FlexCAD: Unified and Versatile Controllable CAD Generation with Fine-tuned Large Language Models

Zhanwei Zhang[1] Shizhao Sun[2] Wenxiao Wang[1] Deng Cai[1] Jiang Bian[2]
1 State Key Lab of CAD&CG, Zhejiang University, 2 Microsoft Research

Motivation

Figure 1: Controllable CAD generation achieved by FlexCAD. In each sub-figure, the left side shows the input: an original CAD model along with the part the user intends to modify (highlighted in blue). The right side displays the output: multiple new CAD models with only the chosen part changed. Users have the flexibility to specify the part in any CAD construction hierarchies, ranging from coarse levels like sketch-extrusion to fine levels like curve (as illustrated from (a) to (f)).

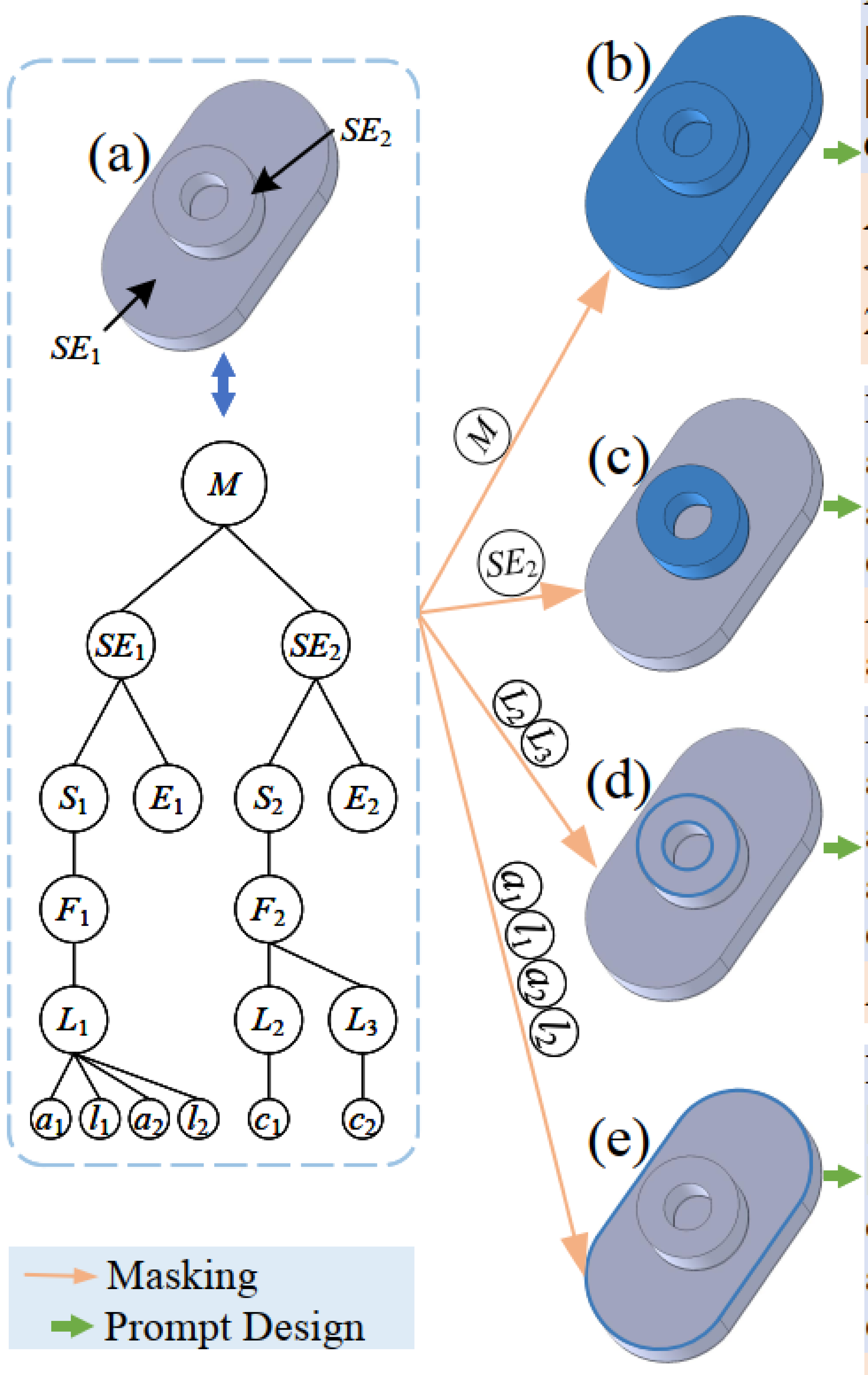
Method

Figure 2: The overall framework of FlexCAD. (a) Training process. Initially, a CAD model is converted into a structured text. Next, a hierarchy-aware masking strategy is proposed to mask a specific field in the text with a special mask token. This field is set differently at each epoch to reflect various hierarchies. Then, LLMs are fine-tuned to predict the masked field. (b) Inference process. The original CAD model is transformed into a structured text with a mask token replacing the part the user wants to change. The fine-tuned LLMs are provided with this masked text to generate diverse predictions, which are then converted into new CAD models by infilling and rendering.

Method

Figure 3: (a) An illustration for construction hierarchies of a CAD model. (b) Structured text representation for the CAD model shown in (a). The colors beneath the texts in (b) are used to indicate the relationship to construction hierarchies depicted in (a), *e.g.*, blue for a curve and green for a loop.

Method



Instruction: Below is a partial description of a CAD sequence where one command has been replaced with the string "[sketch-extrusion mask] [sketch-extrusion mask]":
[sketch-extrusion mask] [sketch-extrusion mask]
Generate a string that could replace "[sketch-extrusion mask] [sketch-extrusion mask]" in the CAD sequence:

Answer: arc,16,17,31,3 <curve_end> line,46,17 <curve_end> arc,46,45,31,59 <curve_end> line,16,45 <curve_end> <loop_end> <face_end> <sketch_end> add,31,38,31,31,31,1,0,0,0,1,0,0,0,1,31,31,31 <extrusion_end> circle,16,31,31,16,46,31,31,46 <curve_end> <loop_end> circle, 24,31,31,24,38,31,31,38 <curve_end> <loop_end> <face_end> <sketch_end> add,31,38,31,31,38,1,0,0,0,1,0,0,0,1,21,31,31 <extrusion_end>

Instruction: Below is a partial description of a CAD sequence where one command has been replaced with the string "[sketch-extrusion mask]":
arc,16,17,31,3 <curve_end> line,46,17 <curve_end> arc,46,45,31,59 <curve_end> line,16,45 <curve_end> <loop_end> <face_end> <sketch_end> add,31,38,31,31,31,1,0,0,0,1,0,0,0,1,31,31,31 <extrusion_end> [sketch-extrusion mask]
Generate a string that could replace "[sketch-extrusion mask]" in the CAD sequence:

Answer: circle,16,31,31,16,46,31,31,46 <curve_end> <loop_end> circle,24,31,31,24,38,31,31,38 <curve_end> <loop_end> <face_end> <sketch_end> add,31,38,31,31,38,1,0,0,0,1,0,0,0,1,21,31,31 <extrusion_end>

Instruction: Below is a partial description of a CAD sequence where one command has been replaced with the string "[loop mask] [loop mask]":
arc,16,17,31,3 <curve_end> line,46,17 <curve_end> arc,46,45,31,59 <curve_end> line,16,45 <curve_end> <loop_end> <face_end> <sketch_end> add,31,38,31,31,31,1,0,0,0,1,0,0,0,1,31,31,31 <extrusion_end> [loop mask] [loop mask] <face_end> <sketch_end> add,31,38,31,31,38,1,0,0,0,1,0,0,0,1,21,31,31 <extrusion_end>
Generate a string that could replace "[loop mask] [loop mask]" in the CAD sequence:

Answer: circle,16,31,31,16,46,31,31,46 <curve_end> <loop_end> circle,24,31,31,24,38,31,31,38 <curve_end> <loop_end>

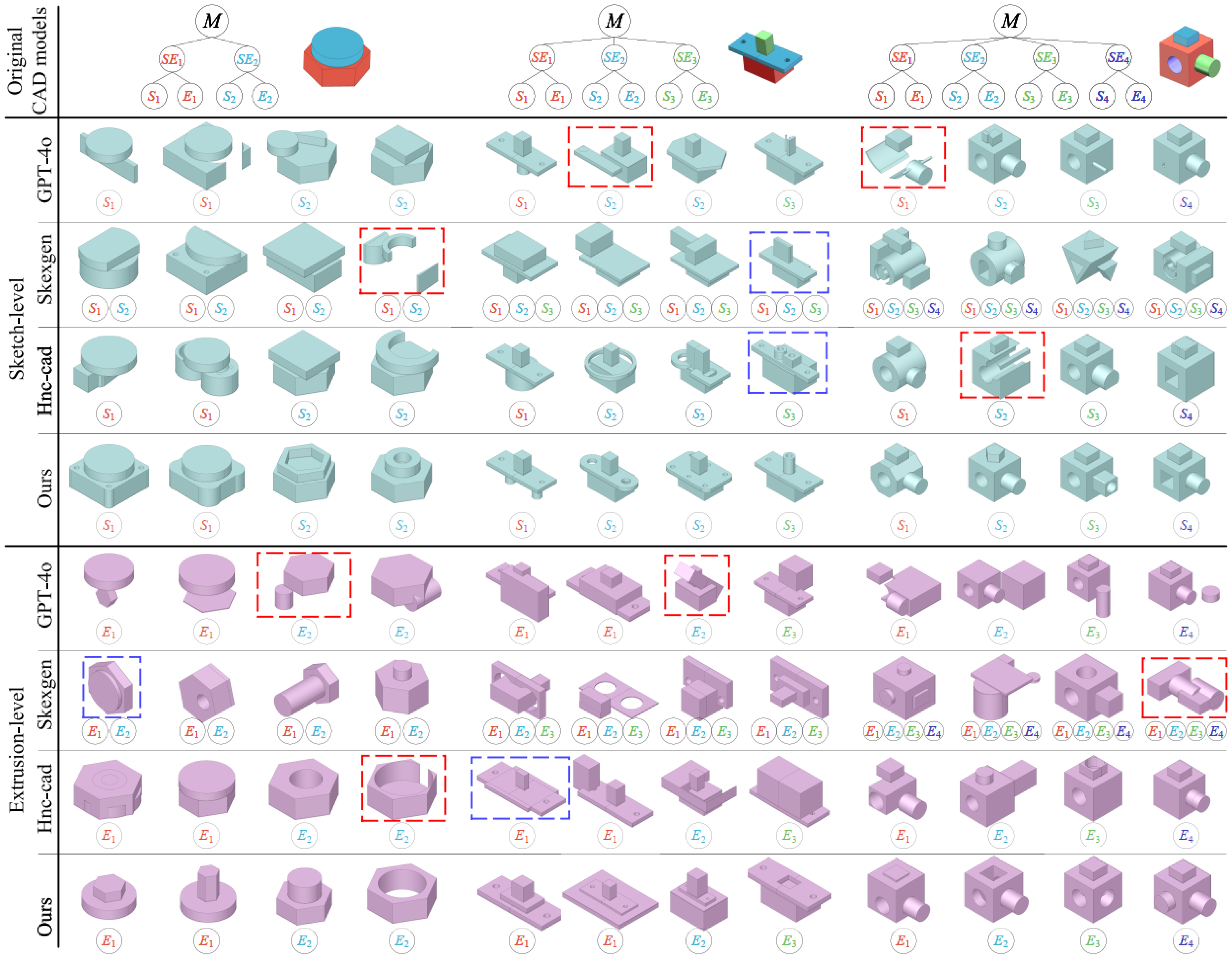
Instruction: Below is a partial description of a CAD sequence where one command has been replaced with the string "[arc mask] [line mask] [arc mask] [line mask]":
[arc mask] [line mask] [arc mask] [line mask] <loop_end> <face_end> <sketch_end> add,31,38,31,31,31,1,0,0,0,1,0,0,0,1,31,31,31 <extrusion_end> circle,16,31,31,16,46,31,31,46 <curve_end> <loop_end> circle,24,31,31,24,38,31,31,38 <curve_end> <loop_end> <face_end> <sketch_end> add,31,38,31,31,38,1,0,0,0,1,0,0,0,1,21,31,31 <extrusion_end>
Generate a string that could replace "[arc mask] [line mask] [arc mask] [line mask]" in the CAD sequence:

Answer: arc,16,17,31,3 <curve_end> line,46,17 <curve_end> arc,46,45,31,59 <curve_end> line,16,45 <curve_end>

Figure 4: (a) illustrates a CAD model and its structural diagram. (b), (c), (d) and (e) are four examples for prompt templates with the mask tokens designed to represent different construction hierarchies. The masked field for different hierarchies in the CAD model are highlighted in blue.

Experiments & Results

Method	Sketch-level							Extrusion-level						
	COV↑	MMD↓	JSD↓	Novel↑	Unique↑	PV↑	Realism↑	COV↑	MMD↓	JSD↓	Novel↑	Unique↑	PV↑	Realism↑
GPT-4o	58.2%	1.34	1.43	69.7%	72.8%	62.3%	23.2%	53.3%	1.42	2.14	58.6%	65.3%	48.8%	19.7%
SkexGen	60.6%	1.27	1.51	90.7%*	**93.5%**	68.7%	34.8%	63.6%	1.23	1.44	**89.3%**	89.1%*	76.1%	35.2%
Hnc-cad	62.4%*	1.21*	1.07*	87.6%	92.1%	72.6%*	36.3%*	65.6%*	1.25*	1.38*	86.2%	87.8%	79.7%*	38.0% *
Ours	**65.6%**	**1.19**	**0.82**	**92.1%**	92.6%*	**93.4%**	**39.6%**	**68.5%**	**1.19**	**1.32**	87.6%*	**90.4%**	**93.3%**	**42.1%**



Experiments & Results

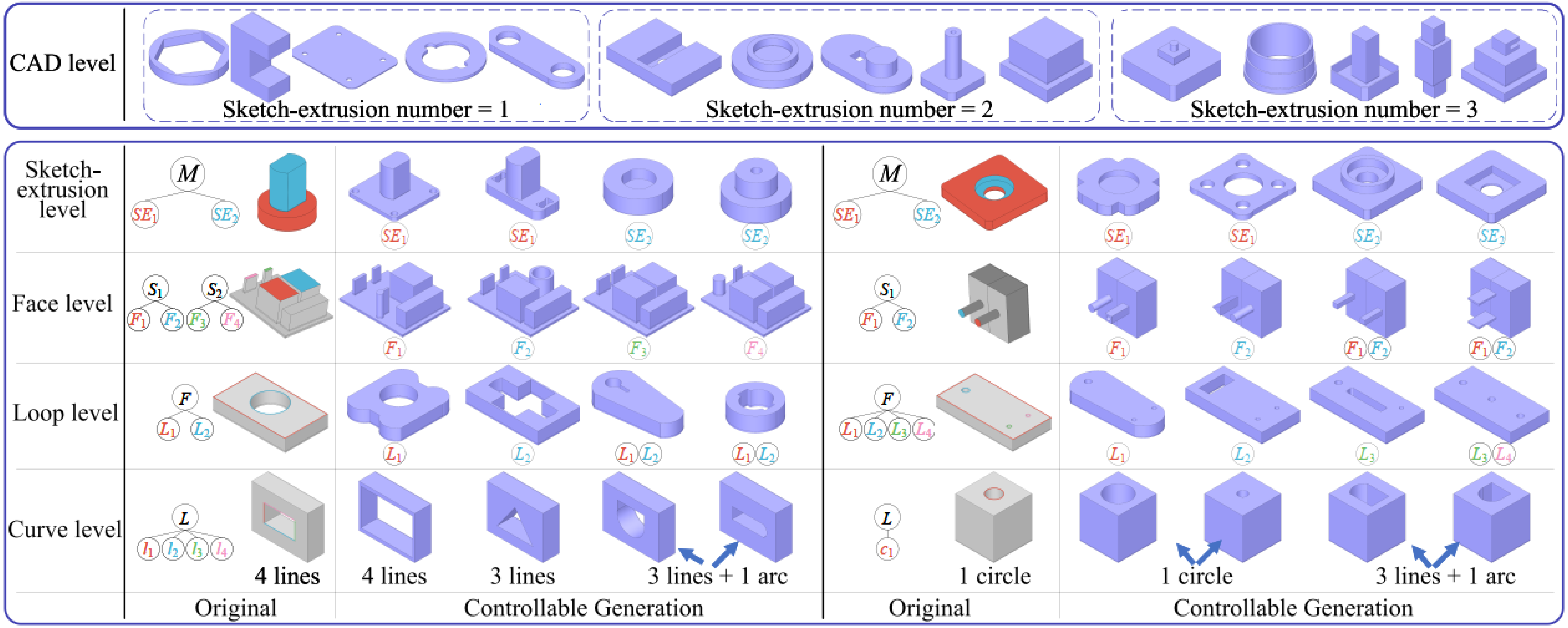


Figure 6: Our FlexCAD achieves controllable generation across different hierarchies, as introduced in 3.2. For the CAD level, we produce CAD models aligning with the required sketch-extrusion number. For the sketch-extrusion, face and loop levels, the left side of each sub-figure shows an original CAD model along with its local structural diagram. The color of each highlighted field matches that in the diagrams. The right side shows the predictions with only the masked part being masked and edited. And the masked part is marked below the predictions. Similarly, for the curve level, below the predictions are user-defined curve type and number. Best viewed in color.

Experiments & Results

Figure 7: Two examples of iterative editing. (a) Based on a simple CAD model, a new sketch-extrusion is generated by adding a [sketch-extrusion mask] at the end of the CAD text. Similarly, four peripheral internal loops are created. Then, a central quadrilateral loop is added and its geometry is altered. Finally, the new extrusion is adjusted to better match the original model. (b) Based on a complex CAD model, modifications can be progressively applied at various sketch-extrusions with the loop-extrusion-face-curve-loop level controllable generation to ultimately meet user needs.

Experiments & Results

Table 2: Ablation studies for fine-tuning LLMs with different settings. Pre-trained denotes the initial pre-trained weights. Full and LoRA indicate full and a few parameters are trainable, respectively. Transformer-4M is a small transformer-based Vaswani et al. (2017) language model. Its total number of trainable parameters is comparable to that of our model with LoRA. Llama-3-8B-From-Scratch and Llama-3-8B-Full denote training full parameters without and with the initial pre-trained weights, respectively. Llama-3-8B-Instruct is an instruction-tuned model in an 8B size Meta (2024). For Llama-3-70B Meta (2024), we fine-tune only 0.023% of its parameters, around 16.3 million. Best performances are in **bold** and the second-bests are marked by $*$.

Model	COV↑	MMD↓	JSD↓	Novel↑	Unique↑	PV↑
Transformer-4M (w/o Pre-trained, Full)	59.4%	1.37	1.02	85.8%	86.9%	80.2%
Llama-3-8B-From-Scratch (w/o Pre-trained, Full)	63.0%	1.23	0.91	89.7%	90.2%	89.5%
Llama-3-8B-Full (w/ Pre-trained, Full)	66.4%*	1.20	0.85	92.6%*	91.0%	91.7%
Llama-3-8B-Instruct (w/ Pre-trained, LoRA)	65.3%	1.22	0.89	91.4%	92.1%*	90.5%
Llama-3-8B (w/ Pre-trained, LoRA, ours)	65.6%	1.19*	**0.82**	92.1%	**92.6%**	93.4%*
Llama-3-70B (w/ Pre-trained, LoRA)	**68.2%**	**1.13**	0.84*	**93.0%**	91.8%	**94.6%**

Table 3: Effectiveness analysis of the hierarchy-aware masking strategy and unified training. Random Masking denotes randomly masking 15%-50% continuous tokens within each CAD text, instead of the hierarchy-aware field. w/o Hierarchy-specific Tokens indicates that when masking, we utilize the generic token [mask], rather than employing hierarchy-specific mask tokens, such as [face mask], [loop mask] and etc. w/o Unified Training represents that we solely train a single task, *i.e.*, the sketch-level controllable generation. Best performances are in **bold**.

Model	COV↑	MMD↓	JSD↓	Novel↑	Unique↑	PV↑
Random Masking	63.0%	1.25	1.02	88.2%	91.5%	90.6%
w/o Hierarchy-specific Tokens	63.7%	1.20	0.95	90.8%	91.7%	91.5%
w/o Unified Training	64.3%	**1.17**	0.89	91.6%	90.9%	92.2%
Ours	**65.6%**	1.19	**0.82**	**92.1%**	**92.6%**	**93.4%**

Thank you!