <curve_end> <loop_end> <face_end> [loop mask]
<face_end> <sketch_end> add,29,33,31,31,1,0,0,0,1,0,0,0,1,32,18,31 <extrusion_end>

Generate an string that could replace "[loop mask]" in the CAD sequence. Notably, the string denotes an isosceles right triangle.

Answer:

Figure A2: A five-shot prompt example used in Table 1 of our main paper.









	<i>an ellipse with rectangular extensions at both the top and bottom</i>		<i>a quarter circle</i>		<i>a rectangle missing one corner</i>		<i>a broken ring shape</i>		<i>a polygon that looks like an irregular cross shape</i>		<i>a shape that looks like a water droplet</i>	
	<i>a heart-shaped loop</i>		<i>a major-arc connected to a rectangle</i>		<i>a shape that looks like a semicircular ring</i>		<i>a rectangle with rounded corners and four notches</i>		<i>a rectangle with a semicircular notch at the top</i>		<i>a rectangle with a U-shaped notch at the top</i>	
	<i>a cross-shaped loop</i>		<i>a shape formed by two smoothly connected circles</i>		<i>a semicircle connected to a rectangle below it</i>		<i>a rounded rectangle with a circle at the top and bottom</i>		<i>a quarter circle connected to a rectangle</i>		<i>an irregular polygon with a pointed tip at the top</i>	
	<i>a rectangle with a semicircular notch at the bottom</i>		<i>a rectangle with a triangular notch at the bottom</i>		<i>an irregular polygon</i>		<i>a rectangle with four pointed corners and four notches</i>		<i>a rectangle connected to a semicircle</i>		<i>a smooth cross-shaped loop</i>	
	<i>a letter 'I'</i>		<i>a letter 'T'</i>		<i>a letter 'U'</i>		<i>a letter 'H'</i>		<i>a letter 'L'</i>		<i>a letter 'C'</i>	
	<i>a letter 'I'</i>		<i>a letter 'Y'</i>		<i>a letter 'V'</i>		<i>a letter 'T'</i>		<i>a letter 'X'</i>		<i>a letter 'L'</i>	
	<i>a letter 'L'</i>		<i>a letter 'U'</i>		<i>a letter 'V'</i>		<i>a letter 'X'</i>		<i>a letter 'H'</i>		<i>a letter 'T'</i>	
	<i>a square</i>		<i>a regular polygon with 8 sides</i>		<i>a regular polygon with 6 sides</i>		<i>a rectangle</i>		<i>an elongated rectangle</i>		<i>a rhombus</i>	
	<i>an isosceles trapezoid</i>		<i>a right trapezoid</i>		<i>an irregular quadrilateral</i>		<i>a square</i>		<i>a rhombus</i>		<i>a circle</i>	

Figure A3: Additional Qualitative Results.