ICLR

Plan then Act: Bi-level CAD Command Sequence Generation

Qiangya Guo*, Gang Dai*, Zhuoman Liu, Shuangping Huang,
Yunqing Hu, Huiyuan Zhang, Tianshui Chen

South China University of Technology, Guangdong University of Technology,
Hong Kong Polytechnic University, Zhuzhou CRRC Times Electric Co., Ltd

THE HONG KONG POLYTECHNIC UNIVERSITY
香港理工大學

中国中车
CRRC

Overview

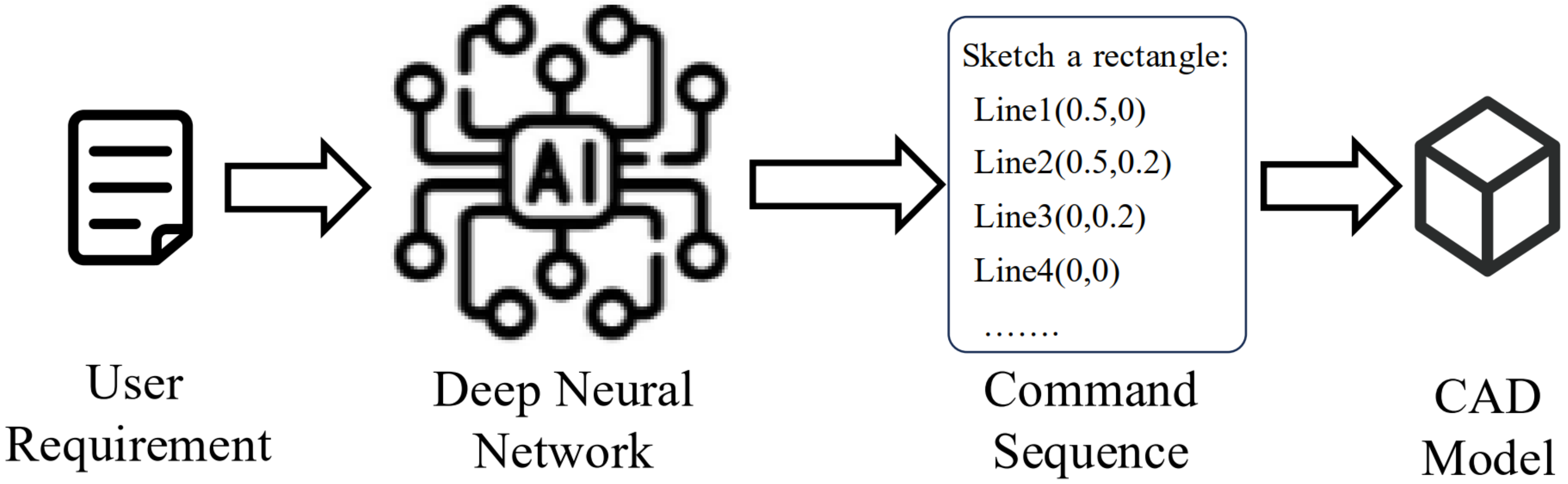
- Background
- Proposed Method
  - High-level plan generation
  - Low-level command generation
- Experimental Results
- Conclusion

Overview

- Background
- Proposed Method
  - High-level plan generation
  - Low-level command generation
- Experimental Results
- Conclusion

CAD Command Sequence Generation

- CAD command sequence generation aims to synthesize executable CAD operation sequences that satisfy user requirements



[ICLR 2026] Plan then Act: Bi-level CAD Command Sequence Generation

Challenges

Limitation of existing methods

- LLMs pre-trained on large-scale general data often struggle to directly synthesize task-specific CAD control sequences
- Small models trained from scratch exhibit limited semantic parsing and modeling capabilities, frequently failing to handle the complex CAD modeling tasks

Challenges

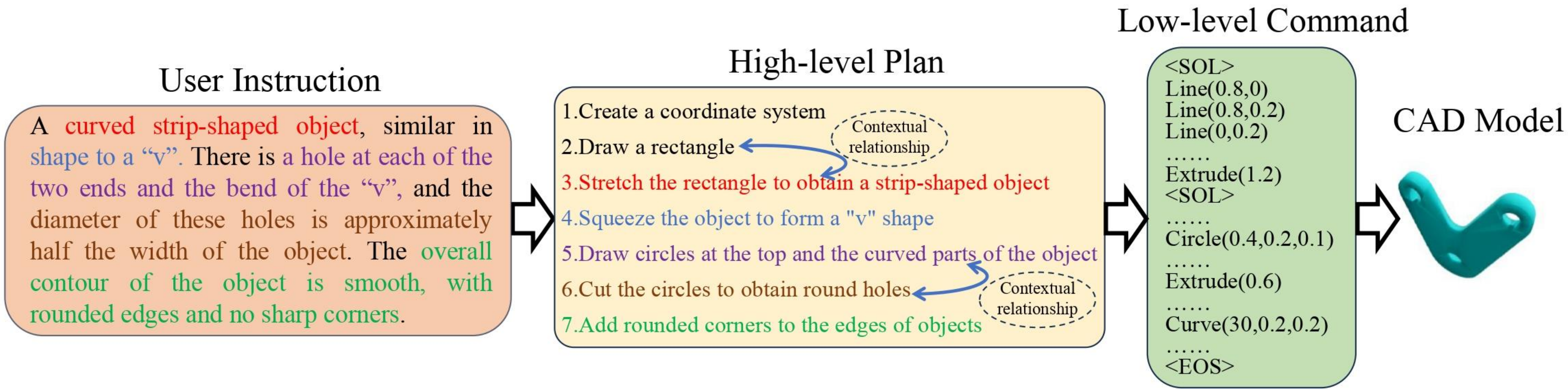
- How to accurately understand user semantics
- How to generate CAD control sequences with strict domain rules

Overview

- Background
- Proposed Method
  - High-level plan generation
  - Low-level command generation
- Experimental Results
- Conclusion

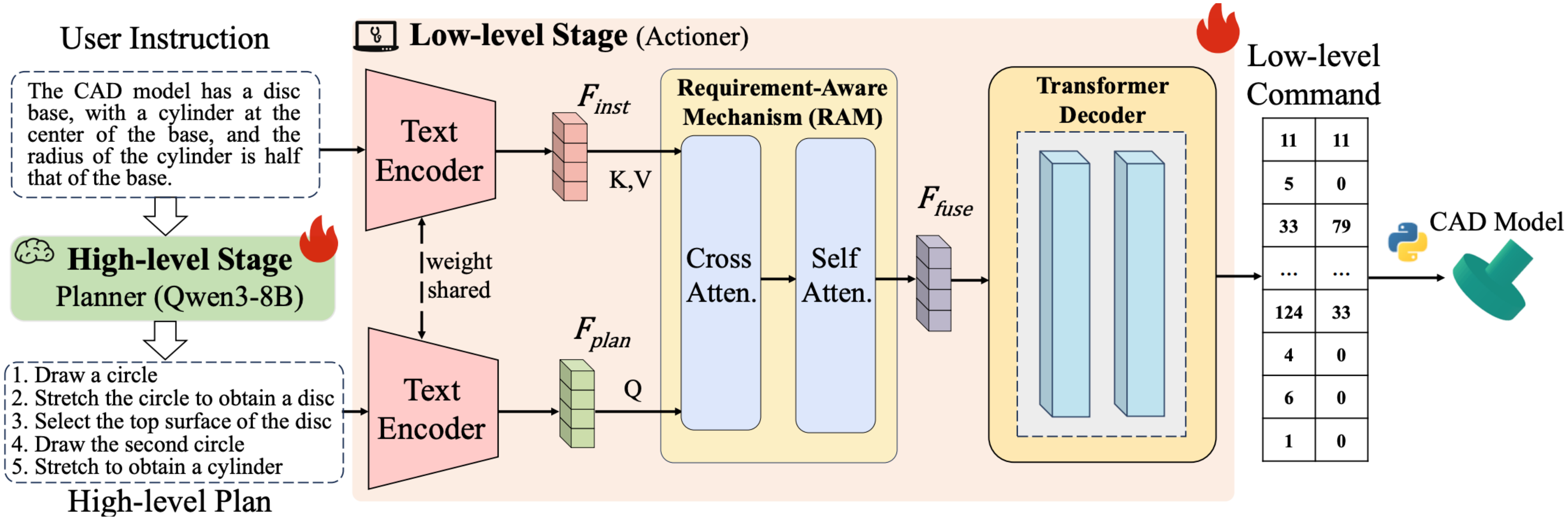
Motivation

- Inspired by human planning strategies, we propose an assumption that decomposing user instructions into a chain-like operation plan can improve the accuracy of command sequence generation
  - Utilize an LLM to parse user instructions and generate a chain-like operational plan
  - A small model is designed to convert the operational plan into task-specific CAD control sequences



[ICLR 2026] Plan then Act: Bi-level CAD Command Sequence Generation

Overall Pipeline



- During the high-level stage, an LLM-based Planner completes the planning process, parsing user instruction into **a high-level operation plan**
- At the low-level generation stage, an Actioner equipped with a requirement-aware mechanism **extracts design requirements (e.g., dimensions, geometric relationships)** and generate **low-level executable command sequence**

[ICLR 2026] Plan then Act: Bi-level CAD Command Sequence Generation

High-level Plan Generation

Prompt: you are a CAD task planner. Your role is to generate CAD operation steps as a high-level operation plan based on user design instruction. There is an example for reference.

example: { user instruction :“ A rectangular prism with a flat top and bottom. The length of the prism is approximately three times its width. The height of the prism is approximately two times its width. The prism is solid and has no holes or openings. ”,

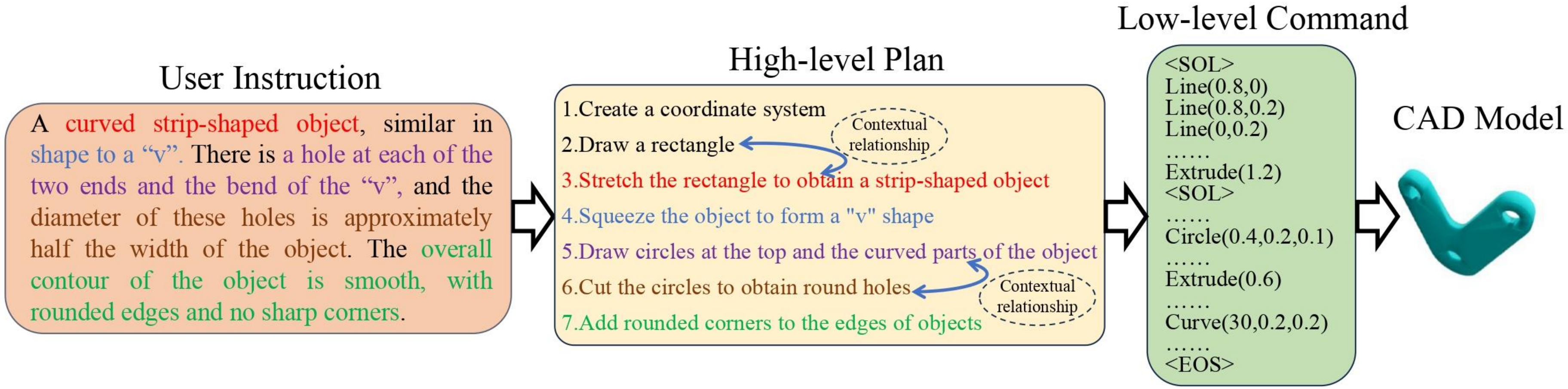
high-level plan:“

1. Create the coordinate system
2. Draw a rectangle
4. scale the rectangle
5. Transform the 2D sketch into 3D
6. Extrude the 2D sketch to get a rectangular prism ”}

Now, based on the given reference, parse user instructions into a high-level operation plan. Output only the high-level plan.

- During high-level plan generation, we use a prompt that includes an example to guide Qwen3-8B to assume the role of a CAD Planner, encouraging it to generate high-level operation plans
- Due to LLM hallucinations, we do not request the LLM to generate a detailed operational process with specific parameters. Instead, we use a simple yet complete workflow as the plan

Low-level Command Generation



- Requirement-Aware Mechanism（RAM）
  - First, we use cross-attention, employing the operation steps in the high-level plan as queries to retrieve and integrate the most relevant requirement information in the user instruction

$$O = Atten_1(Q_1 = F_{plan}, K_1 = V_1 = F_{inst})$$

  - Then, we use self-attention to capture the contextual relationships between the operational steps

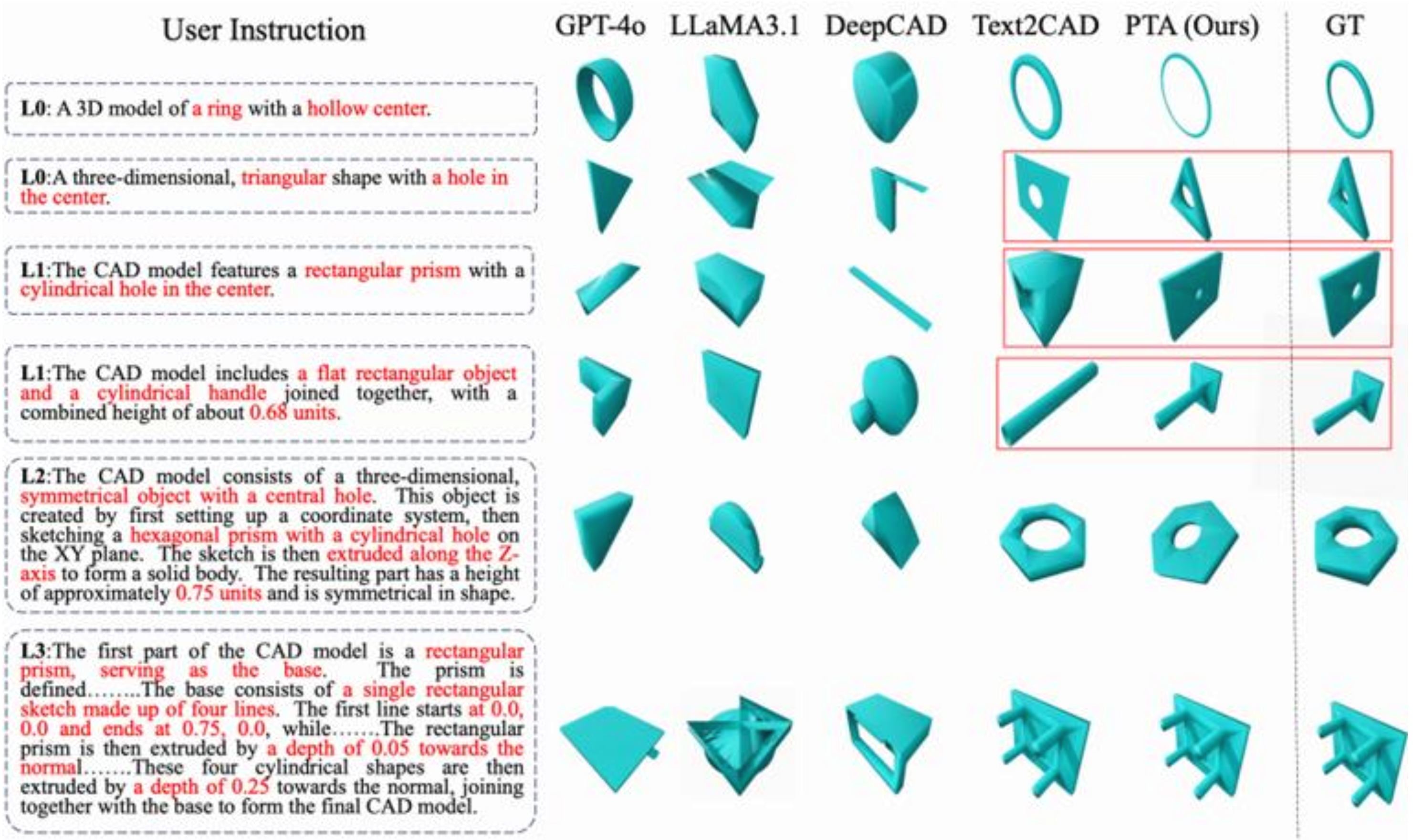
$$F_{fuse} = Atten_2(Q_2 = K_2 = V_2 = F_{plan} + O)$$

Overview

- Background
- Proposed Method
  - High-level plan generation
  - Low-level command generation
- Experimental Results
- Conclusion

CAD Command Sequence Generation

- Qualitative comparisons with LLMs, SOTA methods



- The visualization results demonstrate that our PTA has advantages in generating **both overall shapes and local details**

[ICLR 2026] Plan then Act: Bi-level CAD Command Sequence Generation

CAD Command Sequence Generation

- Quantitative comparisons with LLMs, SOTA methods

User Level	Method	GPT-4V↑	CD↓		JSD↓	MMD↓	COV↑	F1↑		IR↓
			Mean	Median				Sketch	Extrusion	
Abstract (L0)	GPT-4o	7.60	338.52	247.11	207.71	40.78	10.37	16.12	63.51	70.35
	LLaMA3.1	10.70	262.26	201.16	176.36	35.85	19.32	22.54	58.61	33.46
	DeepCAD	13.50	253.06	191.44	137.89	30.10	18.77	30.49	67.07	14.96
	Text2CAD	23.90	233.84	187.31	130.13	24.06	24.22	39.50	85.30	1.78
	PTA (Ours)	**44.30**	**183.14**	**113.79**	**26.35**	**19.69**	**40.63**	**50.61**	**90.33**	**0.63**
Beginner (L1)	GPT-4o	6.80	278.75	232.70	187.62	39.36	9.66	16.76	72.34	73.71
	LLaMA3.1	13.20	269.45	201.41	156.35	40.11	17.37	23.63	65.34	43.52
	DeepCAD	18.60	281.35	232.33	139.00	32.65	15.89	32.77	76.06	16.61
	Text2CAD	20.50	244.84	206.28	129.40	22.91	21.87	43.74	91.75	1.50
	PTA (Ours)	**40.90**	**200.25**	**142.50**	**25.52**	**18.66**	**37.50**	**52.07**	**94.53**	**0.68**
Intermedia. (L2)	GPT-4o	8.00	260.63	200.36	103.66	32.44	20.79	22.10	73.23	81.47
	LLaMA3.1	18.40	214.70	115.31	87.99	30.51	25.66	29.38	72.29	37.99
	DeepCAD	15.30	210.79	111.06	50.34	25.36	27.21	40.76	62.51	15.63
	Text2CAD	24.60	150.04	74.16	30.65	17.33	40.65	48.55	93.40	1.77
	PTA (Ours)	**33.70**	**140.53**	**67.73**	**11.91**	**16.95**	**43.75**	**52.55**	**94.20**	**1.37**
Expert (L3)	GPT-4o	11.20	83.52	44.83	63.99	13.42	40.83	32.07	73.86	66.91
	LLaMA3.1	14.90	75.91	20.24	30.03	8.57	44.36	43.69	67.04	26.57
	DeepCAD	17.60	69.34	8.24	20.45	7.66	53.21	62.67	54.86	12.81
	Text2CAD	25.80	26.41	0.37	3.15	5.18	73.34	63.80	93.31	0.93
	PTA (Ours)	**30.50**	**20.08**	**0.31**	**2.85**	**4.76**	**78.12**	**64.95**	**94.59**	**0.71**

- Our PTA outperforms closed-source LLM, open-source LLM, and previous CAD command sequence generation methods across all quantitative metrics

CAD Command Sequence Generation

- Ablation studies of bi-level generation and the Requirement-Aware Mechanism

User Level	Method	CD↓ Mean	CD↓ Median	Avg. F1↑	IR↓
Abstract (L0)	Planner only	248.93	218.97	41.87	20.06
	Actioner only	247.21	200.32	45.16	2.66
	PTA (Ours)	**183.14**	**113.79**	**60.54**	**0.63**
Beginner (L1)	Planner only	247.33	206.71	46.66	25.37
	Actioner only	266.57	189.59	53.34	4.71
	PTA (Ours)	**200.25**	**142.50**	**62.68**	**0.68**
Intermedia. (L2)	Planner only	199.36	93.38	53.73	18.90
	Actioner only	180.72	79.74	55.48	2.17
	PTA (Ours)	**140.53**	**67.73**	**62.96**	**1.37**
Expert (L3)	Planner only	45.78	15.33	52.91	15.66
	Actioner only	37.13	0.45	67.89	1.14
	PTA (Ours)	**20.08**	**0.31**	**72.36**	**0.71**

Table 2: Comparison between single-stage generation and our bi-level generation (PTA)

User Level	Guidance in Actioner	CD↓ Mean	CD↓ Median	Avg. F1↑	IR↓
Abstract (L0)	F_{plan} only	280.07	259.31	28.82	3.05
	Concat	198.36	159.98	55.91	2.32
	Add	208.06	163.27	55.17	2.92
	RAM	**183.14**	**113.79**	**60.54**	**0.63**
Beginner (L1)	F_{plan} only	299.62	238.04	40.63	3.57
	Concat	228.72	171.87	56.99	1.78
	Add	227.51	180.91	57.11	1.24
	RAM	**200.25**	**142.50**	**62.68**	**0.68**
Intermediate (L2)	F_{plan} only	259.79	216.74	47.43	3.66
	Concat	147.66	70.16	60.14	1.97
	Add	151.02	69.77	59.90	0.93
	RAM	**140.53**	**67.73**	**62.96**	**1.37**
Expert (L3)	F_{plan} only	200.33	179.31	59.82	2.31
	Concat	24.38	0.36	71.44	0.93
	Add	24.84	0.35	71.62	1.01
	RAM	**20.08**	**0.31**	**72.36**	**0.71**

Table 3: Ablation Study of RAM.

- Bi-level generation outperforms single-stage generation
- Using RAM for information fusion is superior to feature concatenation or addition

[ICLR 2026] Plan then Act: Bi-level CAD Command Sequence Generation

Overview

- Background
- Proposed Method
  - High-level plan generation
  - Low-level command generation
- Experimental Results
- Conclusion

Conclusion

Contributions

- We propose PTA, a new **bi-level CAD modeling method** that first decomposes user instructions into high-level operation plans via LLM, then completes the low-level command sequence generation
- We design a requirement-aware mechanism that automatically **extracts critical requirement information** from the user instruction, improving the alignment of the generated CAD sequences with user requirements

Future Work

- Explore the potential of PTA in multimodality-conditioned CAD generation