Supplementary Materials of CADMorph: Geometry-Driven Parametric CAD Editing via a Plan–Generate–Verify Loop

Anonymous Author(s)
Affiliation
Address
email

A Outline

This supplementary material will cover the following aspects in addition to the main paper. It will cover the following aspects.

- Justification about using voxelized tSDFs for 3D data representation.
- More detailed qualitative results.
- The method to draw Figure 3 in the main paper.
- Analysis of inference time between different methods.
- Full prompts of experiments on GPT-series.
- An illustrative visualization of the iterative editing process.
- Discussion on failure cases.
- More details of human evaluation.
- Broader Impact.

B Why CADMorph Uses a Voxel-Based Truncated SDF (tSDF)

Boundary fidelity over appearance. A CAD construction sequence encodes an object solely through the precise, watertight *surface* produced by sketches and extrusions. Representations that are (i) only *approximate* (e.g. point clouds, Gaussian splats) or (ii) *view-conditioned* (e.g. NeRF) inject non-geometric signals (texture, lighting) that CAD editing neither requires nor exploits. A geometry-centric format is therefore mandatory.

Compatibility with convolutional attention. Among geometry-centric formats, the contenders are meshes and voxels. CADMorph inspects cross-attention maps inside a 3-D U-Net based transformer, mirroring how 2-D diffusion models align pixels with text. This inspection demands a *regular grid* so that sliding convolutions capture spatial correlations—an assumption satisfied by voxels but violated by irregular meshes. Within voxel grids, a binary occupancy field provides only inside/outside signals, whereas a **truncated signed-distance field (tSDF)** additionally offers signed distance, furnishing sub-voxel geometric detail and richer gradients for learning.

Boolean algebra and extensibility. CAD solids arise from Boolean combinations of sketch–extrusion primitives. SDFs support these combinations analytically:

$$f_{A\cup B}(\mathbf{x}) = \min\big(f_A(\mathbf{x}),\ f_B(\mathbf{x})\big), \tag{1}$$

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

$$f_{A\setminus B}(\mathbf{x}) = \max\big(f_A(\mathbf{x}),\, -f_B(\mathbf{x})\big), \tag{2}$$

$$f_{A\cap B}(\mathbf{x}) = \max\big(f_A(\mathbf{x}),\, f_B(\mathbf{x})\big). \tag{3}$$

Because Eqs. (1)–(3) are differentiable and grid-aligned, CADMorph can *compose* or *extend* primitive sequences by operating directly on their tSDFs—a capability we plan to exploit for more intricate construction sequences in future work.

In this sense, voxelized tSDFs strike the best balance: exact boundary geometry, seamless integration with convolutional attention, and elegant Boolean composition. These properties make tSDFs the natural choice for CADMorph.

C More Qualitative Results.

In this section we will put forward more qualitative results. The revised construction sequences, along with the their renderings, are shown from Figure 5 to Figure 28. From the figures we can discover that *all* methods are able to produce grammatically coherent construction sequences, but the modality gap between the construction sequence and the corresponding geometric shape introduces a huge deviation between the rendered result and the target shape, thus calling for a specially-designed geometric module that brings the corresponding geometric information explicitly to the generated construction sequences.

D Method to Draw the Cross-attention Map.

In this section we briefly illustrate how to project the cross-attention map onto the retrieved mesh of the tSDF. We first follow the practice in prompt-to-prompt to obtain the cross-attention map of the LDM in the DDIM inversion process.

Then the attention map is averaged and reshaped into the same size as the shape latent z, and linearly interpolated into the unit 3D voxel space afterwards. The interpolated attention score for each token is then projected onto the derived mesh vertices of the SDF.

For all the projected SDF scores, we then apply a low-pass filter and only remains the top 10% of them. Finally, on the vertices where no points remain, we further perform an 8 nearest-neighbor search and assign its value as the average of the values via inverse distance weighting.

E Inference Time Cost.

In this section, we compare the time costs of various editing methods. As shown in Table 1, our method, CADMorph, requires several minutes to edit a single example. This time cost is consistent with other recent 3D generation and editing methods that combine diffusion-based models with per-shape optimization [Poole et al., 2022, Lin et al., 2023, Liu et al., 2023, Sun et al., 2024, Yi et al., 2024, Haque et al., 2023, Fang et al., 2024, Wu et al., 2024]. Although CADMorph seems to be slower than purely feedforward approaches or LLM-based API calls, this is expected given its reliance on diffusion processes, LLMs and iterative optimization strategies. The observed latency reflects the trade-off between performance and computational overhead inherent in such hybrid frameworks.

To evaluate time efficiency more comprehensively, we report both the elapsed time via API usage and the raw inference time on the official ChatGPT interface for methods involving closed-source LLMs. Notably, the o4-mini and o4-mini-high variants show longer elapsed times, which we attribute partially to possible queuing delays in the API service. To validate this, we tested ten samples using the official ChatGPT interface (`https://chatgpt.com/`), where we measured the average inference time excluding API latency. The results confirm that while queuing introduces some overhead, the primary time consumption originates from the visual reasoning process itself. In particular, these models often perform iterative inspection of multi-view inputs, leading to increased computation time. In comparison, CADMorph's total elapsed time is only about 20% longer than GPT-based reasoning variants, while offering improved editing performance. This suggests a favorable trade-off between runtime and quality, especially when compared to feedforward models such as CAD-Diffuser and FlexCAD, which are limited in their ability to exploit additional test-time computation for better performance.

Table 1: Time usage for different editing methods. we report both the average elapsed time on the apis and the inference time on the official web interface for methods using closed source LLMs, as apis may be affected by congestions. The methods that are run locally remain the same time usage in the two criteria.

Method	Elapse Time (Minutes)↓	Inferece Time (Minutes)↓
Feedforward methods.		
CAD-Diffuser	0.18	0.18
FlexCAD	0.13	0.13
Methods Using closed-source LLMs. (Approximation)		
GPT-4o	1.27	1.07
o4-mini	5.36	3.24
o4-mini-high	6.02	5.32
Our method.		
CADMorph	7.26	7.26

F Full prompts of experiments on GPT-series.

We follow the same prompting method for the comparison experiments on GPT-4o, o4-mini and o4-mini-high. The prompts are listed in Listing 1. In practice, the image stacks four different views of the target geometric shape.

Listing 1: Prompts used in comparative experiments in GPT-4o, o4-mini, and o4-mini-high. The prompt first tells the exact rules of writing the construction sequences of a CAD model, then both the original parametric sequence and the target geometric shape are given to the model.

```
{
Input: original_param_seq, rendered_img_of_stacked_mult_views
prompt0 = """
Below is a CAD file encoded into a structuralized text. The grammar of the CAD file
is as follows.
Carefully read the below grammar, try to understand the structuralized text and
render it in your heart.
A CAD file consists of several sketch and extrusion pairs.
A sketch consists of several loops with [SOL] resembling the start of a loop.
Each loop consists of several primitives which could be lines, arcs or circles with
their corresponding parameters.
For a line, its corresponding parameters are the 2D cordinates of its endpoint.
For a circle, its corresponding parameters are the 2D cordinates of its heart and
its radius.
For an arc, its corresponding parameters are the 2D cordinates of its endpoint, its
sweep angle and a flag showing whether it is counterclockwise.
Extrusion also has its parameters, which resembles the 3D sketch plane orientation,
the xyz origin in the 3D sketch plane, its scale of associated sketch profile,its
extrude distances toward both sides, the type of boolens operation between the it
and the privious sketch-extrusion pair, and the type of the extrusion.
Each sketch-extrusion pair is followed by a [SEP] token, and an [EOS] token at the
end of the construction sequence.
Note that for all the parameters which does not reflect the relationships, they are
normalized and quantized into 256 bins.
"""
prompt1 = """
Use what you get from the structuralized text shown above,carefully read this image,
 which is a multi-view rendering of a CAD model, modify the structuralized text so
that it corresponds to the given image. Think step by step before answering.
"""
prompt2 = """
Remove all the unnecessary words in your response and give me the modifiled CAD file
 using the format given above.
"""
user_prompt_1 = prompt0 + "\n" + original_param_seq + "\n" + prompt1 \
```

```
    + load_image_to_base64_url(rendered_img_of_stacked_mult_views)
user_prompt_2 = prompt2
}
```

The `user_prompt_2` is given to GPTs after `user_prompt_1` is responded, where `user_prompt_1`, `response_1` and `user_prompt_2` are sent to GPT altogether.

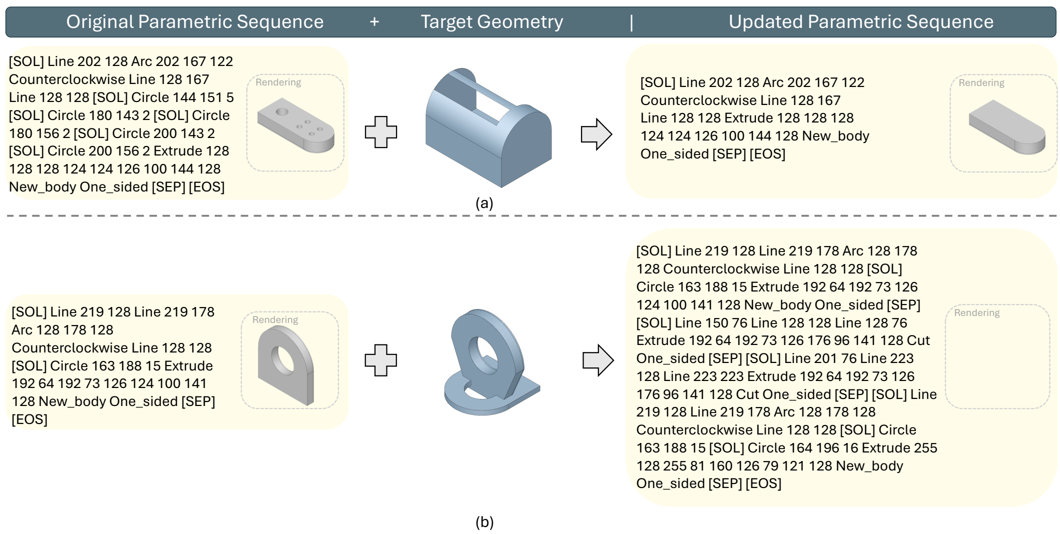


Figure 1: Some typical failure cases. The rendering left blank means the sequence id not able to render visible shapes.

G Visualization of the editing process.

In this section we put forward some illustrative examples on the visualization of the editing process via peeking the renderings of all the shapes in the queue in different turns. From Figure 4(a), we can discover that the parametric sequence quickly saturates to the right modification when the edit is easy. From Figure 4(b), we can discover that when the modification is hard to complete, the queue is slowly updated, namely, each iteration of CADMorph slowly changed the construction sequence to better fit the shape. It is also worth noting that in some cases, the queue remains stagnant, thereby showing the importance of the queue, which prevents the model from changing for the worse.

H Failure Cases.

In this section we put forward some failure cases. Despite the case discussed for the bottom example of Figure 6 of the main paper where CADMorph shows a certain robustness to inaccurate shape given, here we put some more representative failure cases.

Some typical failure cases are shown in Figure 1. From Figure 1(a) we can discover that CADMorph successfully removes the hole but stops at somewhere in the middle, as it fails to propose a mechanism that can bore the empty space in the middle of the arch while remaining the bar at the top, which is also quite challenging for experienced human designers. Another example is shown at Figure 1(b). The construction sequence shows that CADMorph tried to reuse the given shape and cut the unnecessary parts out. However, it turns out that the edited construction sequence fails to render a valid shape. We hypothesize that the reason might be an error in the CAD kernel, thereby showing that the MPP model still needs to acquire knowledge about the topology of CAD shapes and the rendering mechanism of the CAD kernel, which we leave as future work.

I Details of Human Evaluation.

Here we put forward more details of Human Evaluations mentioned in the main paper. The cover of human evaluations and an example question is illustrated in Figure 2 and Figure 3 respectively.

J Broader Impact.

This paper propose CADMorph, an automatic CAD editing method driven by geometric guidance. The method could reduce the labor of human designers for manual shape modifications. However, the edited mechanical part may cause harm to the entire assembly if it is not thoroughly examined by human designers.

User Study on the Quality of Edited
CAD Construction Sequences under Different Methods

You are invited to answer this questionnaire. Your answer is of great importance to assess the quality of different CAD editing methods.

Below are the edited CAD construction sequences under 6 different methods. Would you please kindly rank the editing results on the editing quality.

For the sequence you think has the best editing quality, please rank 1. For the sequence you think has the worst editing quality , please rank 6.

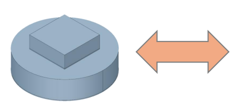
Hint: If you have no idea about the editing quality, you can first look at the **rendering** of each method in the dashed box and **compare it with the target shape**, you can also read the *construction sequence* and compare it with *the sequence in the input line*. Give your overall ranking of all methods on a 50-50 basis of the two criteria.

Write your rankings in the boxes next to the results for each question.

Note: Each CAD sequence is in a flattened CSG tree format and is encoded as the following shows.

Commands	Parameters
⟨SOL⟩	$\emptyset$
L (Line)	x, y : line end-point
A (Arc)	x, y : arc end-point α : sweep angle f : counter-clockwise flag
R (Circle)	x, y : center r : radius
E (Extrude)	θ, ϕ, γ : sketch plane orientation p_x, p_y, p_z : sketch plane origin s : scale of associated sketch profile e_1, e_2 : extrude distances toward both sides b : boolean type, u : extrude type
⟨EOS⟩	$\emptyset$

Table 1. **CAD commands** and their parameters. ⟨SOL⟩ indicates the start of a loop; ⟨EOS⟩ indicates the end of the whole sequence.



[SOL] Circle 176 128 48 Extrude 128 128 128 32 159 127 105 164 128 New_body One_sided [SEP] [SOL] Line 223 128 Line 223 223 Line 128 223 Line 128 128 Extrude 128 128 128 88 86 163 69 152 128 New_body One_sided [SEP] [EOS]

Figure 2: The cover of the user study.

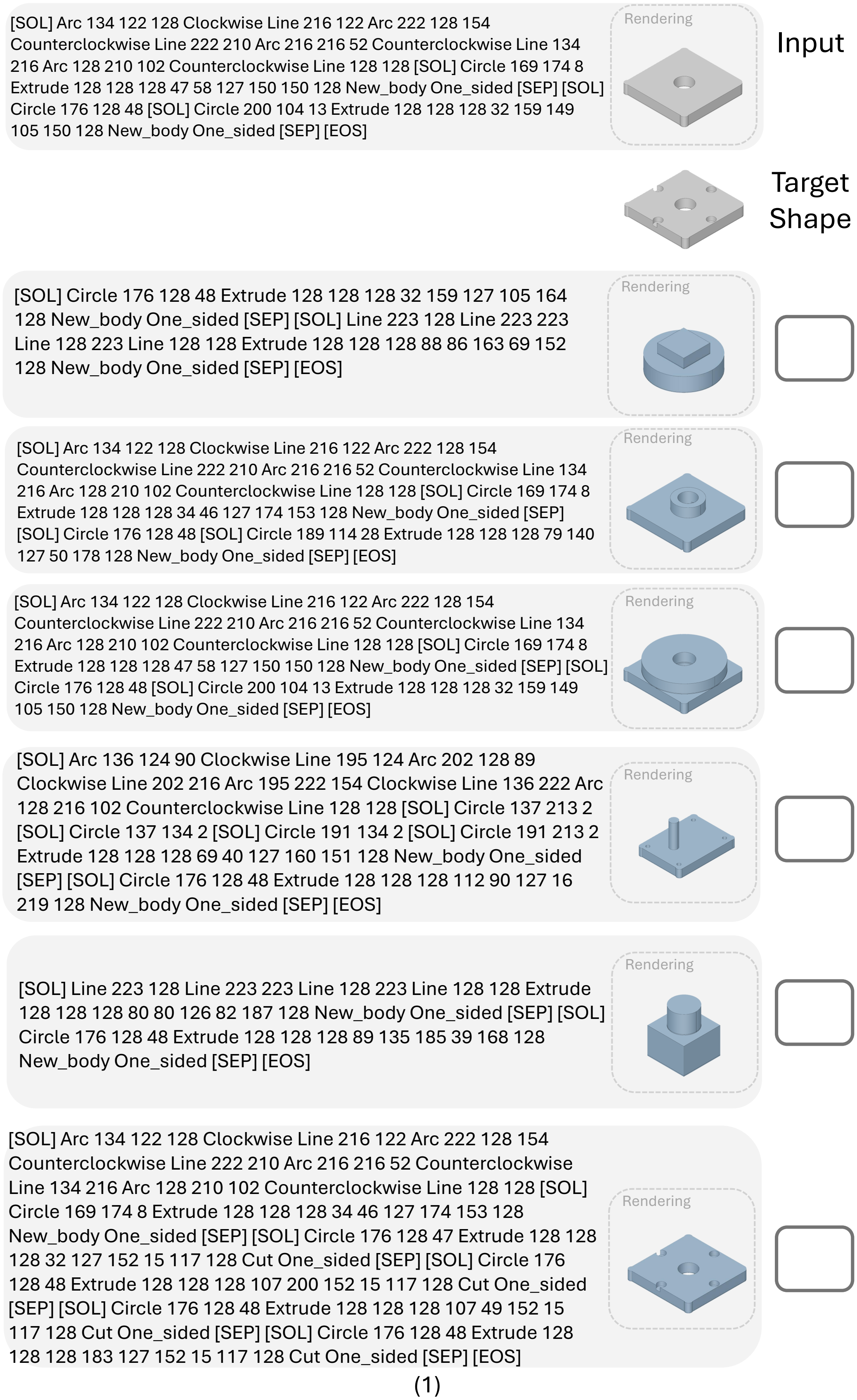


Figure 3: An Example question in the human evaluation.

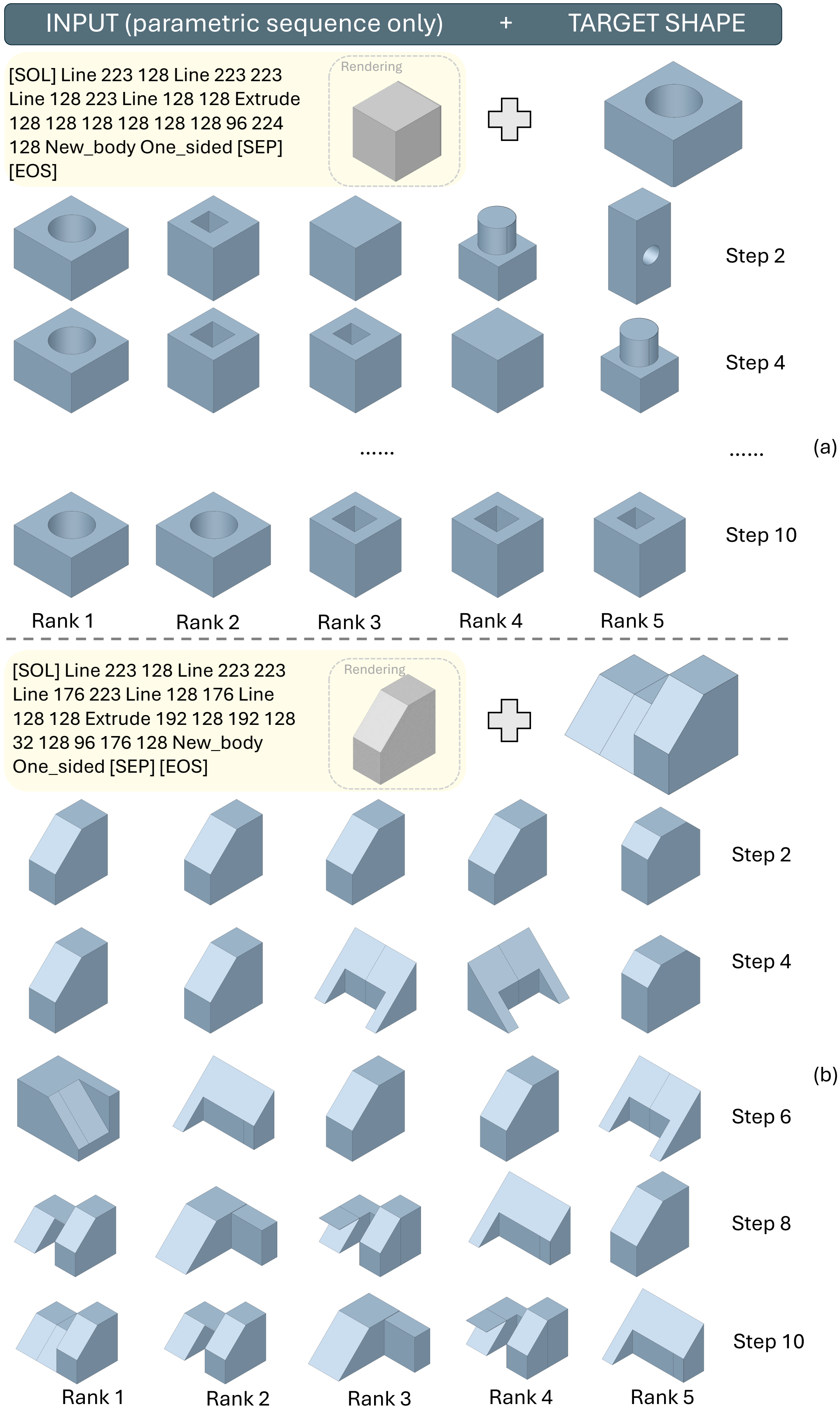
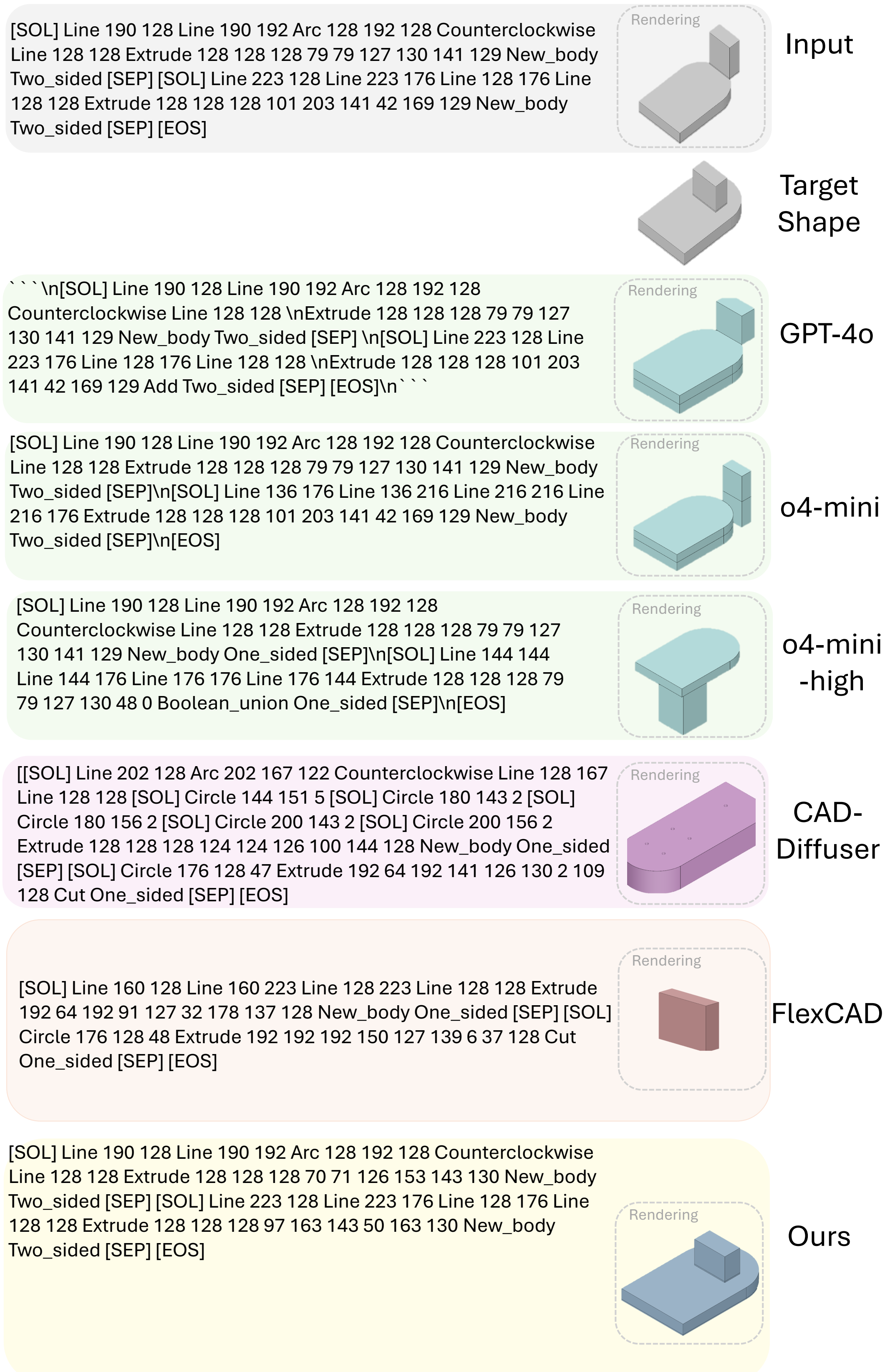
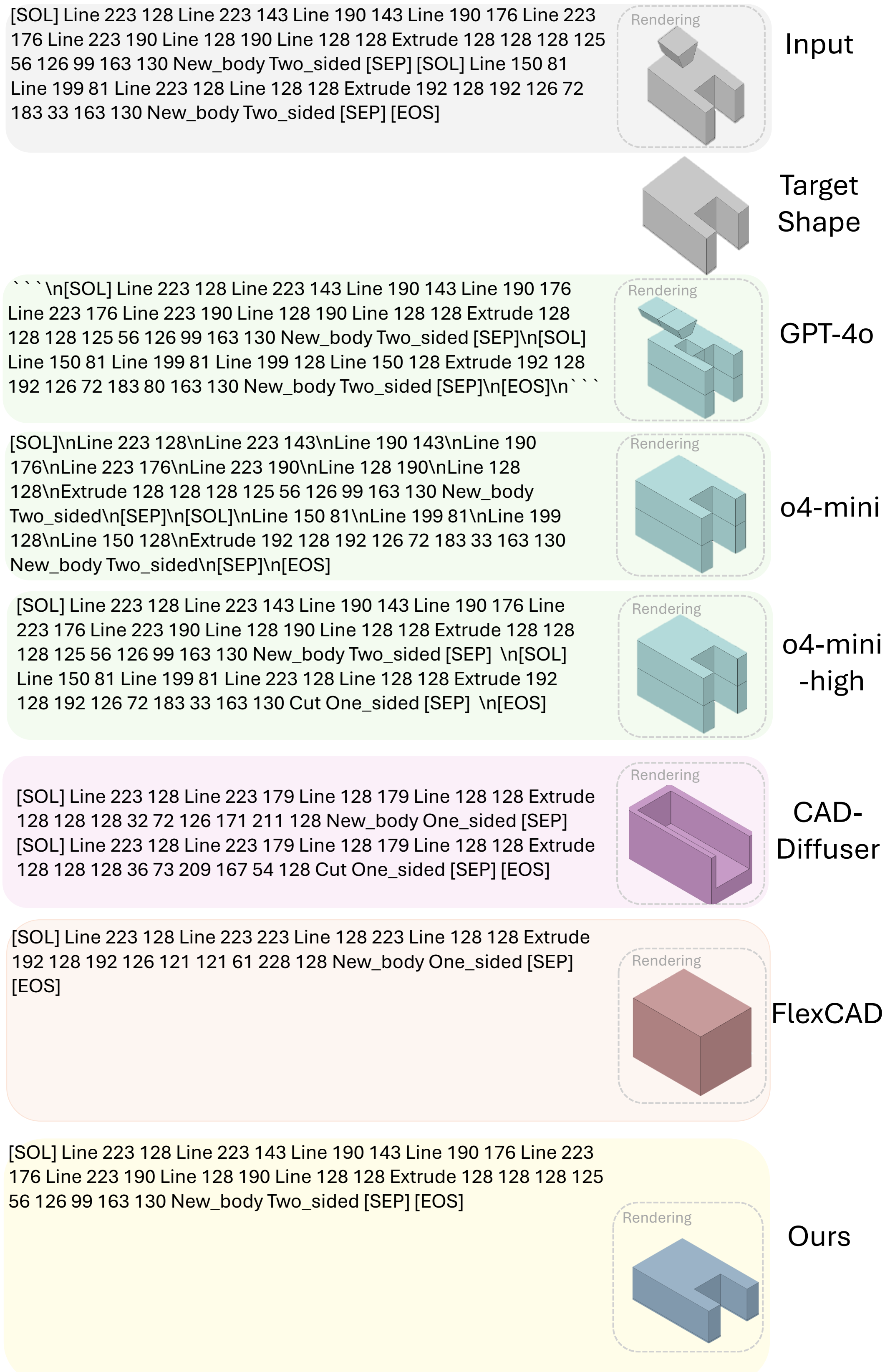


Figure 4: Visualization of the priority queue throughout the editing process. We sample every two editing iterations during the editing process. The queue remains not significantly changed is omitted by "..." for brevity.



(1)

Figure 5: Full results of the original parametric sequence (with renderings), target shapes, and the edited sequence of different methods (with renderings). Blank represents not able to generate any sequence or the sequence is not able to render shapes after removing irrelevant characters. (1 of 24)

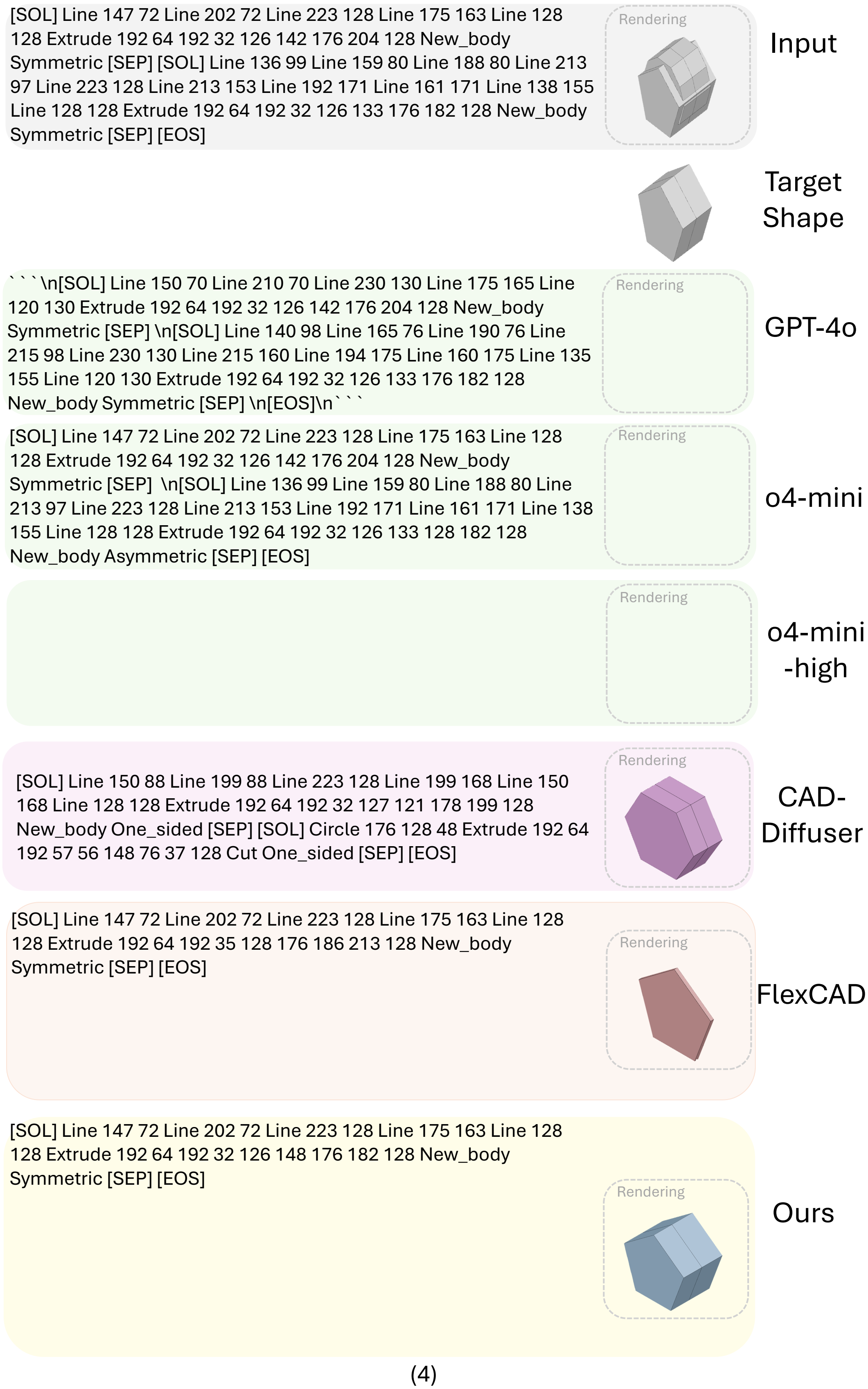


(2)

Figure 6: Full results of the original parametric sequence (with renderings), target shapes, and the edited sequence of different methods (with renderings). Blank represents not able to generate any sequence or the sequence is not able to render shapes after removing irrelevant characters. (2 of 24)

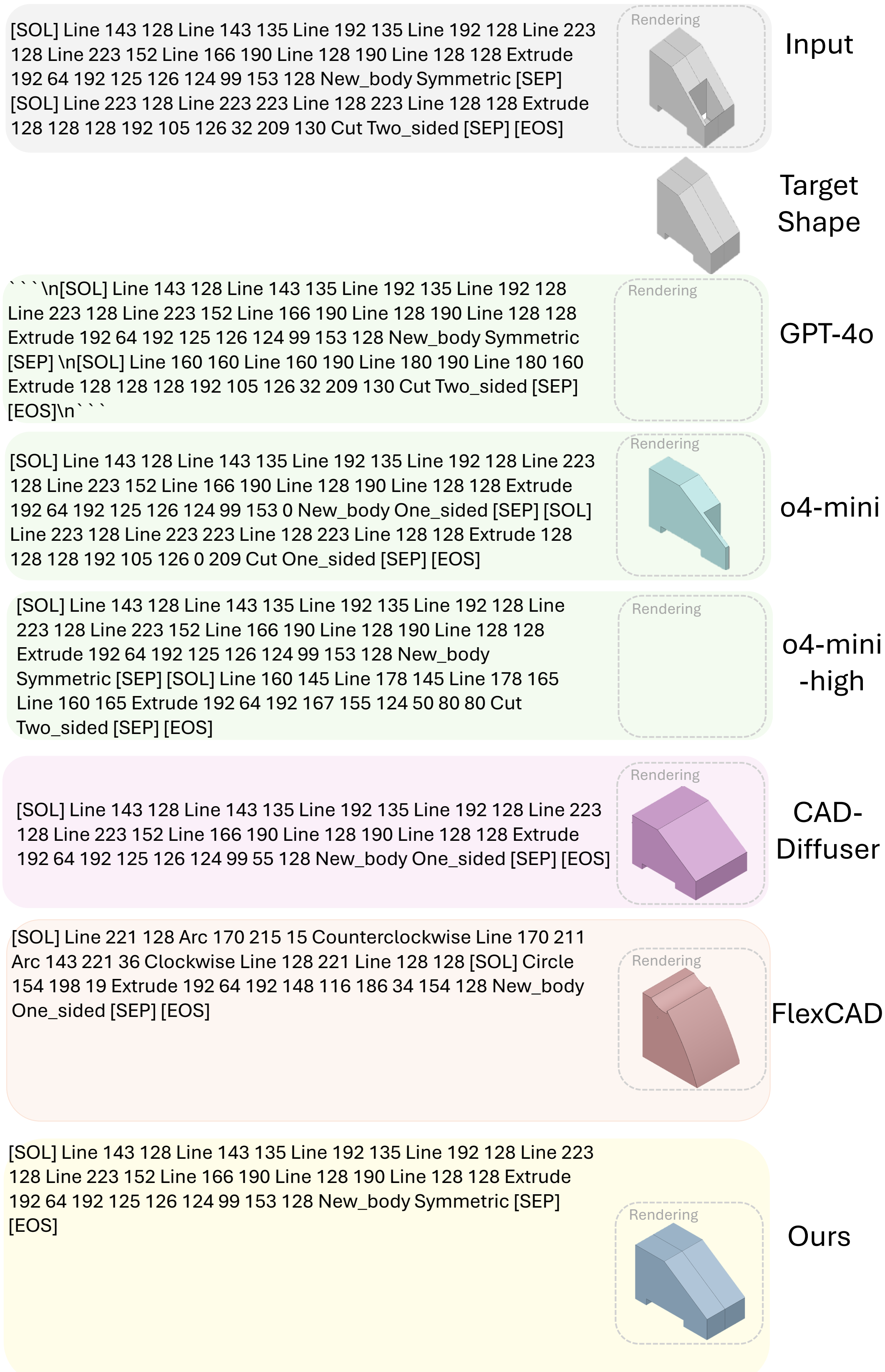
(3)

Figure 7: Full results of the original parametric sequence (with renderings), target shapes, and the edited sequence of different methods (with renderings). Blank represents not able to generate any sequence or the sequence is not able to render shapes after removing irrelevant characters. (3 of 24)



(4)

Figure 8: Full results of the original parametric sequence (with renderings), target shapes, and the edited sequence of different methods (with renderings). Blank represents not able to generate any sequence or the sequence is not able to render shapes after removing irrelevant characters. (4 of 24)

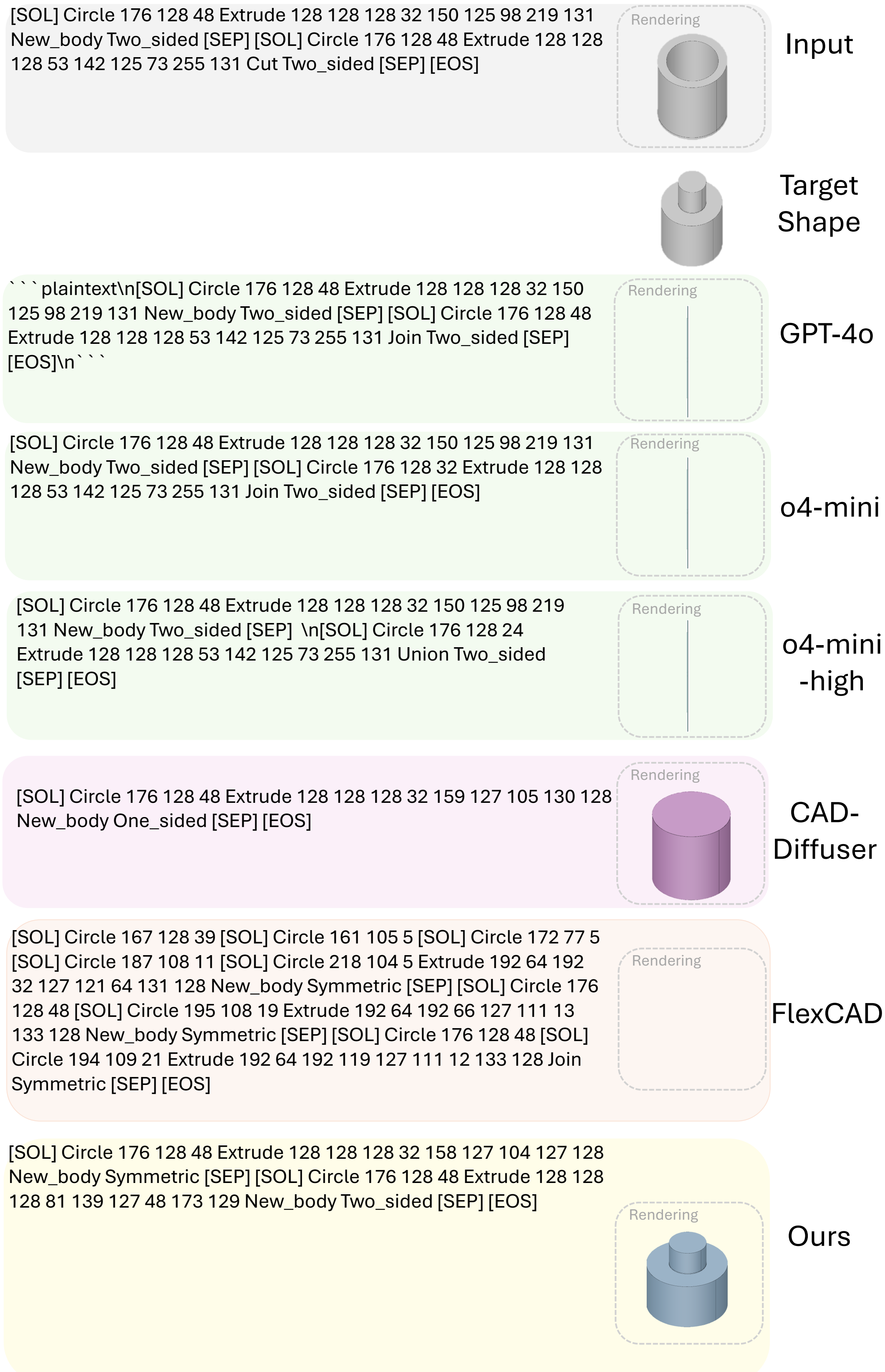


(5)

Figure 9: Full results of the original parametric sequence (with renderings), target shapes, and the edited sequence of different methods (with renderings). Blank represents not able to generate any sequence or the sequence is not able to render shapes after removing irrelevant characters. (5 of 24)

(6)

Figure 10: Full results of the original parametric sequence (with renderings), target shapes, and the edited sequence of different methods (with renderings). Blank represents not able to generate any sequence or the sequence is not able to render shapes after removing irrelevant characters. (6 of 24)



(7)

Figure 11: Full results of the original parametric sequence (with renderings), target shapes, and the edited sequence of different methods (with renderings). Blank represents not able to generate any sequence or the sequence is not able to render shapes after removing irrelevant characters. (7 of 24)

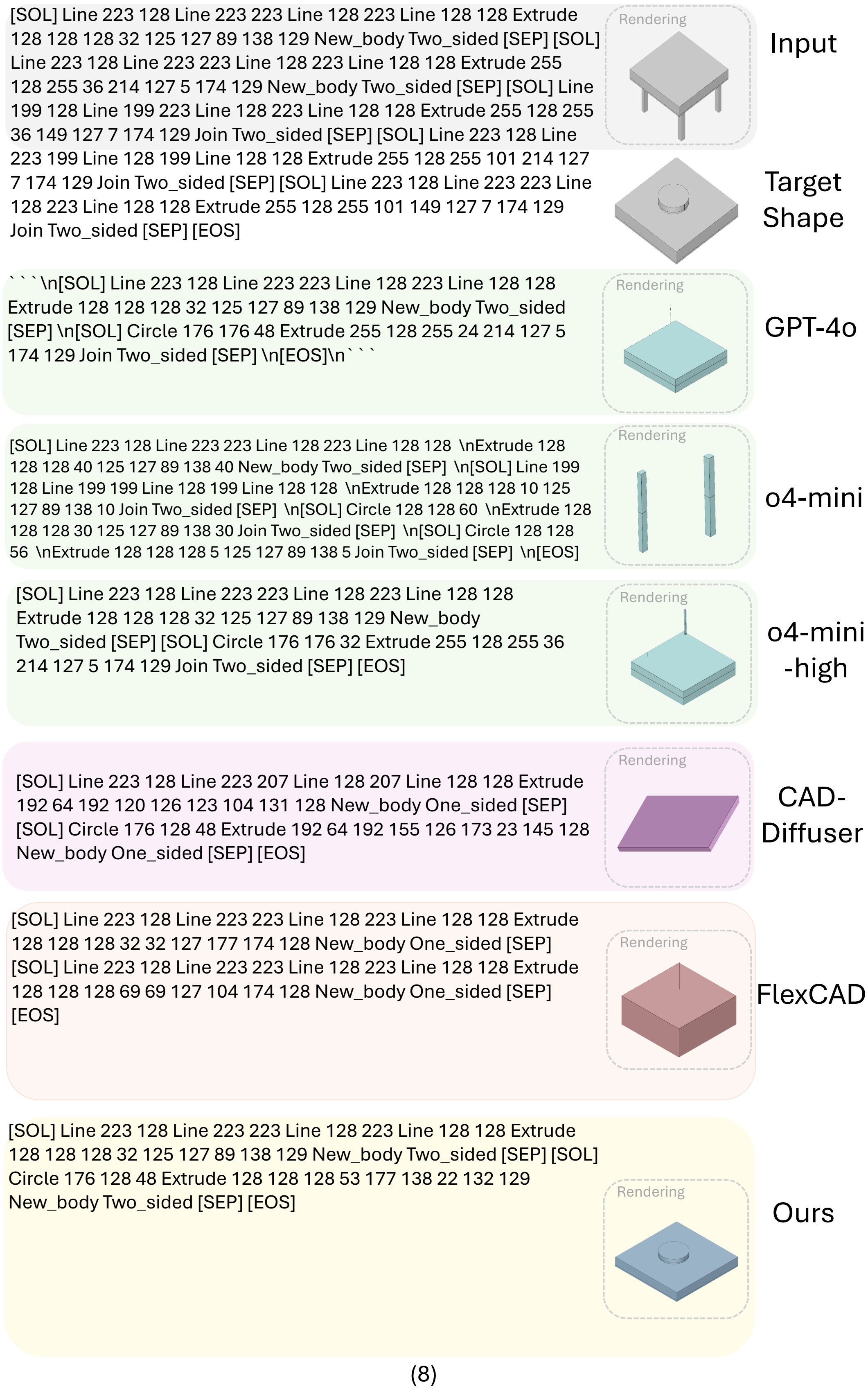
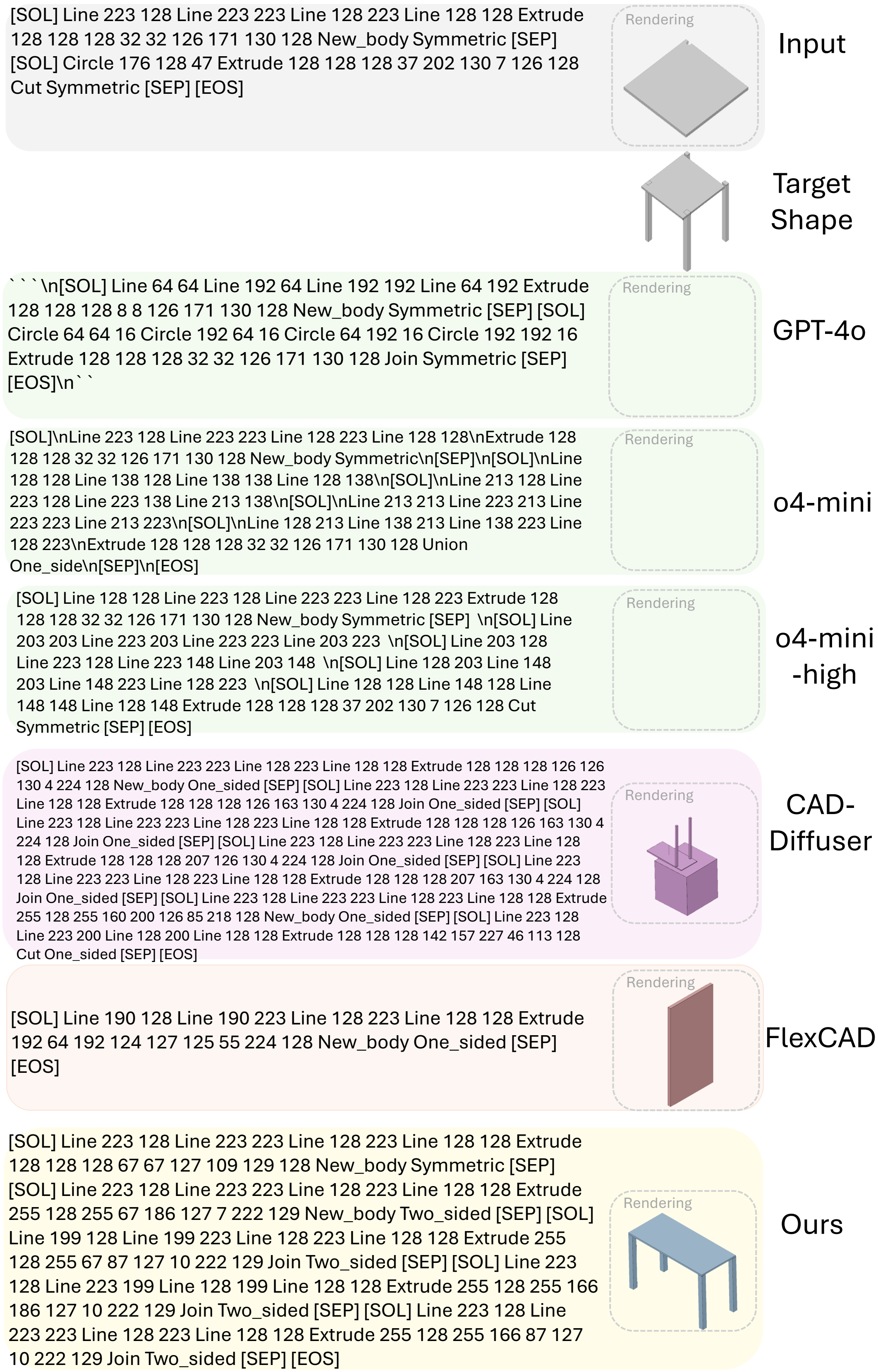


Figure 12: Full results of the original parametric sequence (with renderings), target shapes, and the edited sequence of different methods (with renderings). Blank represents not able to generate any sequence or the sequence is not able to render shapes after removing irrelevant characters. (8 of 24)



(9)

Figure 13: Full results of the original parametric sequence (with renderings), target shapes, and the edited sequence of different methods (with renderings). Blank represents not able to generate any sequence or the sequence is not able to render shapes after removing irrelevant characters. (9 of 24)

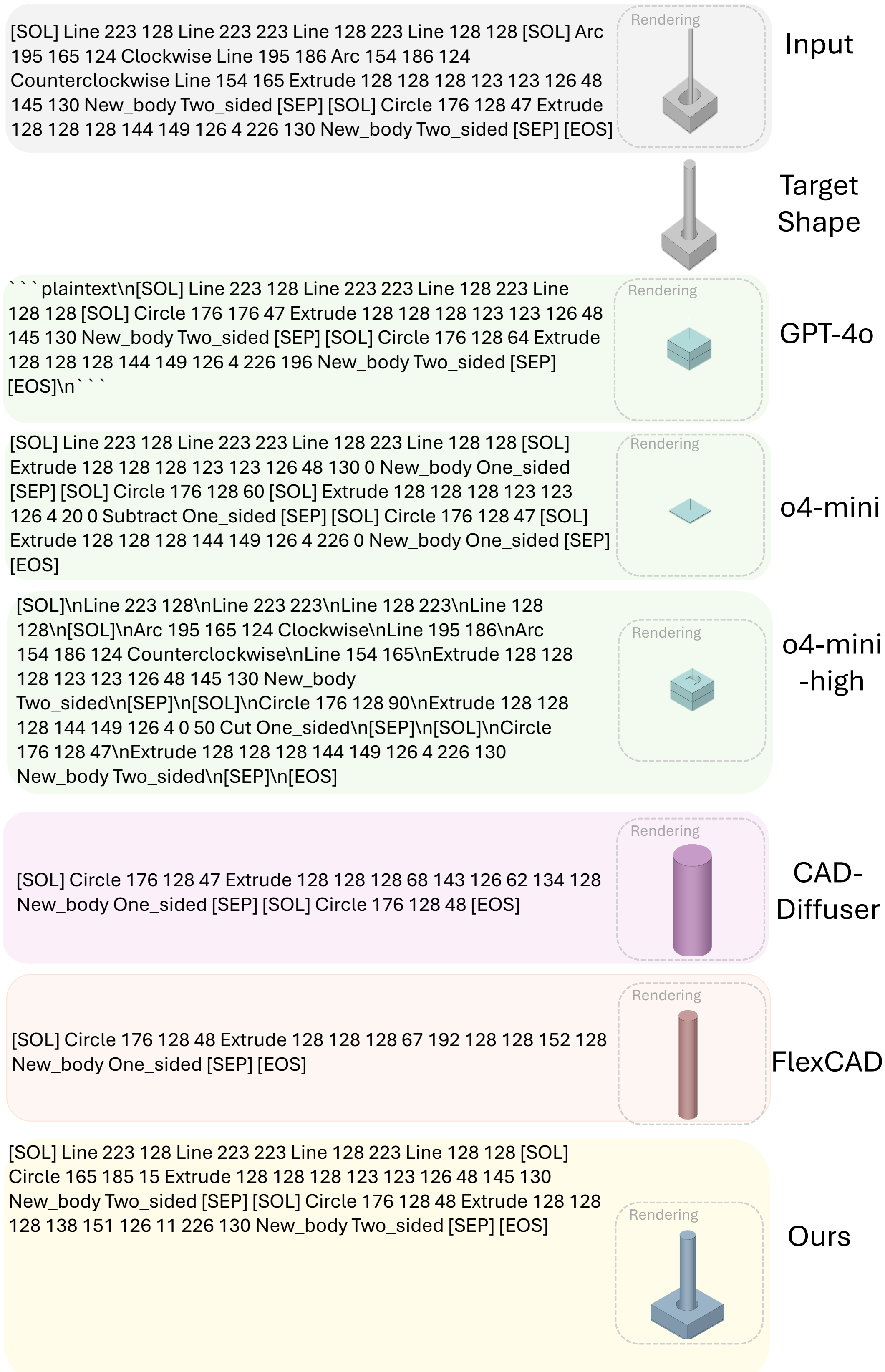
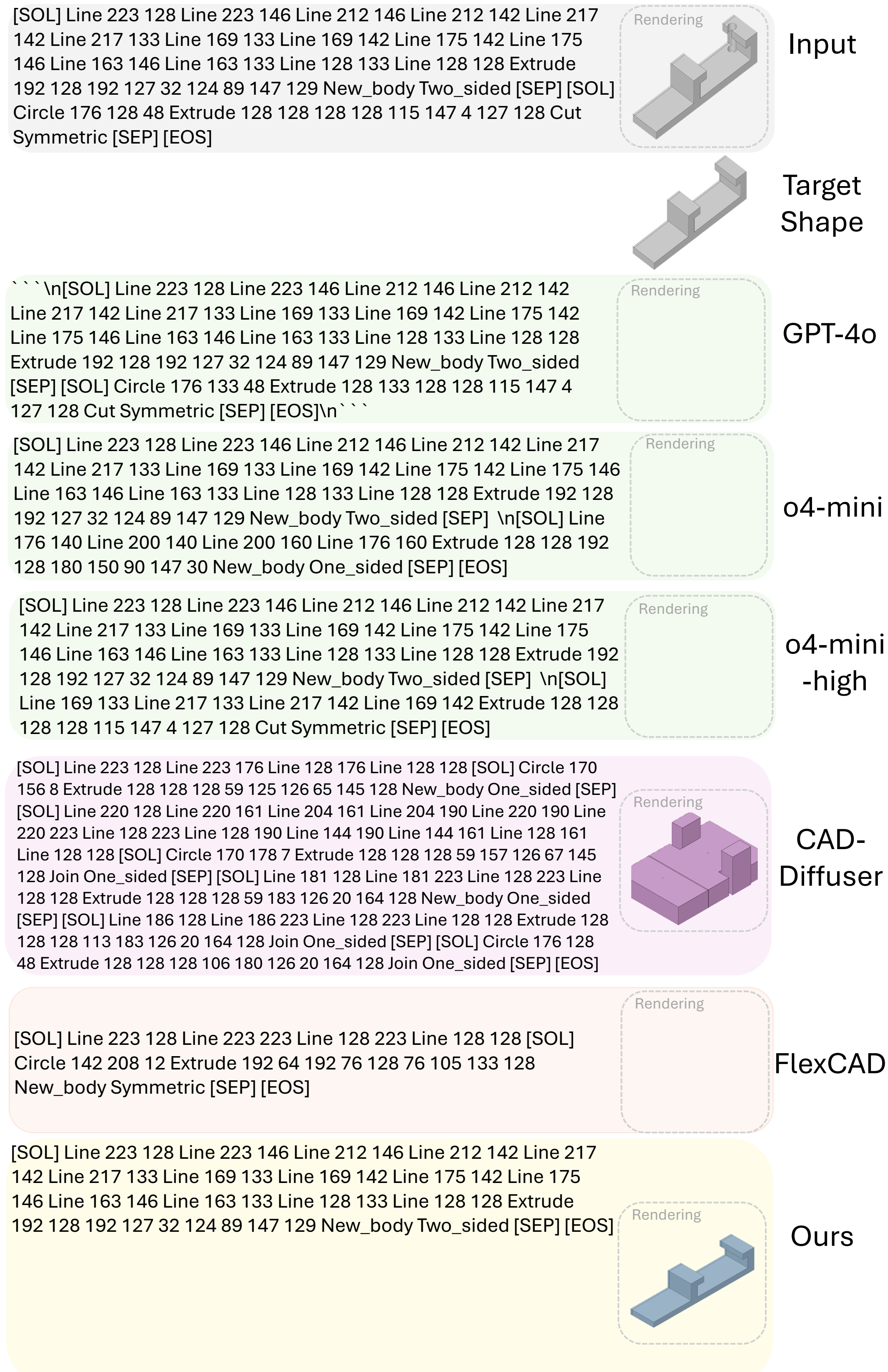


Figure 14: Full results of the original parametric sequence (with renderings), target shapes, and the edited sequence of different methods (with renderings). Blank represents not able to generate any sequence or the sequence is not able to render shapes after removing irrelevant characters. (10 of 24)



(11)

Figure 15: Full results of the original parametric sequence (with renderings), target shapes, and the edited sequence of different methods (with renderings). Blank represents not able to generate any sequence or the sequence is not able to render shapes after removing irrelevant characters. (11 of 24)

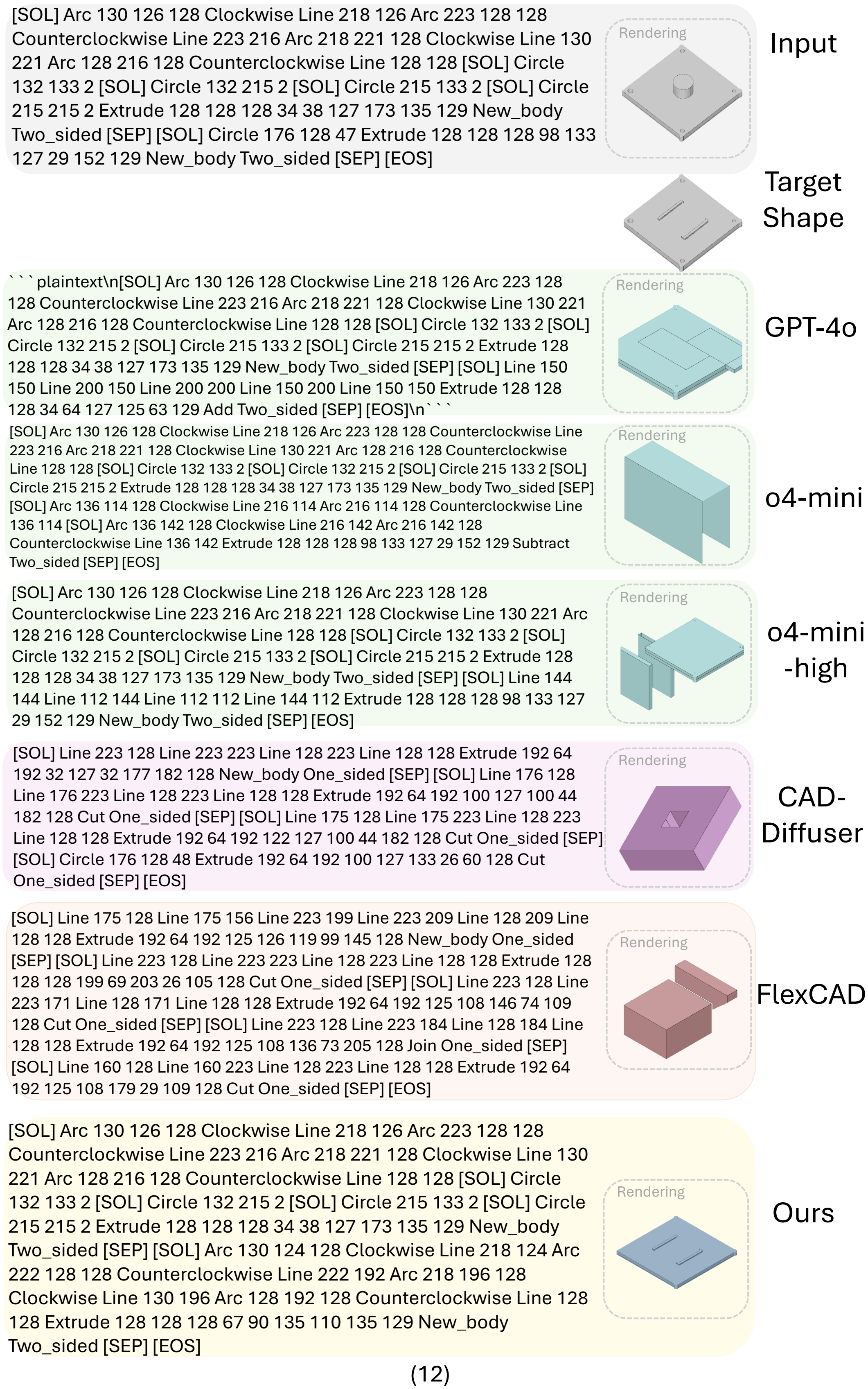
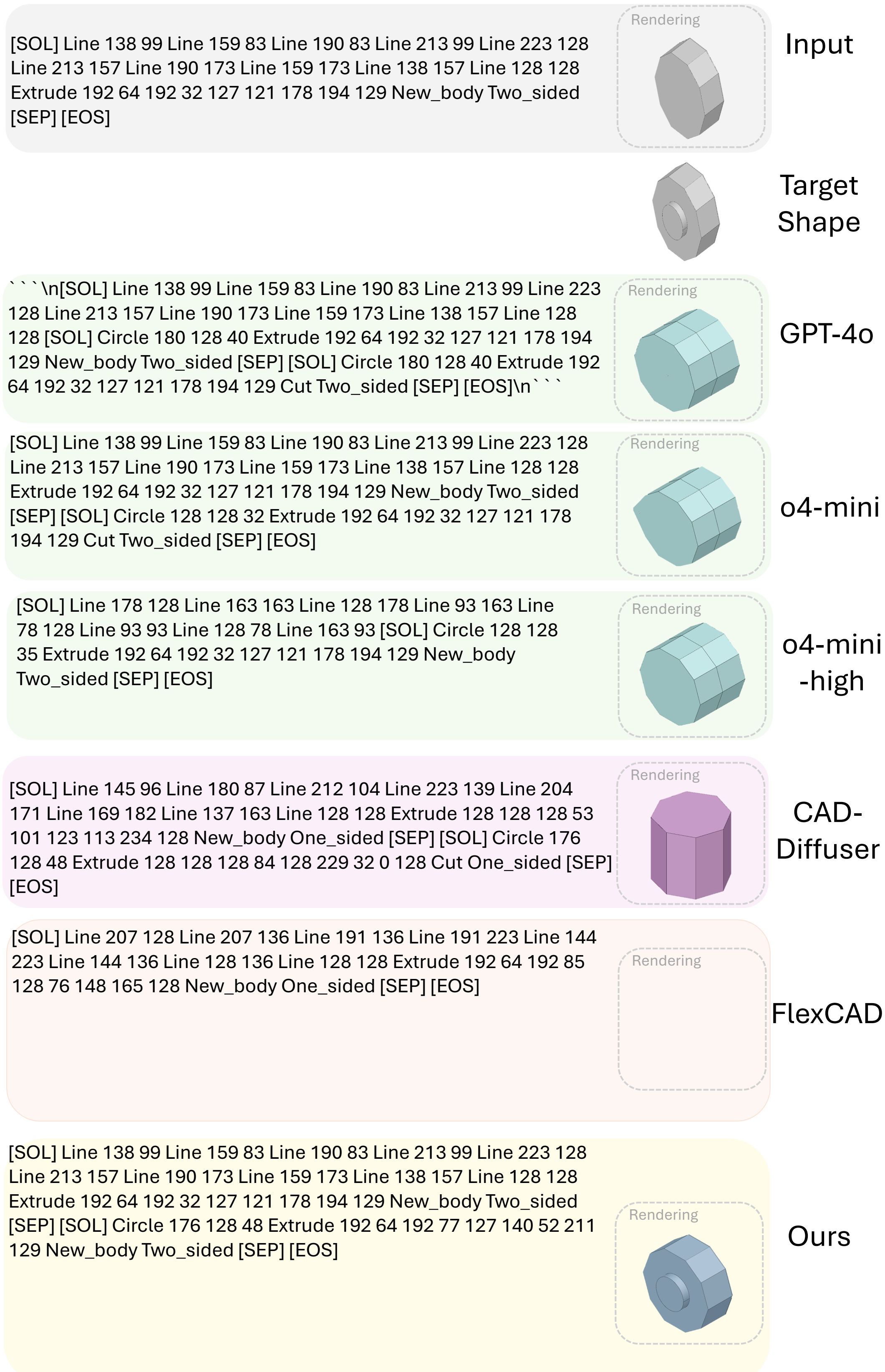


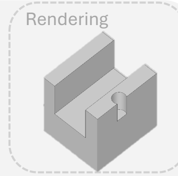
Figure 16: Full results of the original parametric sequence (with renderings), target shapes, and the edited sequence of different methods (with renderings). Blank represents not able to generate any sequence or the sequence is not able to render shapes after removing irrelevant characters. (12 of 24)



(13)

Figure 17: Full results of the original parametric sequence (with renderings), target shapes, and the edited sequence of different methods (with renderings). Blank represents not able to generate any sequence or the sequence is not able to render shapes after removing irrelevant characters. (13 of 24)

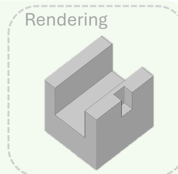
[SOL] Line 223 128 Line 223 188 Line 202 188 Line 202 156 Line 147 156 Line 147 188 Line 128 188 Line 128 128 Extrude 192 64 192 124 127 121 86 222 129 New_body Two_sided [SEP] [SOL] Circle 176 128 48 Extrude 128 128 128 196 76 183 12 127 128 Cut Symmetric [SEP] [EOS]



Input

Target Shape

```\n[SOL] Line 223 128 Line 223 188 Line 202 188 Line 202 156 Line 147 156 Line 147 188 Line 128 188 Line 128 128 Extrude 192 64 192 124 127 121 86 222 129 New_body Two_sided [SEP] \n[SOL] Line 147 188 Line 188 188 Line 188 156 Line 147 156 Extrude 128 128 128 196 76 183 12 127 128 Cut Symmetric [SEP] \n[EOS]\n```



GPT-4o

[SOL] Line 223 128 Line 223 188 Line 202 188 Line 202 156 Line 147 156 Line 147 188 Line 128 188 Line 128 128 Extrude 192 64 192 124 127 121 86 222 129 New_body Two_sided [SEP] [SOL] Line 147 156 Line 147 188 Line 202 188 Line 202 156 Extrude 128 128 128 196 76 183 12 127 128 Cut Symmetric [SEP] [EOS]

Rendering

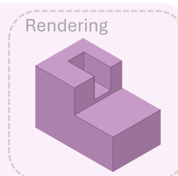
o4-mini

[SOL]\nLine 223 128\nLine 223 188\nLine 202 188\nLine 202 156\nLine 147 156\nLine 147 188\nLine 128 188\nLine 128 128\nExtrude 192 64 192 124 127 121 86 222 129 New_body Two_sided\n[SEP]\n[SOL]\nLine 153 128\nLine 153 156\nLine 207 156\nLine 207 128\nLine 153 128\nExtrude 128 128 128 196 76 183 12 127 128 Cut Symmetric\n[SEP]\n[EOS]

Rendering

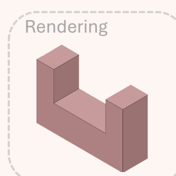
o4-mini -high

[SOL] Line 223 128 Line 223 165 Line 176 165 Line 176 194 Line 128 194 Line 128 128 Extrude 192 64 192 48 127 59 156 224 128 New_body One_sided [SEP] [SOL] Line 223 128 Line 223 160 Line 128 160 Line 128 128 Extrude 128 128 128 73 71 127 97 176 128 Cut One_sided [SEP] [EOS]



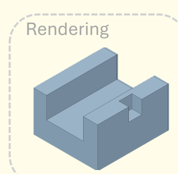
CAD-Diffuser

[SOL] Line 223 128 Line 223 188 Line 204 188 Line 204 156 Line 147 156 Line 147 188 Line 128 188 Line 128 128 Extrude 192 64 192 46 126 70 178 180 128 New_body One_sided [SEP] [SOL] Circle 176 128 48 Extrude 192 64 192 115 71 123 5 92 128 Cut One_sided [SEP] [EOS]



FlexCAD

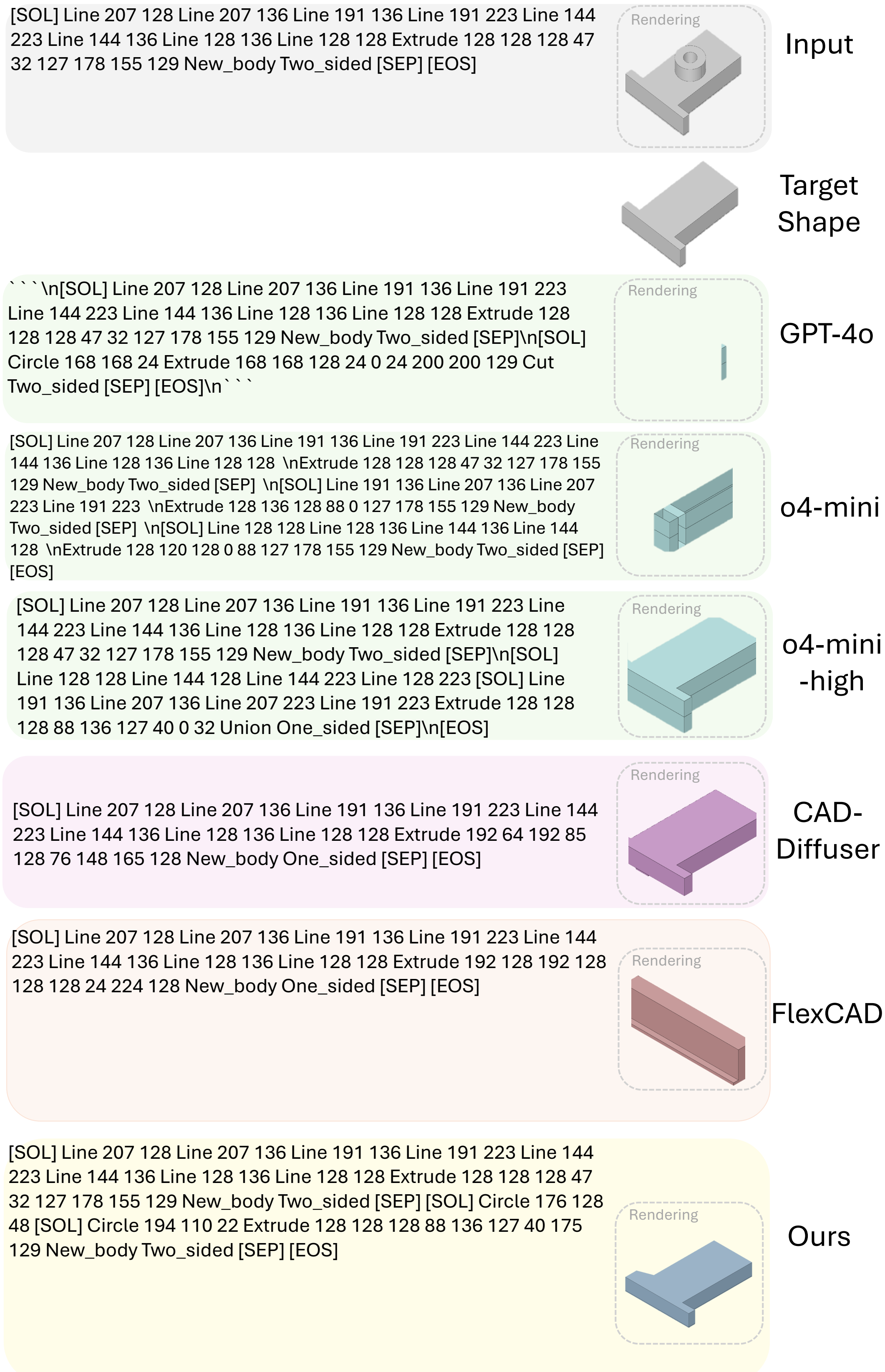
[SOL] Line 223 128 Line 223 188 Line 202 188 Line 202 156 Line 147 156 Line 147 188 Line 128 188 Line 128 128 Extrude 192 64 192 124 127 121 86 222 129 New_body Two_sided [SEP] [SOL] Line 223 128 Line 223 223 Line 128 223 Line 128 128 Extrude 128 128 128 197 74 183 18 127 128 Cut Symmetric [SEP] [EOS]



Ours

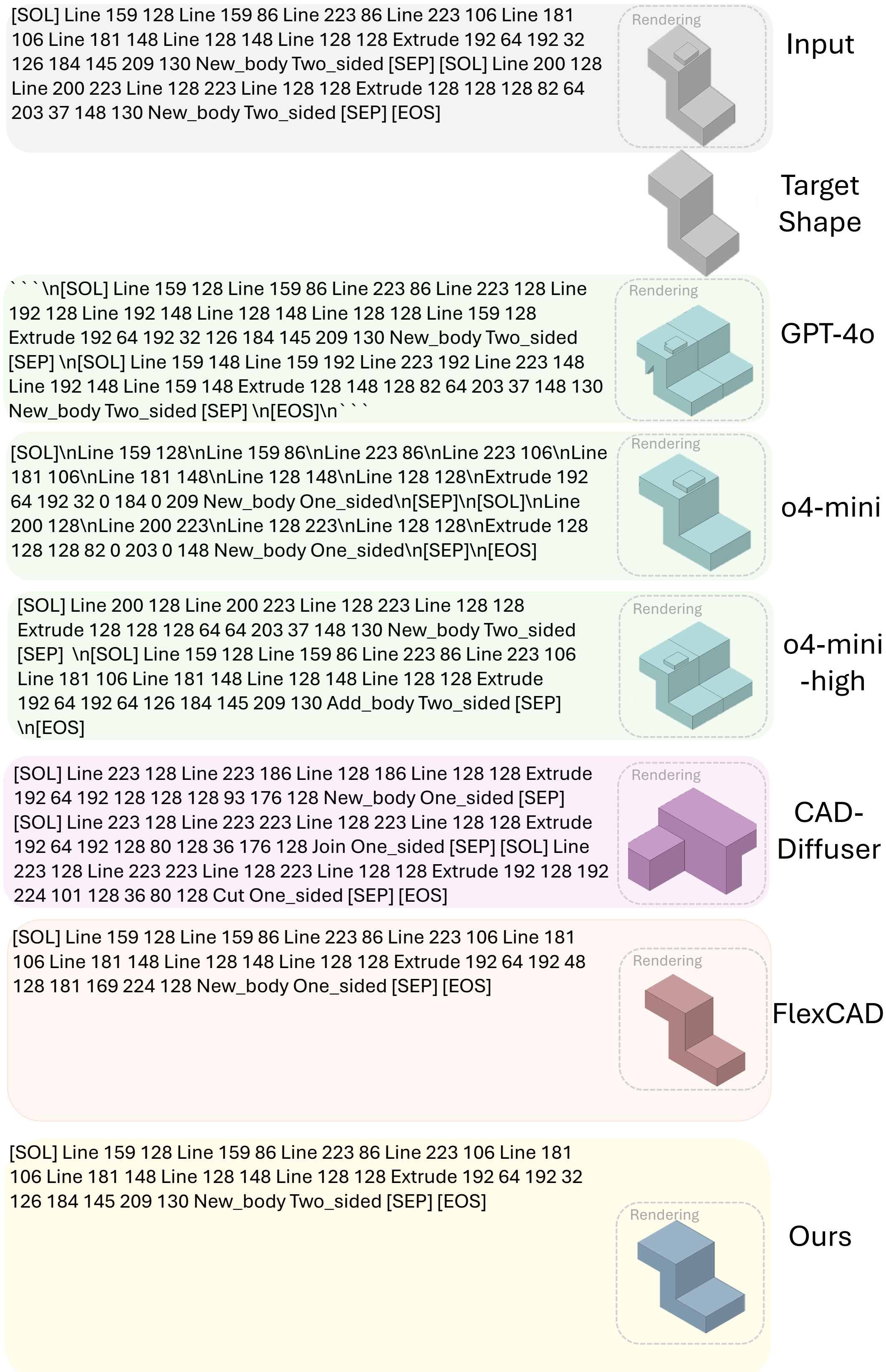
(14)

Figure 18: Full results of the original parametric sequence (with renderings), target shapes, and the edited sequence of different methods (with renderings). Blank represents not able to generate any sequence or the sequence is not able to render shapes after removing irrelevant characters. (14 of 24)
```



(15)

Figure 19: Full results of the original parametric sequence (with renderings), target shapes, and the edited sequence of different methods (with renderings). Blank represents not able to generate any sequence or the sequence is not able to render shapes after removing irrelevant characters. (15 of 24)



(16)

Figure 20: Full results of the original parametric sequence (with renderings), target shapes, and the edited sequence of different methods (with renderings). Blank represents not able to generate any sequence or the sequence is not able to render shapes after removing irrelevant characters. (16 of 24)

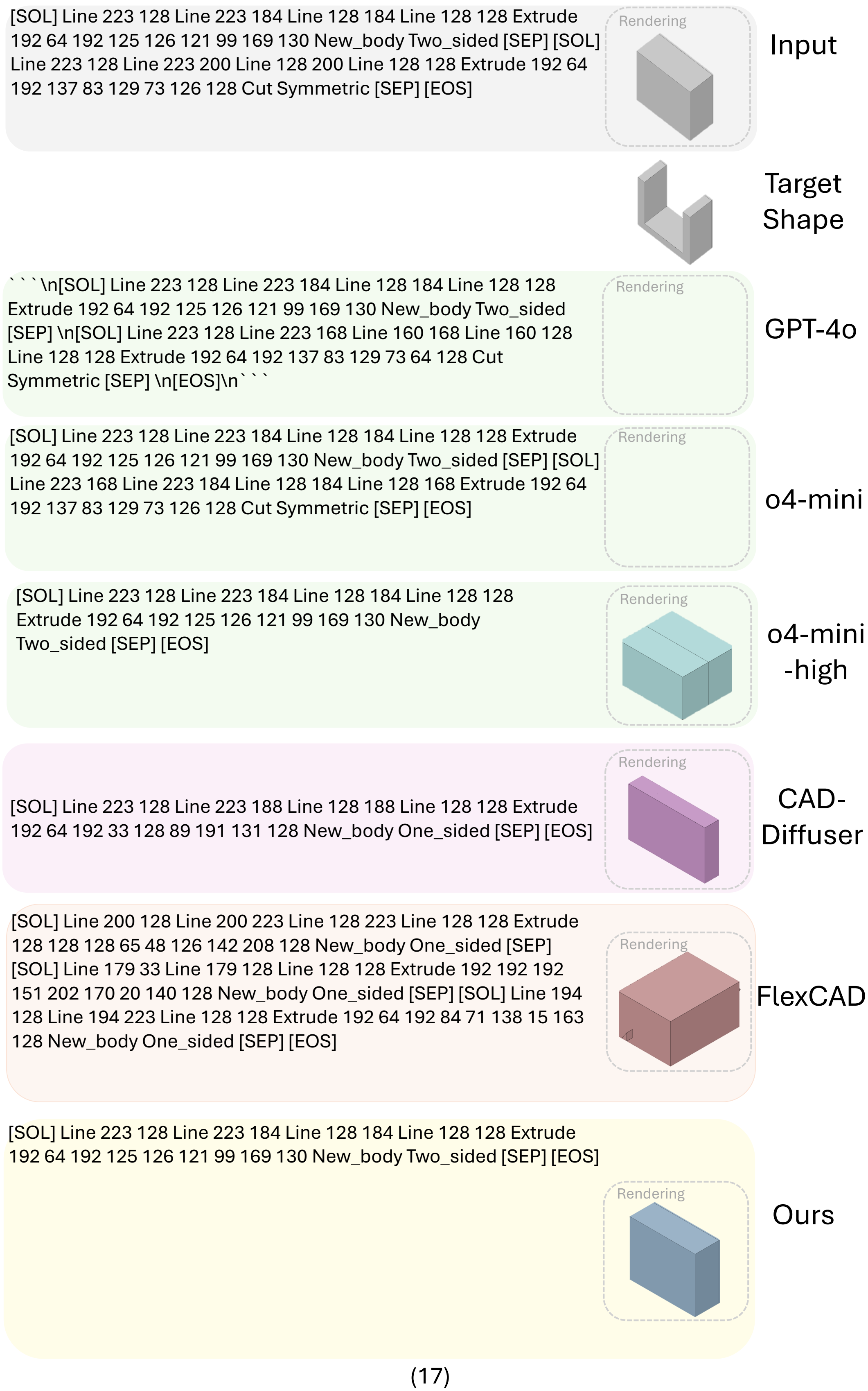
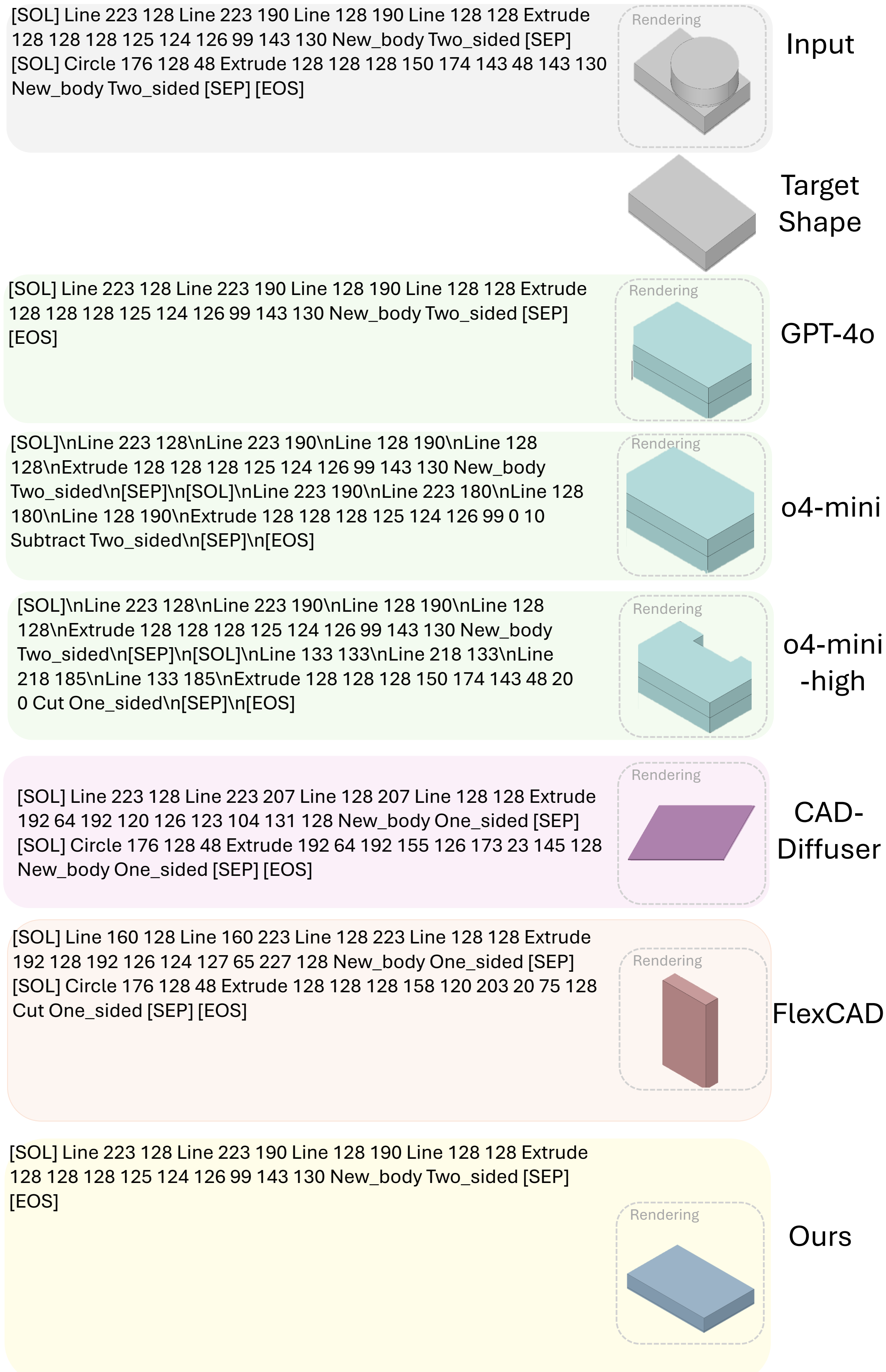
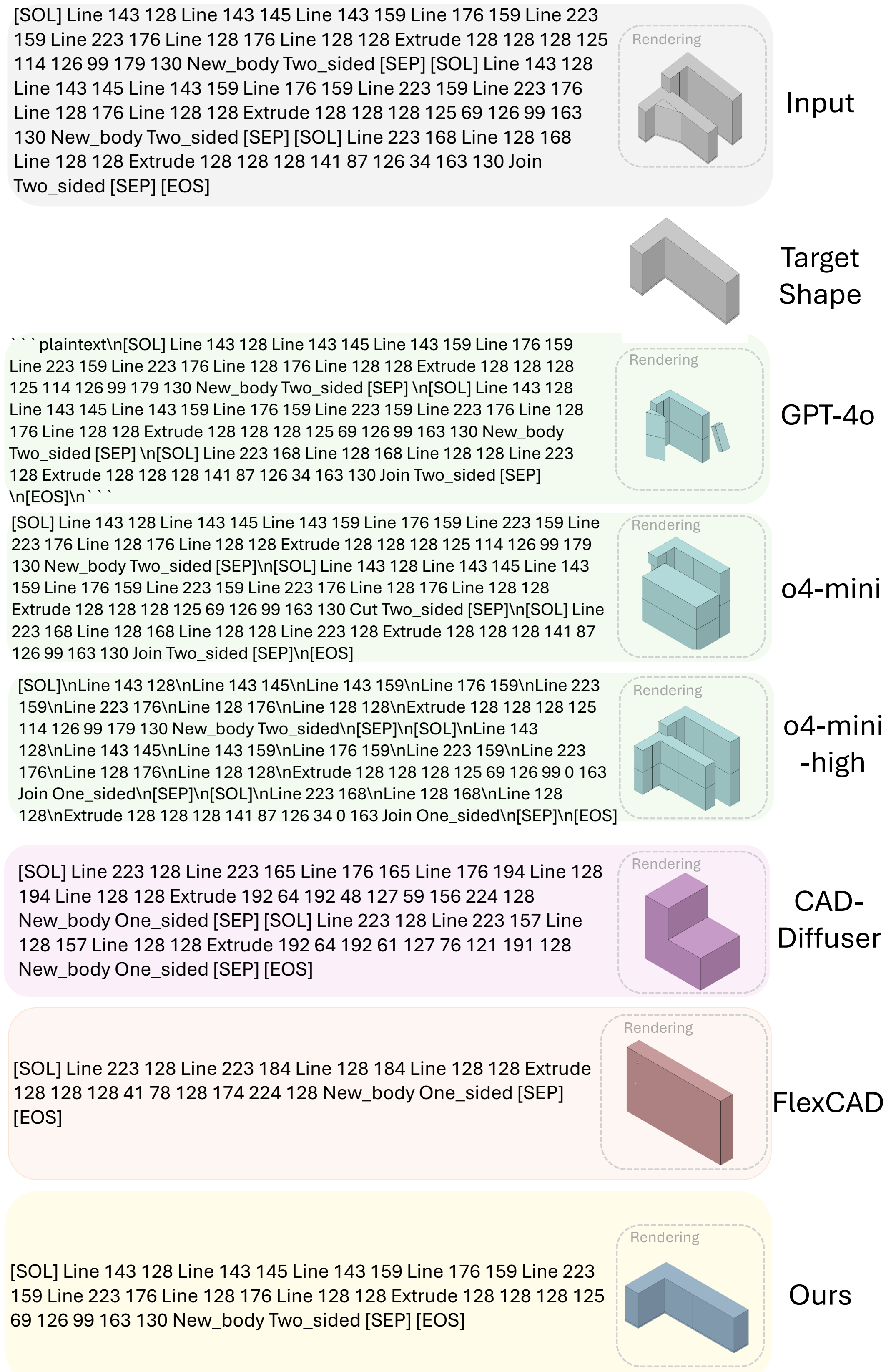


Figure 21: Full results of the original parametric sequence (with renderings), target shapes, and the edited sequence of different methods (with renderings). Blank represents not able to generate any sequence or the sequence is not able to render shapes after removing irrelevant characters. (17 of 24)



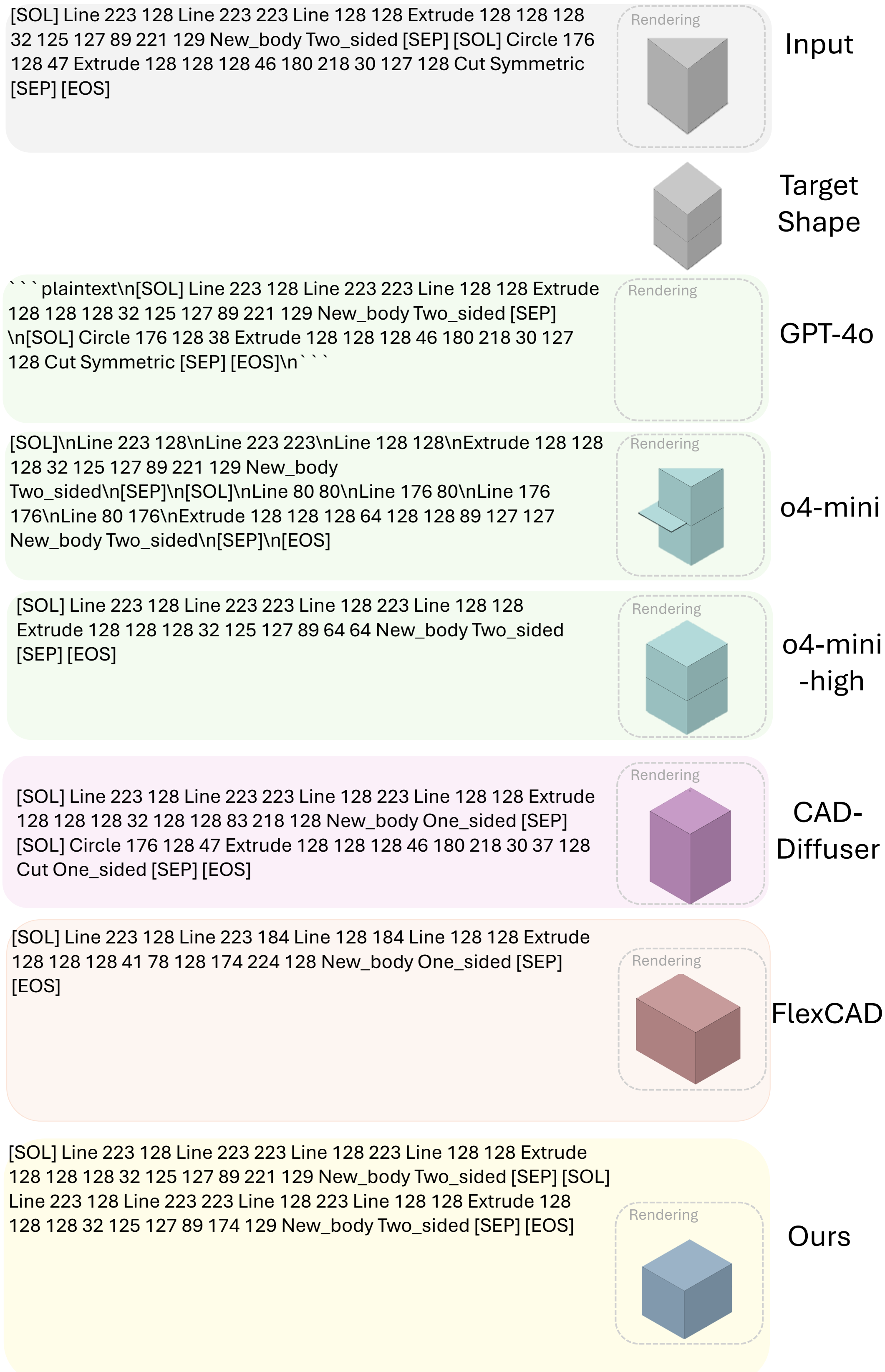
(18)

Figure 22: Full results of the original parametric sequence (with renderings), target shapes, and the edited sequence of different methods (with renderings). Blank represents not able to generate any sequence or the sequence is not able to render shapes after removing irrelevant characters. (18 of 24)



(19)

Figure 23: Full results of the original parametric sequence (with renderings), target shapes, and the edited sequence of different methods (with renderings). Blank represents not able to generate any sequence or the sequence is not able to render shapes after removing irrelevant characters. (19 of 24)

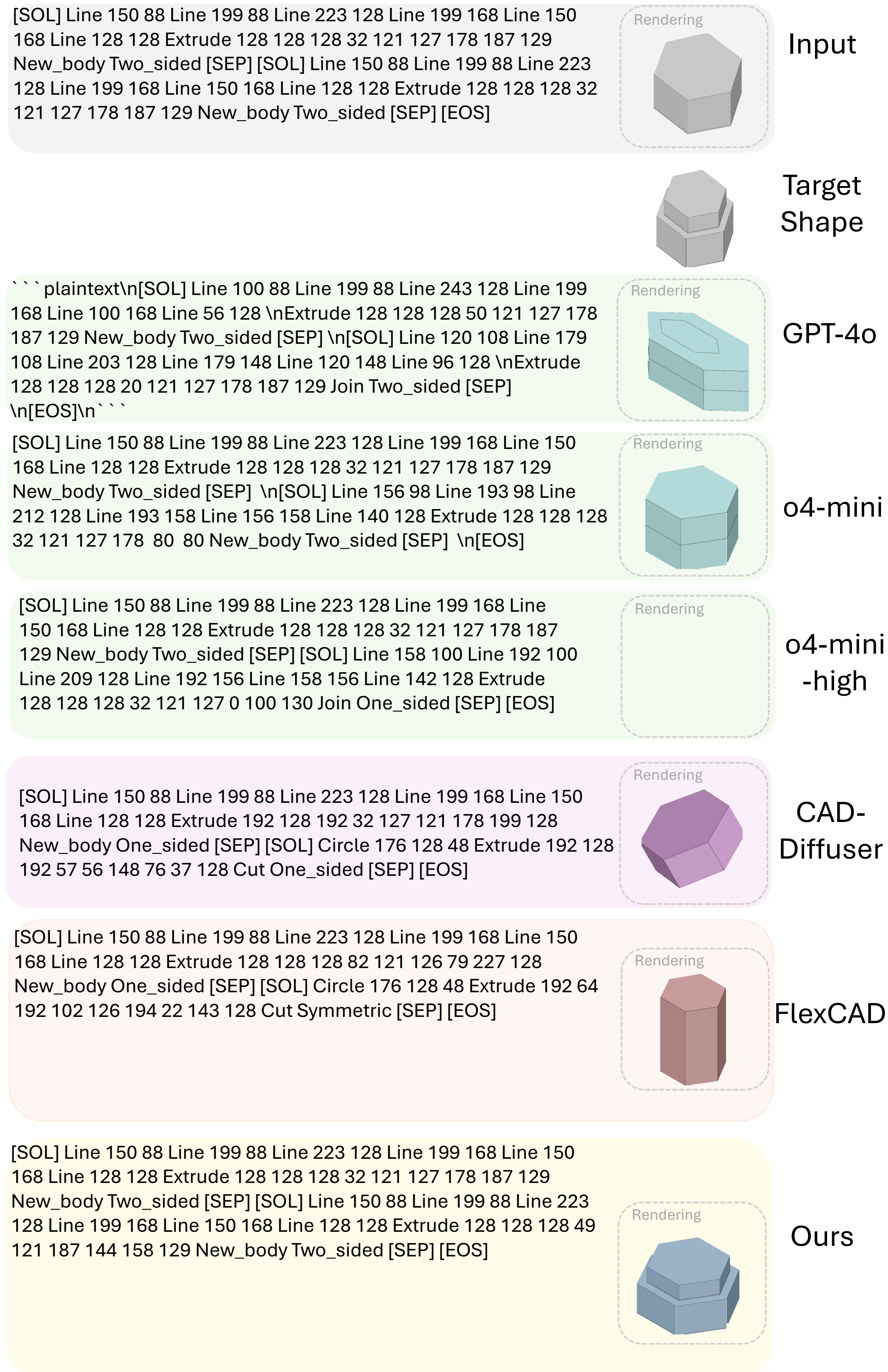


(20)

Figure 24: Full results of the original parametric sequence (with renderings), target shapes, and the edited sequence of different methods (with renderings). Blank represents not able to generate any sequence or the sequence is not able to render shapes after removing irrelevant characters. (20 of 24)

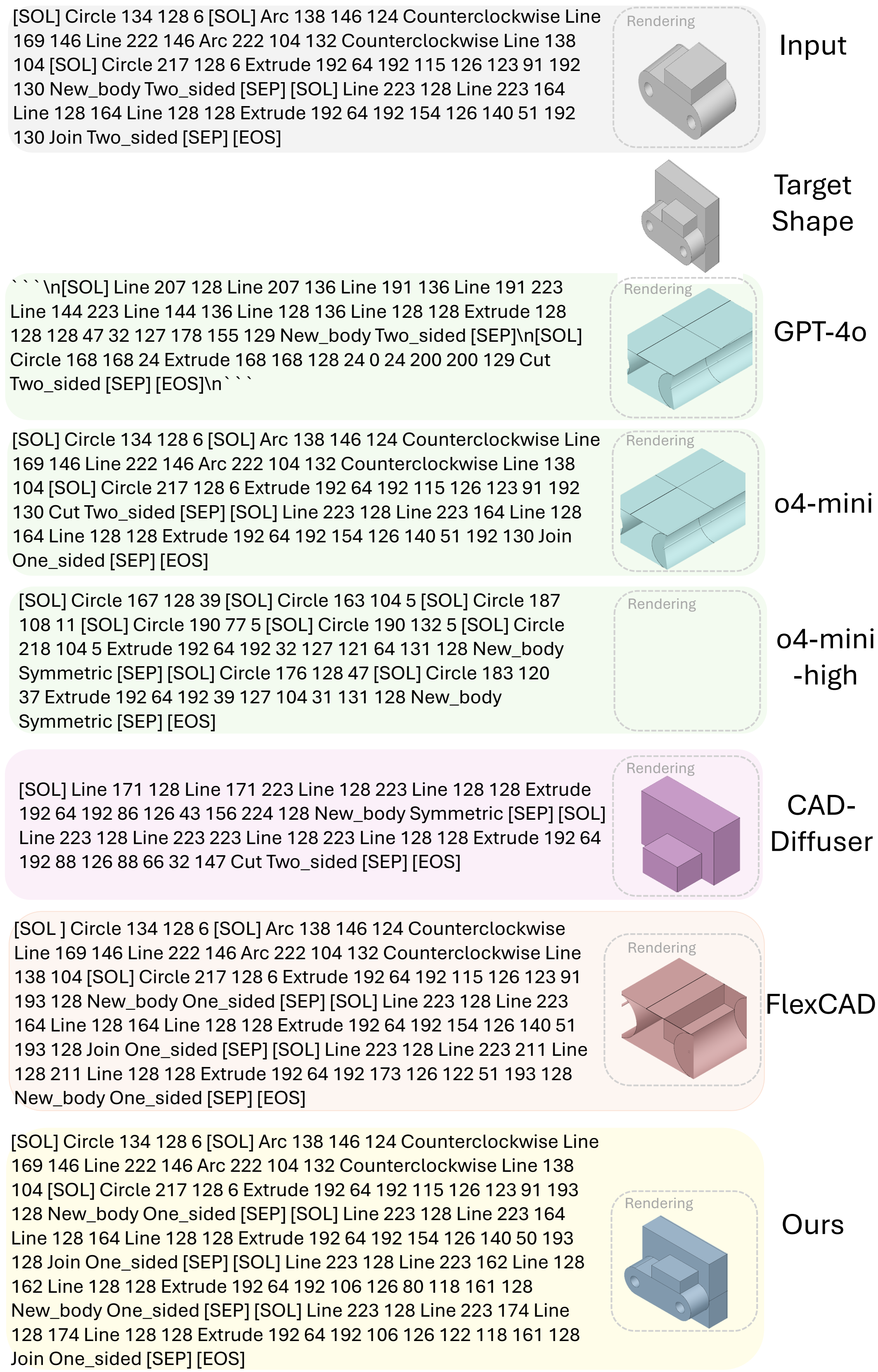
(21)

Figure 25: Full results of the original parametric sequence (with renderings), target shapes, and the edited sequence of different methods (with renderings). Blank represents not able to generate any sequence or the sequence is not able to render shapes after removing irrelevant characters. (21 of 24)



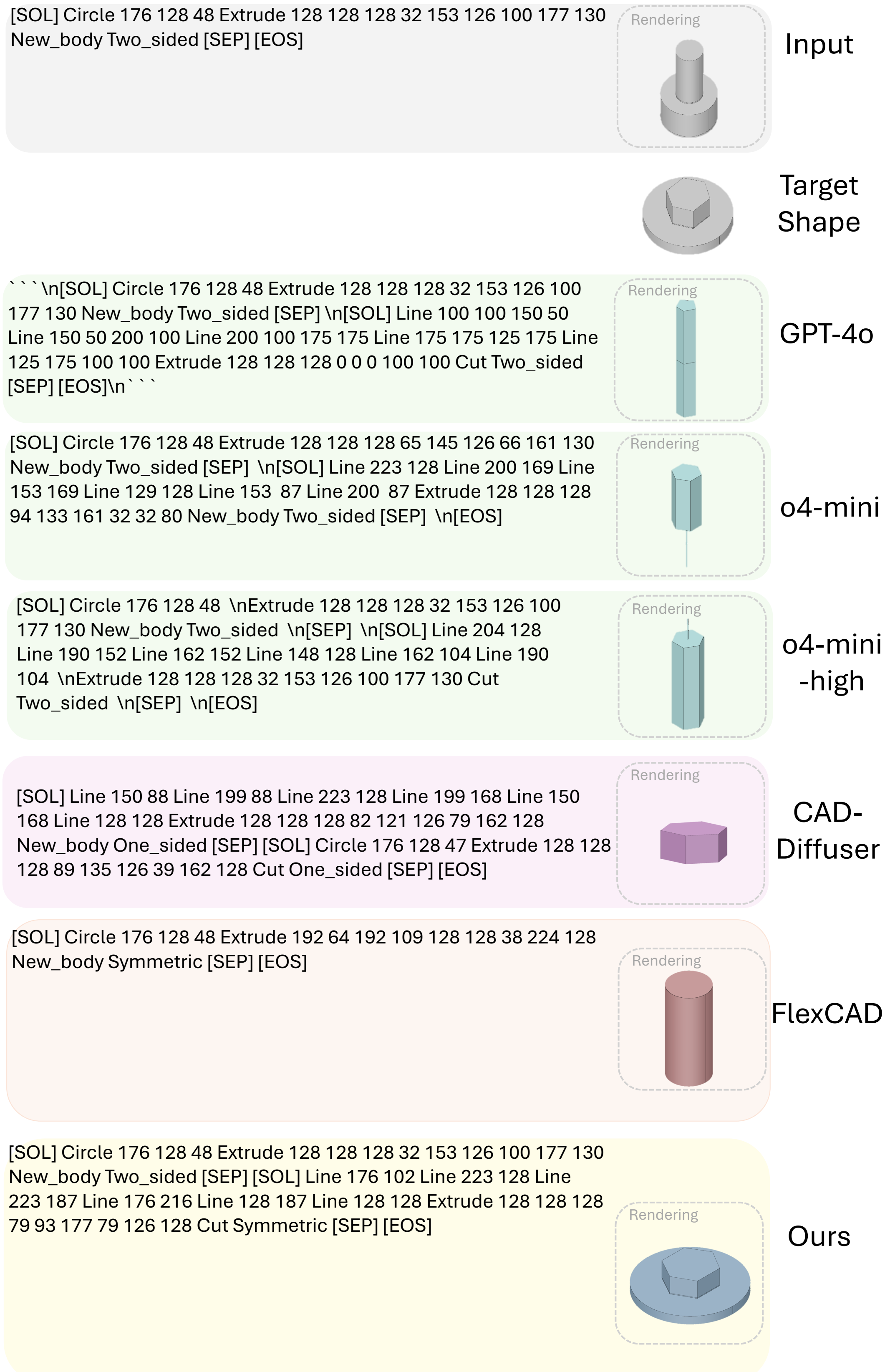
(22)

Figure 26: Full results of the original parametric sequence (with renderings), target shapes, and the edited sequence of different methods (with renderings). Blank represents not able to generate any sequence or the sequence is not able to render shapes after removing irrelevant characters. (22 of 24)



(23)

Figure 27: Full results of the original parametric sequence (with renderings), target shapes, and the edited sequence of different methods (with renderings). Blank represents not able to generate any sequence or the sequence is not able to render shapes after removing irrelevant characters. (23 of 24)



(24)

Figure 28: Full results of the original parametric sequence (with renderings), target shapes, and the edited sequence of different methods (with renderings). Blank represents not able to generate any sequence or the sequence is not able to render shapes after removing irrelevant characters. (24 of 24)

References

Jiemin Fang, Junjie Wang, Xiaopeng Zhang, Lingxi Xie, and Qi Tian. Gaussianeditor: Editing 3d gaussians delicately with text instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. doi: 10.48550/arXiv.2311.16037. arXiv:2311.16037.

Ayaan Haque, Matthew Tancik, Alexei A. Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. doi: 10.48550/arXiv.2303.12789. arXiv:2303.12789.

Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3D: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 300–309, 2023. doi: 10.48550/arXiv.2211.10440.

Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9298–9309, October 2023.

Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. doi: 10.48550/arXiv.2209.14988.

Jingxiang Sun, Cheng Peng, Ruizhi Shao, Yuan-Chen Guo, Xiaochen Zhao, Yangguang Li, Yanpei Cao, Bo Zhang, and Yebin Liu. Dreamcraft3d++: Efficient hierarchical 3d generation with multi-plane reconstruction model. *arXiv preprint arXiv:2410.12928*, 2024. doi: 10.48550/arXiv.2410.12928.

Jing Wu, Jia-Wang Bian, Xinghui Li, Guangrun Wang, Ian Reid, Philip Torr, and Victor Adrian Prisacariu. Gaussctrl: Multi-view consistent text-driven 3d gaussian splatting editing. In *European Conference on Computer Vision (ECCV)*, 2024. doi: 10.48550/arXiv.2403.08733. arXiv:2403.08733.

Taoran Yi, Jiemin Fang, Junjie Wang, Guanjun Wu, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Qi Tian, and Xinggang Wang. Gaussiandreamer: Fast generation from text to 3d gaussians by bridging 2d and 3d diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. doi: 10.48550/arXiv.2310.08529. arXiv:2310.08529.