
LLM-driven Indoor Scene Layout Generation via Scaled Human-aligned Data Synthesis and Multi-Stage Preference Optimization Supplementary Material

Anonymous Author(s)

Affiliation

Address

email

1 A More details of 3D-SynthPlace dataset

2 A.1 3D scene layout alignment

3 In the Holodeck [3] generation data, (0, 0) is generally used as the bottom-left corner of the floor,
4 while the single room information in 3D-Front is extracted from a large room formed by a collection
5 of multiple rooms, making the floor center more uncertain. To align the centers of the rooms in both
6 datasets, we agree in the prompt to use (0, 0) as the floor center. We calculate the geometric center of
7 the floor based on the floor boundary coordinates of both datasets and translate the floor boundary
8 and the coordinates of objects within the room accordingly. The offset is the difference between (0,
9 0) and the original geometric center of the floor. Additionally, we find that although both Holodeck
10 and 3D-Front use a y-axis up coordinate system, their xz-axis rotation directions are opposite. To
11 unify the rotation directions, we standardize to the 3D-Front rotation direction by adding 180° to the
12 rotation angle of objects in the Holodeck, thus making the rotation angles consistent.

13 A.2 The deleted Holodeck generated data

14 As we discussed in Section 3.2, the 3D scene layout rooms which generated by the Holodeck have
15 the low success rate. We have to filter the data and need to delete the bad generated data based on the
16 three problems: (1) under-populated layouts, (2) clustered object placements in large rooms, and (3)
17 erroneous object counts or orientations.

18 **Under-populated Layouts:** These layouts have a noticeably insufficient number of objects, making
19 the space appear sparse and unnatural. Particularly in bathrooms and bedrooms, the lack of sufficient
20 furniture and decorations results in a lack of liveliness and functionality. Such layouts may not
21 effectively simulate real-life environments, impacting the model’s training effectiveness.

22 **Clustered Object Placements in Large Rooms:** In larger rooms, objects are placed too closely
23 together, leading to uneven space utilization. This can cause visual disharmony and fail to realistically
24 reflect the reasonable distribution of furniture in real life. This issue is especially pronounced in
25 living rooms and kitchens, affecting both functionality and aesthetics.

26 **Erroneous Object Counts or Orientations:** Errors in the number or orientation of objects lead
27 to unreasonable room layouts. For example, the orientation of furniture may not match the room’s
28 structure, or the number of objects may be too many or too few, affecting the overall harmony of
29 the room. Such errors can mislead the model in learning spatial relationships, impacting reasoning
30 capabilities.

31 We also illustrate some deleted layouts to show the error of the room scenes in Fig. 1. In addition
32 to filtering out invalid layouts, we exclude small objects produced during the Holodeck generation
33 process, as they are not compatible with the problem definition of our room layout formulation.



Figure 1: The deleted error data which generated by Holodeck.

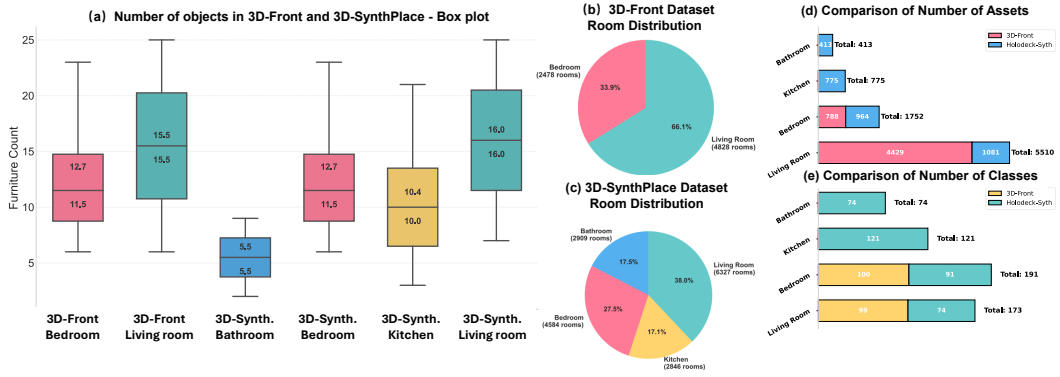


Figure 2: Comparison of object counts, room type distributions, and asset/class statistics between the 3D-Front and 3D-SynthPlace datasets.

34 A.3 Data distribution description

35 The Fig. 2 presents a comprehensive distribution comparison between the 3D-Front and 3D-
36 SynthPlace datasets across various dimensions:

37 **Number of Objects Fig. 2(a):** The box plot illustrates the distribution of furniture counts in different
38 room types for both datasets. The 3D-Front dataset shows a higher median count in living rooms
39 compared to bedrooms, while the 3D-SynthPlace dataset exhibits a similar pattern with slightly
40 higher counts in living rooms and kitchens.

41 **Room Distribution Fig. 2(b & c):** Pie charts depict the room distribution within each dataset. The
42 3D-Front dataset is predominantly composed of living rooms (66.1%), followed by bedrooms (33.9%).
43 In contrast, the 3D-SynthPlace dataset has a more diverse distribution, with living rooms (38%) and
44 bedrooms (27.5%) being the most common, alongside significant portions of bathrooms (17.5%) and
45 kitchens (17.1%).

46 **Comparison of Number of Assets Fig. 2(d):** A bar chart compares the total number of assets
47 across different room types. The 3D-Front dataset has a significantly higher total number of assets,
48 especially in living rooms, compared to the Holodeck-Synth dataset.

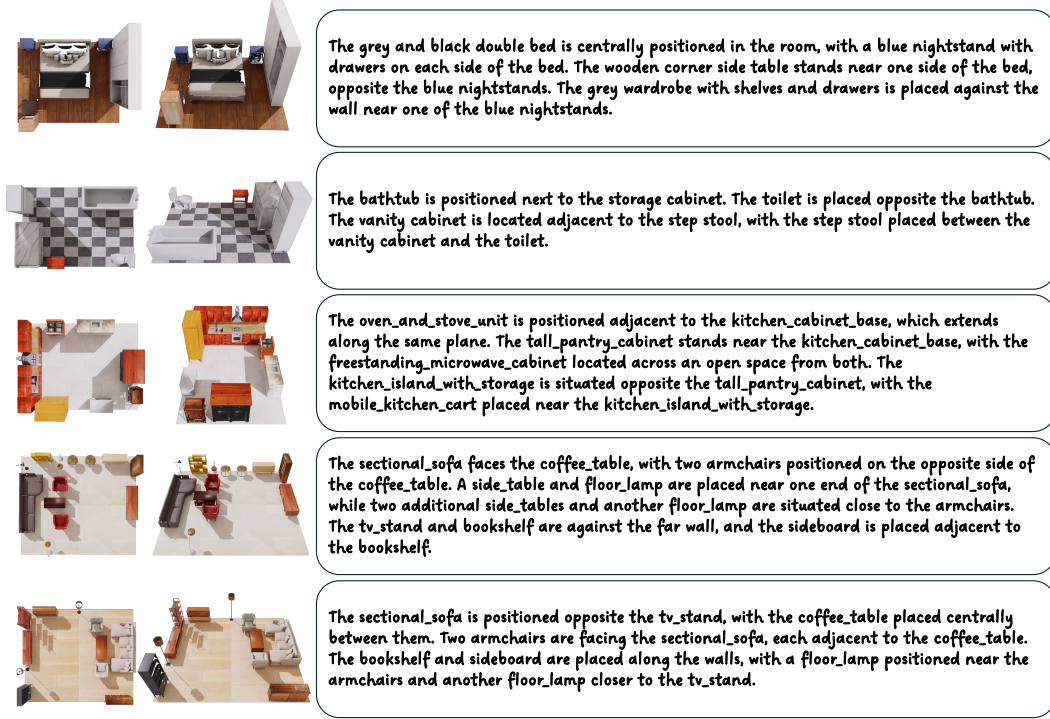


Figure 3: Reasoning high-level semantic description with the 3D scene layouts.

49 **Comparison of Number of Classes Fig. 2(e):** Another bar chart compares the number of classes
50 available in each dataset. The 3D-Front dataset generally has more classes across different room
51 types, with the most notable difference in the living room category.

52 A.4 The reasoning description illustration

53 In Fig. 4, we show some reasoning high-level semantic descriptions with the 3D scene layouts.
54 Examples of generated room layouts paired with natural language descriptions. Each row shows two
55 3D layout visualizations of the same room from different angles, accompanied by a GPT-generated
56 description of the spatial arrangement and object relations. The descriptions emphasize functionality,
57 object alignment, and spatial reasoning without relying on directional terms like "left" or "right".

58 B More details of SFT

59 We discussed how to do the SFT in the Section 3.3. And in this part, we would like to talk about the
60 details of the meta instruction prompt. First, we illustrate the prompt template in the Table 1 and Table
61 2. As shown in the tables, to fine-tune the model for the 3D room layout generation task, we design a
62 structured meta prompt that sets the model as a skilled room layout designer. The prompt guides the
63 model through a step-by-step reasoning and generation process, covering object extraction, spatial
64 analysis, layout planning, and final output formatting. It explicitly incorporates design heuristics
65 such as edge-aligned placement, alignment to walls, and functional constraints (e.g., chairs must face
66 desks). The model is instructed to reason about object relationships without using explicit directions
67 (e.g., "left", "right") and to output both the reasoning process and the final layout in a well-defined
68 JSON format. The response is enclosed in a structured JSON-like format for better consistency and
69 parsing. The prompt also includes post-checklists to ensure validity of the generated layout (e.g.,
70 no overlap, correct coordinate units, functional flow). This design ensures that the model not only
71 generates spatially valid scenes but also explains its decisions in a transparent and interpretable way.

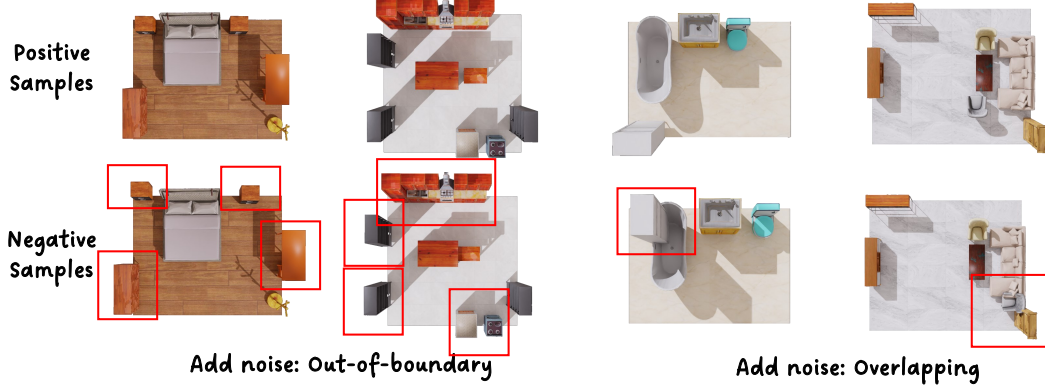


Figure 4: Some positive scene layouts and negative scene layouts with noise additions in the DPO-stage2.

72 C More details of DPO

73 In Section 3.4, we introduce a two-stage DPO (Direct Preference Optimization) framework designed
 74 to align generated room layouts with human preferences. We begin by selecting 1,100 scene layouts
 75 from the Holodeck-Synth dataset and another 1,100 from the 3D-Front dataset. These layouts are
 76 characterized by reasonable, functional, and human-aligned spatial arrangements. In the second
 77 stage of DPO training (DPO-stage2), we generate hard negative samples by adding targeted spatial
 78 violations to the positive examples. Specifically, as illustrated in the , we introduce two types of
 79 noise: (1) Out-of-boundary violations, where furniture items extend beyond the floor area; (2) Object-
 80 overlap violations, where furniture pieces intersect or collide with each other. Despite improvements
 81 from SFT and DPO-stage1, the model still occasionally produces such errors. Therefore, we further
 82 fine-tune the model using these challenging negative samples to enhance spatial reasoning and layout
 83 robustness.

84 D More details of experiments

85 D.1 Inference and rendering setup.

86 All layout generation results are produced using the optimized model obtained after the second-
 87 stage Direct Preference Optimization (DPO). To ensure physically plausible and visually accurate
 88 visualization of scenes, we employ **NVIDIA Isaac Sim** [2] as the rendering and simulation backend.

89 D.2 Details of the evaluation metrics

90 **Object overlap rate (OOR).** The Object Overlap Rate (OOR) quantifies the spatial overlap between
 91 a set of 2D bounding boxes. To calculate the OOR, first extract the position and size information
 92 from each object, where the position is typically represented as (x, y) and the size is given by width
 93 and depth. Using this information, create 2D bounding boxes with the center at the object’s position
 94 and dimensions determined by the size. Compute the area of each bounding box and, for each pair of
 95 bounding boxes, calculate the area of their intersection. Sum the intersection areas of all object pairs
 96 to obtain the total intersection area, and sum the areas of all objects to get the total area. The OOR is
 97 then defined as the total intersection area divided by the total area, as given by the formula:

$$\text{OOR} = \frac{\sum_{i=1}^N \sum_{j=i+1}^N \text{Area}(\text{Intersection}(B_i, B_j))}{\sum_{i=1}^N \text{Area}(B_i)}$$

98 where B_i represents the bounding box of the i -th object, $\text{Intersection}(B_i, B_j)$ denotes the intersection
 99 area of two bounding boxes, and $\text{Area}(B_i)$ is the area of the bounding box.

100 **GPT-4o evaluation.** Following a similar approach to the evaluation protocol introduced in I-
101 Design [1], we employ GPT-4o as an automated evaluator to assess the quality of our generated room
102 layouts. Specifically, GPT-4o is prompted to assign scores (ranging from 0 to 10) based on multiple
103 criteria, including functionality, spatial arrangement, and aesthetic coherence, in alignment with the
104 given user preferences. The detailed prompt template used for this evaluation is provided in Table 3.

105 **D.3 More qualitative results.**

106 Fig. 5 and Fig. 6 present additional qualitative comparisons between the outputs of our OptiScene
107 model (left two columns) and the ground truth (right two columns) across various room configurations.
108 It is important to emphasize that the objective of OptiScene is not to converge to the ground truth
109 layouts exactly, but rather to generate plausible and functional indoor scenes based on high-level
110 semantic and spatial constraints. The generated layouts exhibit a even more efficient use of space,
111 better object alignment, and clearer functional zoning compared to the ground truth. These examples
112 demonstrate the model’s capacity to generalize and produce reasonable, sometimes even preferable,
113 alternative layouts that remain faithful to the intended room semantics. This highlights the potential
114 of OptiScene to support diverse and creative room layout configurations.

115 **D.4 More downstream tasks results.**

116 In Fig. 7 and Fig. 8, we illustrate more downstream tasks results with room editing and robot
117 navigation. As shown in figures, with appropriate instruction, the model has the good capability to
118 edit the room layout and generate the robot navigation path.

119 **E Limitation and future work**

120 In our paper, we introduced the 3D-SynthPlace dataset and the OptiScene algorithm. The former
121 significantly expands the previously insufficient 3D scene layout datasets, while the latter is a human-
122 aligned indoor layout generation framework based on a fine-tuned open-source LLM. However, our
123 work still has some limitations that we hope to improve in future work: 1. We only considered the
124 generation of a single room layout, without considering the generation of multiple room layouts. We
125 will consider how to generate the layout of an entire house or building in the future. 2. Currently,
126 we only created a scene dataset based on furniture, and we will collect more small objects and wall
127 objects data in the future to build a more complete and detailed room layout, further enhancing the
128 authenticity of the room.

Table 1: Meta Prompt Template for generation task. (Part 1.)

Meta Prompt Template for generation task (Part 1.)

You are a skilled room layout designer. Your task is to arrange [Objects] within a given [Room Type] effectively. Follow these guidance to complete your design:

(1) Extract the [Room Type], [Room Area], [Objects], and [Bounding Box Size] from the provided JSON data. (2) Analyze the spatial relationships among [Objects] within the specified [Room Type]. Pay special attention to ****avoiding overlap**** and ****consider other spatial factors like accessibility and aesthetics****.

(3) Determine and design the precise location of all [Objects] ensuring that their bounding boxes do not overlap and that the layout is functional and visually appealing.

(4) I prefer objects to be placed at the edge (the most important constraint) of the room if possible which makes the room look more spacious.

(5) The objects are usually aligned in some ways (parallel or perpendicular to walls).

(6) Chairs must be placed near to the table/desk and face to the table/desk.

(7) Before specifying the detailed positions of each object, first reason step-by-step about their general arrangement and relative spatial relationships: a) Which objects need the most space or have fixed positions (like beds, wardrobes) b) Which objects need to be grouped together (like nightstands with bed) c) Traffic flow and accessibility considerations. Then, clearly articulate your reasoning process. Emphasize the spatial relationships between objects without using explicit directional terms like "left," "right," "front," or "back." Summarize the overall arrangement in a logical and natural manner, ensuring that all major objects are accounted for.

(8) After presenting the thought process, report your design with detailed 3D space coordinates and rotation angles for each object in JSON format, as follows:

```
"object": "object",
"coordinates": [
  {
    "x": x,
    "y": y,
    "z": z
  }
],
"rotate":[
  {
    "angle": r
  }
]
```

The centroid of the room is {"x": 0.00, "y": 0.00, "z": 0.00}.

Important Notes about Coordinate System:

- Z-axis points upward (z=0 is floor level)
 - Rotation angles are in radians, measured in the XY-plane
-

Table 2: Meta Prompt Template for generation task. (Part 2.)

Meta Prompt Template for generation task (Part 2.)

(9) The response should follow the following format:

```
<reasoning>
[Reason]
...
[/Reason]
</reasoning>
<answer>
[Design]
...
[/Design]
</answer>
```

First carefully read this example:

```
[Example Room Type]
Bedroom
[/Example Room Type]
```

```
[Example Objects and Bounding Box Size]
/* A fixed example is put here to show the input format*/
[/Example Objects and Bounding Box Size]
```

```
[Example Reason]
/* A fixed example is put here to show the reason format*/
[/Example Reason]
```

```
[Example Output]
/* A fixed example is put here to show the output format*/
[/Example Output]
```

Now, please proceed with the design task as outlined and provide only the JSON formatted output of your design:

```
[Task Room Type]
/*Input room type*/
[/Task Room Type]
```

```
[Task Objects & Bounding Box Size]
/* The JSON format input of objects description
and bounding box size*/
[/Task Objects & Bounding Box Size]
```

Note: the units for the coordinates are meters.

Before submitting your final design, please verify:

- All objects are within room boundaries
- No objects overlap
- Sufficient clearance space exists around furniture
- The layout is practical and functional
- All rotations are properly specified in radians

Now, please proceed with the design task as outlined and provide your thought process and the JSON formatted output of your design:

Table 3: GPT-4o Prompt Template for Room Layout Evaluation.

GPT-4o Prompt Template for Room Layout Evaluation

You are an expert in interior design and human-centric spatial planning. Your task is to evaluate the quality of the following room layout renders based on how well they match the user’s design preferences, which are provided below (in triple backquotes).

Please assign a numerical score from ****0 to 10**** (0 = completely inconsistent, 10 = perfectly aligned) considering the following three aspects:

1. Functionality and Activity-based Alignment

- Does the layout support natural and efficient use of the space for daily activities (e.g., sleeping, working, relaxing, walking)?
- Are key object groupings (e.g., desk and chair, bed and nightstand) placed functionally and accessibly?
- Is there sufficient circulation space to ensure human accessibility?

2. Layout and Furniture Placement

- Are the furniture pieces arranged in a logical, practical way within the room boundaries?
- Are objects positioned with respect to common design principles (e.g., not blocking windows or doors, aligned to walls where appropriate)?
- Are there overlaps or unnatural collisions between objects?

3. Aesthetic Coherence

- Is the overall layout visually balanced and spacious?
- Does the arrangement exhibit good proportions, symmetry or asymmetry, and grouping where needed?
- Is the furniture distribution harmonious and pleasing to the eye, according to the user’s stated aesthetic preferences?

User Preferences:

““{Insert user preferences here.}””

After considering all the above, return your evaluation in the following JSON format:

```
{
  ````{
 "functionality_score": X,
 "layout_score": Y,
 "aesthetics_score": Z,
 "overall_score": M,
 "comments": "Brief explanation of your judgment."
 }````
}
```

**Scoring Guidelines:**

- Scores should be integers from 0 to 10.
  - The overall score can be the average or holistic assessment across the three criteria.
  - Include a brief justification of your scores in the "comments" field (1–3 sentences).
- Note:** The goal is to measure how well the generated layout adheres to functional, spatial, and aesthetic expectations given the user’s input. Be fair and critical.
-



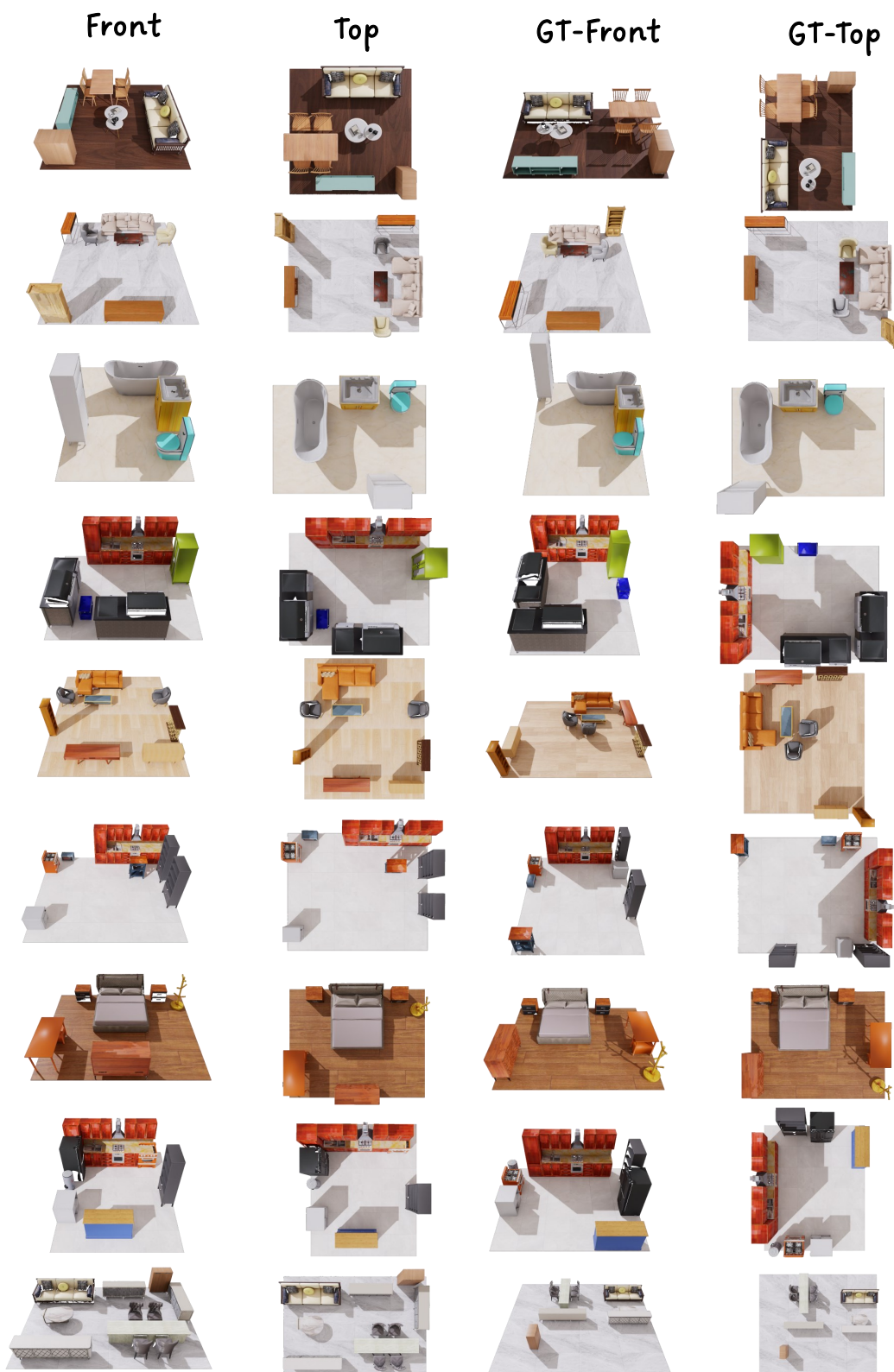


Figure 5: More qualitative results which are compared with the GT. (Part 1.)

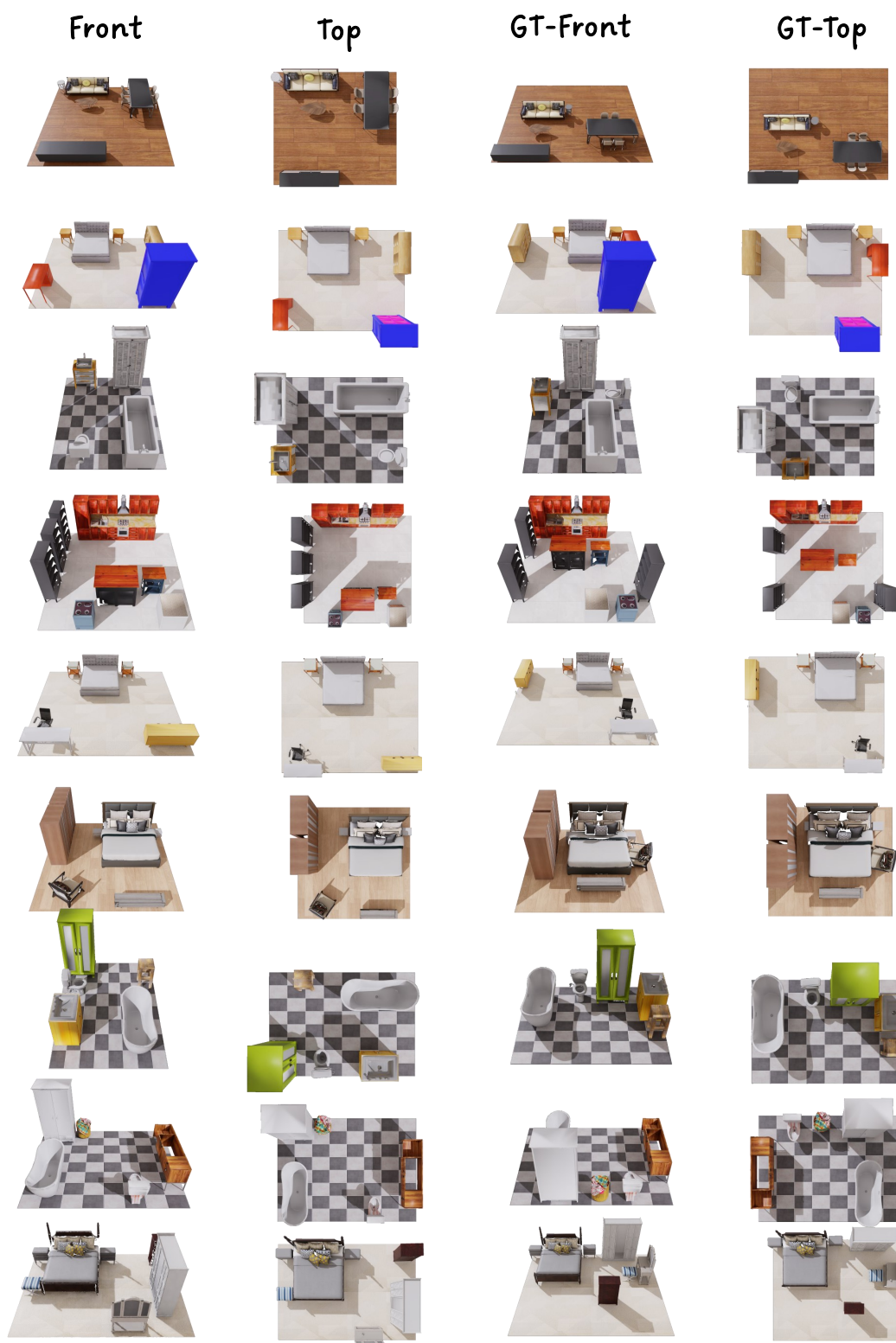


Figure 6: More qualitative results which are compared with the GT. (Part 2.)

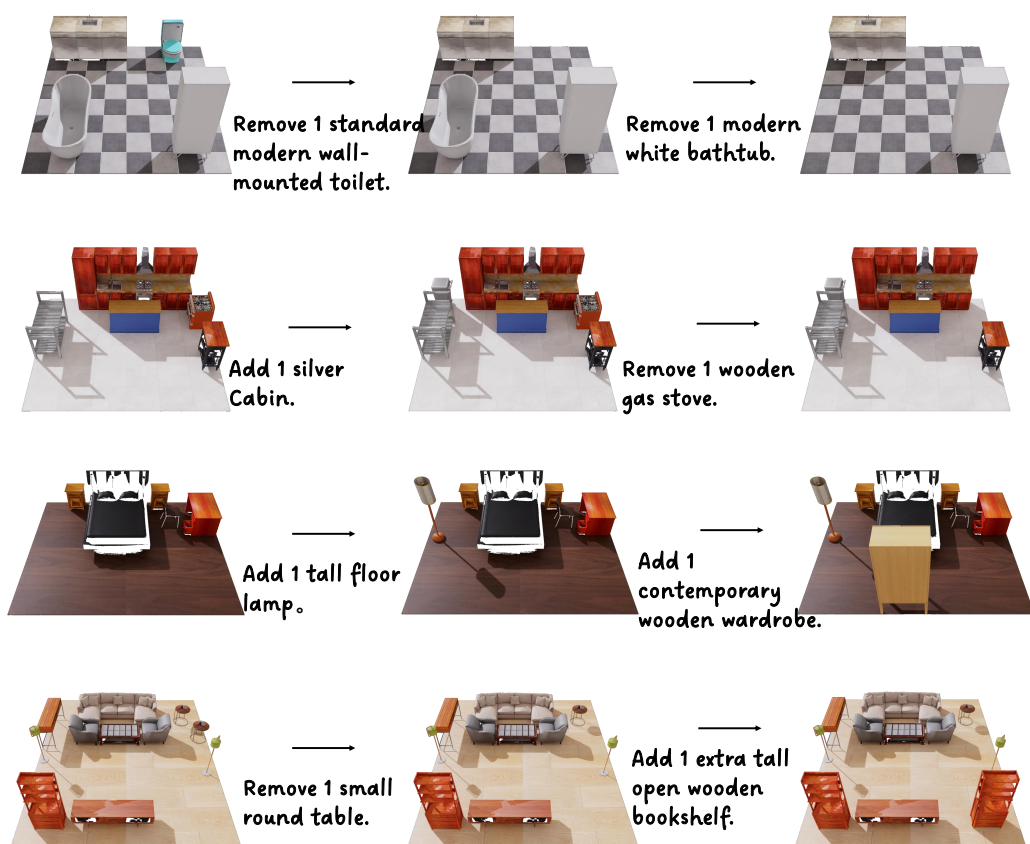


Figure 7: More editing results.



Figure 8: More instruction navigation results.

## References

- [1] Ata Çelen, Guo Han, Konrad Schindler, Luc Van Gool, Iro Armeni, Anton Obukhov, and Xi Wang. I-design: Personalized llm interior designer. *arXiv preprint arXiv:2404.02838*, 2024.
- [2] NVIDIA. Isaac sim 4.0 - robotics simulation and synthetic data generation. <https://developer.nvidia.com/isaac-sim>, 2024.
- [3] Yue Yang, Fan-Yun Sun, Luca Weihs, Eli VanderBilt, Alvaro Herrasti, Winson Han, Jiajun Wu, Nick Haber, Ranjay Krishna, Lingjie Liu, et al. Holodeck: Language guided generation of 3d embodied ai environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16227–16237, 2024.