HierRR: Enhancing Instruction Alignment in Open-Vocabulary Indoor Scene Synthesis via Agentic Hierarchical Reasoning and Reflection

Weilin Sun[1], **Lei Meng**[1,*], **Lihui Zhang**[1], **Manyi Li**[1], **Xiangxu Meng**[1]

[1]School of Software, Shandong University, China

sunweilin@mail.sdu.edu.cn, lmeng@sdu.edu.cn, 202535458@mail.sdu.edu.cn, manyili@sdu.edu.cn, mxx@sdu.edu.cn

Abstract

Open-vocabulary indoor scene synthesis aims to generate plausible layouts from arbitrary user instructions while ensuring physical feasibility and semantic consistency. Existing methods directly infer spatial relations between objects from instructions and solve for layouts based on predefined rules, achieving progress in physical feasibility. However, they often struggle to precisely align with complex instructions due to insufficient comprehension of spatial relations and limited capability to translate them into layouts. In this paper, we propose a Hierarchical Reasoning and Reflection agent framework (HierRR), which leverages a relational hierarchy to enable multi-granularity spatial relation comprehension and attributable layout reasoning, improving semantic consistency while maintaining physical feasibility. Specifically, HierRR introduces the spatial transformation chain-of-thought reasoning module that explicitly models the mapping from spatial relations to executable geometric placements, mitigating semantic drift during reasoning. Furthermore, the semantic- and vision-guided reflection module is employed to iteratively refine spatial relations and layout reasoning from local to global along the hierarchy, effectively bridging the gap between language instructions and geometric layouts. Extensive comparative experiments and ablation studies demonstrate that HierRR generates more plausible and semantically consistent indoor scene layouts than existing methods, while generalizing effectively to complex scenes with numerous objects and diverse spatial constraints.

1 Introduction

Text-to-3D indoor scene synthesis aims to generate object-centric 3D layouts from language instructions, enabling applications such as interior design, embodied simulation, and virtual content creation [1, 2, 3]. With the rise of large vision-language models (VLMs), recent works [4, 5, 6, 7, 8] have moved toward open-vocabulary scene synthesis, supporting free-form language instructions, unrestricted object categories, and diverse compositions. The key challenge lies in grounding instructions into layouts: instructions often specify spatial relations that are compositional and partially implicit. For example, "a table surrounded by chairs" presupposes roughly uniform spacing, center-facing orientations, and no collisions. Satisfying such relations simultaneously is non-trivial, as the generated layout must be both physically plausible and faithful to the instruction.

Existing open-vocabulary scene synthesis methods follow two main paradigms: direct numerical generation and parse-and-optimize generation. The former [9, 10, 11] combines spatial relation

*Corresponding author

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

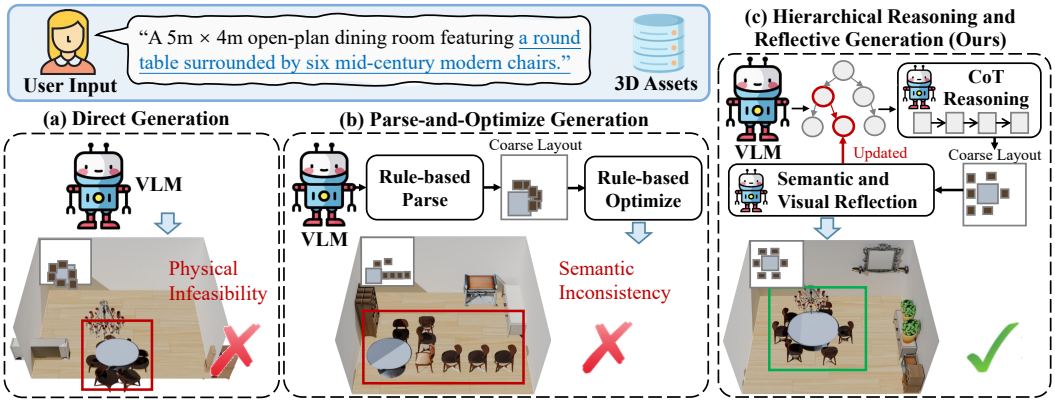

Figure 1: Compared to (a) direct generation and (b) parse-optimize methods, we introduce a hierarchical reasoning and reflection framework (c), which explicitly models layout reasoning based on the hierarchy and incorporates reflection from local to global to iteratively refine spatial relations, effectively mitigating inconsistency between instructions and layouts while ensuring physical feasibility.

understanding with coordinate prediction in a single stage, but it often yields physically implausible results due to the limited ability of VLMs in precise numerical reasoning (Figure 1(a)). The latter [4, 5, 8, 12] infers spatial relations to produce a coarse layout, then solves an optimization problem to improve physical feasibility (Figure 1(b)). However, it struggles to directly comprehend and infer detailed spatial relations when faced with complex, compositional instructions, and also suffers from semantic drift between the intended relations and realized placements.

Inspired by recent advances in VLM-based agentic reasoning and self-reflection, a natural alternative is to formulate layout synthesis as an iterative propose-evaluate-refine process. However, existing methods [13, 14, 15] are not fully suited to scene layout, as they focus on global evaluation and adjustment while lacking awareness of local spatial relations, making it difficult to precisely localize layout defects. For instance, upon detecting that the left side of a room is overly crowded, they may broadly suggest "spread out the objects on the left", while overlooking local coupled relations such as "the chairs should be placed adjacent to the table", leading to erroneous corrections. Such lack of local awareness often requires more iterations to converge, resulting in inefficient feedback loops.

To address the aforementioned issues, we propose HierRR, a Hierarchical Reasoning and Reflection agent framework for open-vocabulary indoor scene synthesis. The key idea is to enable multi-granularity spatial relation comprehension and attributable grounding into layouts by constructing a relational hierarchy to guide iterative reasoning and reflection. As shown in Figure 2, HierRR consists of three modules: 1) the hierarchy generation module organizes spatial relations top-down and decomposes layout generation into locally optimized sub-tasks under global guidance; 2) the spatial transformation chain-of-thought reasoning module explicitly models the mapping from relational constraints to precise geometric placements, mitigating semantic drift while integrating physical constraints to ensure feasibility; 3) the semantic- and vision-guided reflection module enriches spatial relations through fine-grained semantic refinement, while incorporating visual reflection to identify layout issues and generate revision suggestions. The relational hierarchy serves as a shared structure across all three modules. Through bottom-up traversal, reasoning and reflection perceive multi-granularity spatial relations from local to global, while each step focuses on relevant context, thereby avoiding confusion from information overload and enabling efficient layout generation. We conducted extensive comparative experiments and ablation studies, including qualitative and quantitative evaluations. Results show that our framework achieves state-of-the-art performance in generating physically feasible, semantically consistent, and visually realistic layouts, while requiring fewer reflection iterations than prior agent-based methods. Our contributions are summarized below:

- We propose HierRR, a novel hierarchical reasoning and reflection agent framework for open-vocabulary indoor scene synthesis, which enables multi-granularity spatial relation comprehension and attributable grounding into layouts by constructing a relational hierarchical structure.
- We design a spatial transformation chain-of-thought reasoning module that operationalizes relational constraints into executable geometric placements, effectively mitigating semantic drift.

- We develop a semantic- and vision-guided reflection module that pinpoints layout issues and refines spatial relations along the hierarchy, thereby improving reflection efficiency and layout quality.

2 Related Work

Indoor Scene Synthesis. The task aims to generate objects with semantic labels and positions (i.e., layouts). Early work relied on predefined rules [16, 17, 18] or data-driven approaches [19, 20, 21, 22]. With the rise of Large Language Models (LLMs), open-vocabulary scene synthesis became possible; for example, LayoutGPT [9] directly outputs layout coordinates. However, LLMs lack numerical reasoning ability, leading to severe collisions and out-of-bound issues. To mitigate this, subsequent methods shifted to multi-stage parsing and optimization [5, 8]. Holodeck [4] uses an LLM to generate a scene graph describing object relations, then applies rule-based optimization to determine positions. Nevertheless, the decoupling between semantic relations and position solving leads to poor semantic consistency. LayoutVLM [7] employs floor markers to assist a fine-tuned VLM in determining object placements, improving semantic consistency but limiting its ability to handle complex spatial relations. Furthermore, some studies [23, 24] extract layouts rely on text-to-image, object detection, and depth estimation models to improve visual realism, yet the limited viewpoints pose challenges for object recognition and position estimation, making it difficult to align the results with user intent.

Vision-Language Model for Spatial Reasoning. Spatial reasoning is defined as the ability to understand and infer the relations among objects. Recent research explores the spatial reasoning capabilities of VLMs. Some studies [25, 26, 27] enhance VLM spatial reasoning by fine-tuning on collected spatial question-answering datasets. However, this approach limits the models' generalization ability when tackling more challenging, out-of-distribution tasks, especially those requiring multi-step reasoning. To overcome this limitation, Chain-of-Thought (CoT) [28] and its extensions [29, 30] have become key techniques for LLMs or VLMs. CoT prompts the model to decompose problems and generate a sequence of logical steps instead of providing a direct answer, effectively solving multi-step spatial reasoning problems. SpatialCoT [31] utilizes the VLM to generate multi-step spatial reasoning chains for visual understanding tasks, combining coordinate alignment with the reasoning path to boost performance in complex scenes. Inspired by this, we decompose the scene hierarchically and treat the determination of each object's coordinates and orientation as a multi-step reasoning process, using CoT to guide VLMs to determine placements in an attributable manner.

LLM-based Agent Framework. In this framework, the LLMs/VLMs serve as the brain, demonstrating powerful reasoning and planning capabilities through task decomposition [32, 33]. Furthermore, by learning from feedback, LLMs/VLMs can autonomously accomplish complex tasks [32, 33, 34, 35]. Recently, this concept has been adopted in 3D scene synthesis, where the generation process is decomposed into a multi-step pipeline coordinated by multiple LLMs [13, 36, 14, 15, 37, 38, 39]. However, existing works often rely on predefined rule sets and focus on global reflection while neglecting local spatial relations, resulting in inefficient feedback. In contrast, we construct a hierarchy to guide reasoning and reflection, thereby enhancing the understanding of multi-granularity spatial relations. Then, we design the spatial transformation chain-of-thought reasoning with semantic- and vision-guided reflection, which explicitly models the reasoning process and precisely localizes layout issues for refinement, achieving high-quality layout generation.

3 Problem Formulation

Problem. Given a textual user instruction x, our goal is to generate a scene comprising the selected 3D object models and their placements, i.e., $S = \{(D_i, \mathcal{P}_i)|i = 1, ..., n\}$, where D_i and $\mathcal{P}_i$ denote the selected 3D model and spatial placement for the i-th object, respectively. The placement of each object contains its center position coordinates and orientation.

Hierarchical Scene Representation. We propose to use a multi-level hierarchy as the scene representation, as shown in Figure 2 (left). The root node represents the entire scene, intermediate nodes denote rectangular functional areas, and leaf nodes denote the objects within each area. Nodes are connected by two types of edges: parent-child relations and pairwise spatial relations between nodes under the same parent.

Each node in the hierarchy contains several attributes. Assuming the scene's floor plan is an axis-aligned rectangle, the root node is defined as $o^r = \{x^r, s^r\}$, where s^r is the size attribute and x^r is

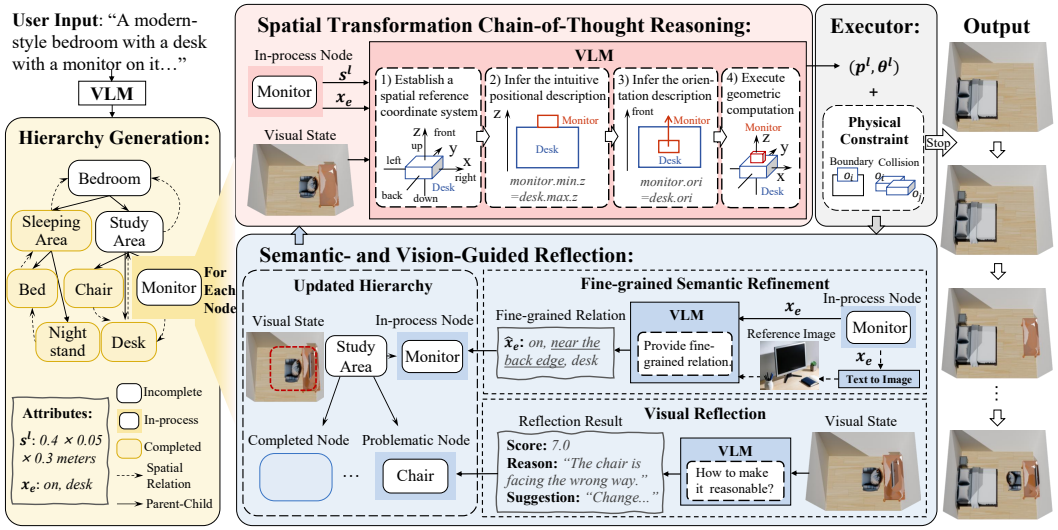

Figure 2: The pipeline of HierRR. First, we construct a hierarchy to organize spatial relations. Second, through bottom-up traversal, the spatial transformation chain-of-thought reasoning module explicitly infers executable placements from spatial relations. Third, the semantic- and vision-guided reflection module refines spatial relations and optimizes the layout reasoning process. Benefiting from the hierarchy, reasoning and reflection can perceive multi-granularity spatial relations, while each step focuses on relevant context, improving semantic consistency and reasoning efficiency. When the traversal reaches the root node, a complete scene is generated.

the text description of the scene. Each intermediate node is defined as $o^a = \{c^a, s^a, p^a, \theta^a\}$, where c^a is the category label of the area, s^a is the size attribute, p^a is the center position, and θ^a is the orientation. Each leaf node is defined as $o^l = \{x^l, c^l, D^l, s^l, p^l, \theta^l\}$, which includes the appearance description x^l, the category label c^l, the 3D model D^l, the size of the object's oriented bounding box s^l, the center position p^l, and the orientation θ^l. In addition, the pairwise spatial relation is represented by x_e, which provides a textual description of the spatial relation between the current node and an existing relevant node. We define the relevant nodes as anchor nodes. During the generation process, anchor nodes are always prioritized over the current node.

4 Method

This section introduces HierRR, which enhances multi-granularity spatial relation comprehension through top-down hierarchical construction and bottom-up reasoning, achieving semantically consistent scene generation. It consists of three modules: the hierarchy generation module plans the overall layout; the spatial transformation chain-of-thought reasoning module (ST-CoT) explicitly infers placements from spatial relations; and the semantic- and vision-guided reflection module iteratively refines spatial relations and layout reasoning. The pipeline is illustrated in Figure 2. Specifically, we traverse the hierarchy bottom-up. For leaf nodes, we invoke ST-CoT to infer initial placements and then apply fine-grained semantic refinement to update spatial relations. For intermediate nodes, we first perform visual reflection on their child nodes' layouts, and then reason about the current node's placement. Once the traversal reaches the root node, the complete scene layout is generated.

4.1 Hierarchical Structure Generation

To effectively plan layouts and organize spatial relations in the scene according to user intent, we propose a VLM-based hierarchy generation module (HSG). Given a textual user instruction x, HSG converts it into structured text describing the hierarchical scene representation $\mathcal{H}$, formulated as:

$$\mathcal{H} = \Phi_{HSG}(x) \tag{1}$$

The hierarchical structure $\mathcal{H}$ directly guides the layout reasoning and iterative reflection processes to achieve multi-granularity spatial relation understanding. Specifically, it serves two main functions.

First, $\mathcal{H}$ determines the placement sequence. During reasoning and reflection, each node in $\mathcal{H}$ is traversed bottom-up, placing leaf nodes first, followed by parent nodes. Since the construction of $\mathcal{H}$ prioritizes certain anchor nodes (e.g., main furniture), when traversing nodes under the same parent node, anchor nodes are automatically prioritized based on dependency relationships, thereby providing coordinate references for the placement of subsequent nodes. Second, $\mathcal{H}$ defines the contextual scope for reasoning and reflection. The top-down construction of $\mathcal{H}$ decomposes the layout, enabling reasoning and reflection to perceive multi-granularity spatial relations, including those at both the object level and the area level, while focusing attention at each step on the relevant local context guided by global hints. For example, when placing an object, only other objects within the same area need to be considered. Relationships between different areas are addressed at higher levels of $\mathcal{H}$. Furthermore, node attributes and spatial relations in $\mathcal{H}$ are dynamically updated during reflection to adapt to fine-grained layout requirements.

4.2 Spatial Transformation Chain-of-Thought for Layout Reasoning

Determining the center coordinates and orientation of a node based on the spatial relations in the hierarchy $\mathcal{H}$ is a complex decision-making task. This task requires considering the boundaries of the anchor node and the current node, involving multiple constraints such as relative positioning, boundary, and collision constraints. To mitigate the semantic drift between spatial relations and placements, we propose the spatial transformation chain-of-thought reasoning module (ST-CoT) to explicitly model this task. The module takes as input the size s of the node to be placed, the spatial relation description x_e, and the current visual state v. The output is the position coordinates and orientation $\mathcal{P}$ of the node, i.e., a placement decision. We define the ST-CoT reasoning process as follows, where $\mathcal{T}_i$ represents the multi-step reasoning sub-tasks:

$$\begin{aligned} \mathcal{T}_i &\sim \Phi_{ST-CoT}(\mathcal{T}_i|s, x_e, v, \mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_{i-1}) \\ \mathcal{P} &\sim \Phi_{ST-CoT}(\mathcal{P}|s, x_e, v, \mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3, \mathcal{T}_4) \end{aligned} \tag{2}$$

Specifically, given s, x_e, and v, we first refer to the imperative paradigm [40] to guide the VLM in generating an intuitive description of the node's position and orientation. This description leverages the node's bounding box attributes, including its minimum and maximum bounds, center point, and orientation, thereby expressing position and orientation information in an easily understandable manner. Based on this description, the VLM further infers precise numerical coordinates and orientation angles. We divide this reasoning process into the following four key sub-tasks $\mathcal{T}_1$ to $\mathcal{T}_4$:

1) $\mathcal{T}_1$: Determine the associated anchor node and establish a spatial reference coordinate system using the anchor as the origin. Based on the textual description of pairwise spatial relations, a suitable cartesian coordinate system is constructed.

2) $\mathcal{T}_2$: Infer the intuitive positional description of the current node in the spatial reference coordinate system. For example, the x-axis represents the left-right direction. If a chair is to be placed to the right of a table with a certain gap, it can be described as: The minimum x-coordinate value of the chair should be greater than the maximum x-coordinate value of the table, formulated as $chair.min.x > table.max.x$.

3) $\mathcal{T}_3$: Infer the intuitive orientation description of the current node relative to either the anchor node or the spatial reference coordinate system, e.g., "the chair faces the table", formulated as $chair.ori = -table.ori$.

4) $\mathcal{T}_4$: Execute geometric computation and spatial coordinate transformation based on the intuitive positional and orientation descriptions, to calculate the coordinates and orientation degree in the reference coordinate system as $\mathcal{P}$.

Furthermore, once the layout reasoning for all child nodes belonging to the same parent node is completed, we introduce collision detection and boundary constraints to optimize their placements $\{\mathcal{P}_{o_i}\}_{o_i \in O_c}$, ensuring the physical plausibility of the current layout. The formal formulation is as follows:

$$\begin{aligned} &\min_{\mathcal{P}'_{o_i}} \sum\nolimits_{o_i \in O_c} ||\mathcal{P}'_{o_i} - \mathcal{P}_{o_i}||_2^2 \\ &s.t. \quad C_{\text{coll}}(\mathcal{P}'_{o_i}, \mathcal{P}'_{o_j}) = 1, \quad \forall o_i, o_j \in O_c, i \neq j \\ &\qquad C_{\text{bound}}(\mathcal{P}'_{o_i}, s_p) = 1, \quad \forall o_i \in O_c \end{aligned} \tag{3}$$

where O_c is the set of child nodes, $\mathcal{P}'_{o_i}$ is the optimized placement of node o_i, and s_p is the size of its parent. C_{coll} refers to the collision constraint, which ensures no overlap between the oriented bounding boxes of any two nodes. C_{bound} denotes the boundary constraint, ensuring that all nodes are kept within the parent boundaries as much as possible.

4.3 Semantic- and Vision-Guided Reflection

Due to ambiguous spatial relations in the initial hierarchy and the lack of visual verification, ST-CoT may produce results that are misaligned with user intent. Therefore, we design a semantic- and vision-guided reflection module that simulates the human observe-reflect-correct cycle to enhance spatial relation comprehension and iteratively refine the ST-CoT layout reasoning. It comprises two key components: fine-grained semantic refinement (SR) and visual reflection (VR).

Fine-grained Semantic Refinement. Directly inferring spatial relation descriptions x_e from user instructions x via the VLM often yields ambiguous results. To obtain clearer and more detailed descriptions, after the initial ST-CoT reasoning for each leaf node, we guide the VLM to evaluate x_e by integrating the current layout. For x_e that requires refinement, we apply different processing strategies depending on the object type. For objects with complex spatial relations, especially small objects placed on surfaces (e.g., a cup and a monitor on a desk), we first generate a reference image $\mathcal{I}$ from x_e, x, and the related objects o_r. Then, the VLM analyzes $\mathcal{I}$ to extract a more detailed spatial description $\hat{x}_e$. For other objects, we guide the VLM to provide $\hat{x}_e$ based on x and common sense. The process is formulated as follows:

$$\hat{x}_e = \Gamma_{\text{SR}}(o, x, x_e, \mathcal{I}) = \begin{cases} f(x_e, \mathcal{I}), & \text{if } o \in \Omega_{\text{small}} \\ f(x, x_e), & \text{otherwise} \end{cases} \tag{4}$$

where o is the node to be placed and Ω_{small} is the set of small objects determined by the VLM based on each object's category and whether it is placed on a surface. Using $\hat{x}_e$, we again invoke ST-CoT to reason about the node placement. Note that this component is invoked only once for each leaf node.

Visual Reflection. To enable visual verification and further enhance the comprehension of spatial relations, we evaluate the layout and provide feedback before traversing intermediate or root nodes. Specifically, given the set of child nodes O_c, the current visual state v, the user instruction x, and the reflection history $\{\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_{t-1}\}$, the child node layout is assessed from two complementary perspectives. First, a physical score is computed based on collision and out-of-bounds detection. Concurrently, a VLM-based semantic evaluation is introduced to assess the rationality and semantic consistency of the placements, producing corresponding justifications. Furthermore, the VLM diagnoses problematic nodes $\mathcal{Q}$ and proposes refinement suggestions. The process is formulated as follows:

$$\mathcal{R}_t = \Gamma_{\text{VR}}(x, v, O_c, \{\mathcal{R}_1, \mathcal{R}_2, ..., \mathcal{R}_{t-1}\}) \rightarrow \mathcal{Q} \subset O_c \tag{5}$$

where $\mathcal{R}_t$ is the reflection result at iteration t, which is a structured output containing scores, reasons, and suggestions, and its core function is to identify $\mathcal{Q}$. For each problematic node in $\mathcal{Q}$, we append the corresponding suggestions and the erroneous reasoning process to the node's attributes within $\mathcal{H}$. The updated node is then fed into the ST-CoT module to perform re-reasoning. If the reflection score decreases after the update, a rollback is initiated; the faulty suggestion is fed back to the VLM to re-reflect on the current step.

Throughout the traversal process, we invoke a text-to-3D model to generate corresponding 3D assets based on the appearance descriptions of objects. Subsequently, these assets are scaled and placed according to the inferred layout, and then integrated into the current visual state. Once all nodes have been traversed to the root, a complete scene is generated.

5 Experiment and Results

5.1 Experiment Settings

Dataset. We perform comparative and ablation studies using the Scene-Eval-100 dataset [41], which is specifically designed for evaluating text-conditioned scene generation. It comprises 100 free-form indoor scene descriptions ranging from simple to complex, covering various scene types. Each description is paired with human-annotated ground-truth attributes (e.g., object counts and spatial

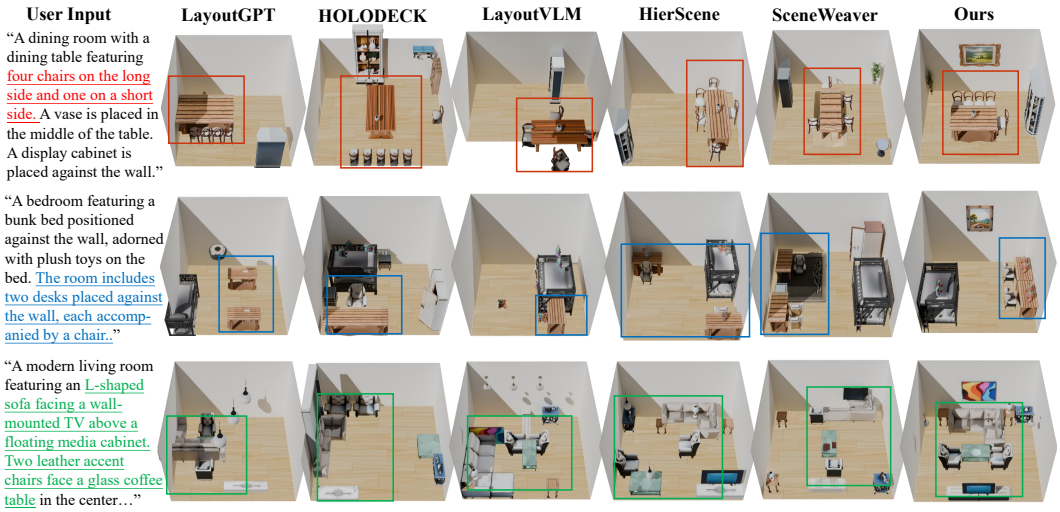

Figure 3: Qualitative comparison. Objects and spatial relations in the user instructions are highlighted with colors and underlines. Compared to existing methods, HierRR generates scene layouts that are more aligned with user intent, visually realistic, and physically feasible.

Table 1: Quantitative comparison between ours and existing methods on Scene-Eval-100 dataset. COL_{ob} is the percentage of objects involved in collisions out of all objects. COL_{sc} is the percentage of scenes containing collisions out of all scenes.

Method	Semantic Consistency			Physical Feasibility			Scene Realism	
	↑ CNT%	↑ OOR%	↑ OAR%	↓ OOB%	↓ COL_{ob}%	↓ COL_{sc}%	↑ Rat.%	↑ Func.%
LayoutGPT	19.54	2.87	14.95	43.10	29.11	43.00	2.89	2.43
HOLODECK	35.44	15.76	29.43	6.99	5.46	18.00	3.39	3.62
LayoutVLM	–	16.33	35.51	19.07	36.92	67.00	3.06	3.36
HierScene	56.90	24.93	42.99	4.11	3.06	7.00	3.53	3.32
SceneWeaver	39.46	16.91	34.11	6.63	8.47	39.00	3.66	3.72
Ours	**72.80**	**41.55**	**62.62**	**0.79**	**0.25**	**1.00**	**4.47**	**4.34**

relations) to enable accurate evaluation of the semantic alignment between the generated scene and its description.

Metrics. We evaluate the generated results across three dimensions: semantic consistency, physical feasibility, and scene realism. For semantic consistency, we assess the match between generated and described object counts (CNT), the accuracy of the spatial relations between objects (OOR), and the spatial relations between objects and architectural elements (e.g., walls and floors) (OAR). For physical feasibility, we compute the collision (COL) and out-of-bounds (OOB) rates using oriented bounding boxes. For scene realism, following [7, 15], we use GPT-4o to assess layout rationality (Rat.) and scene functionality (Func.) based on rendered images and textual instructions.

Implementation. We use GPT-4o [42] for all VLM-based methods, GUROBI [43] for physical constraint optimization, and Hunyuan3D [44] for 3D model generation. Results of generalizing our method to other open-source VLMs and time cost analysis are provided in Appendix B.7 and B.8.

5.2 Comparison

We compare against five state-of-the-art methods: LayoutGPT [9], HOLODECK [4], LayoutVLM [7], HierScene [8], and SceneWeaver [15]. We utilize their original implementations and invoke the same GPT version for inference. Since LayoutVLM requires pre-specified object categories and counts as input, its CNT metric cannot be computed. For a fair comparison, we provide it with the categories and counts generated by our method for evaluation.

Quantitative Evaluation. Table 1 presents the quantitative results. Our method achieves significant improvements over existing approaches. LayoutGPT, which uses in-context learning to directly infer numerical layouts, often yields infeasible layouts. LayoutVLM and HierScene improve physical

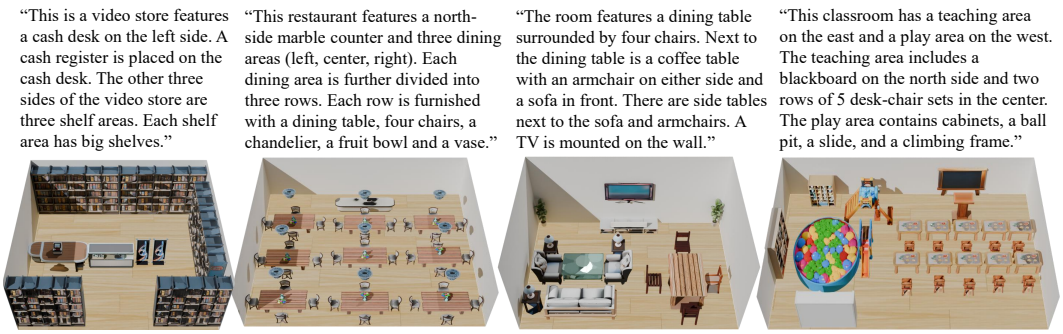

Figure 4: More open-vocabulary scene synthesis results beyond the dataset. HierRR aligns well with user instructions even under challenging conditions, including scenes with numerous objects, complex spatial relationships, and multiple room types.

Table 2: Quantitative evaluation of the ablation study validates our key designs: the ST-CoT for inferring placements, while the SR and VR refine the ST-CoT reasoning process by enriching spatial relation descriptions and performing visual verification.

	Method			Semantic Consistency		Physical Feasibility			Scene Realism	
	ST-CoT	SR	VR	↑ OOR$_{\%}$	↑ OAR$_{\%}$	↓ OOB$_{\%}$	↓ COL$_{ob\%}$	↓ COL$_{sc\%}$	↑ Rat.$_{\%}$	↑ Func.$_{\%}$
Graph	×	×	×	19.48	35.05	50.07	31.75	59.00	3.17	3.28
	✓	×	×	20.92	51.87	14.53	0.51	3.00	3.94	4.11
	✓	✓	×	23.78	52.80	10.43	0.45	3.00	3.85	4.11
	✓	✓	✓	24.93	53.27	7.42	0.45	3.00	4.02	4.17
Hierarchy	×	×	×	32.95	35.98	51.75	34.43	59.00	3.23	3.06
	✓	×	×	38.54	56.54	3.17	0.35	2.00	4.26	4.26
	✓	✓	×	39.83	57.94	2.54	0.50	1.00	4.34	4.32
	✓	✓	✓	**41.55**	**62.62**	**0.79**	**0.25**	**1.00**	**4.47**	**4.34**

feasibility and semantic consistency by fine-tuning VLMs or learning relative relations from datasets, respectively, followed by optimization. However, their performance is limited by specific layout patterns within the training data. HOLODECK and SceneWeaver perform well in terms of scene realism and physical feasibility but struggle with instruction alignment. In contrast, our method leverages the hierarchy to guide iterative reasoning and reflection for refining spatial relations, achieving superior semantic alignment while maintaining realism and feasibility.

Qualitative Evaluation. Figure 3 presents qualitative examples from each method. The results show that our method effectively captures the fine-grained spatial relations specified in the textual instructions. In contrast, LayoutGPT and LayoutVLM often produce overlaps and out-of-bounds placements, while HierScene handles only relatively simple instructions. Although HOLODECK and SceneWeaver generate visually rich scenes, their outputs still exhibit notable shortcomings in semantic alignment. More open-vocabulary results from our method are shown in Figure 4.

Perceptual Study. To further assess result quality, we conduct a perceptual study with 20 participants. We present 36 generated scenes along with their corresponding instructions to the participants, who rate them on a 5-point Likert scale along four aspects: semantic consistency, physical feasibility, layout plausibility, and scene functionality. By providing both rendered scenes and top-view visualizations, participants can detect unreasonable placements that might be occluded in the renderings. As shown in Table 3, our method achieves the highest scores across all four aspects, consistent with the quantitative results in Table 1 and qualitative observations in Figure 3. Additionally, to verify the evaluation capability and stability of GPT-4o, we report human–GPT alignment in Appendix B.1.

5.3 Ablation Study

We conduct ablation studies to evaluate key components. Using direct VLM-based coordinate and orientation inference as baseline, we incrementally integrate spatial transformation chain-of-thought reasoning(ST-CoT), fine-grained semantic refinement(SR), and visual reflection(VR). To verify the advantages of hierarchy, we assess each component under two scene representations: a relational

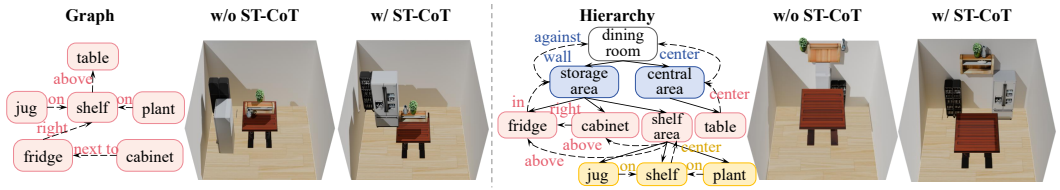

Figure 5: Visualization of ST-CoT on graph and hierarchy. While effective in both, ST-CoT performs better on the hierarchy by planning multi-level relations and placement order to better guide reasoning.

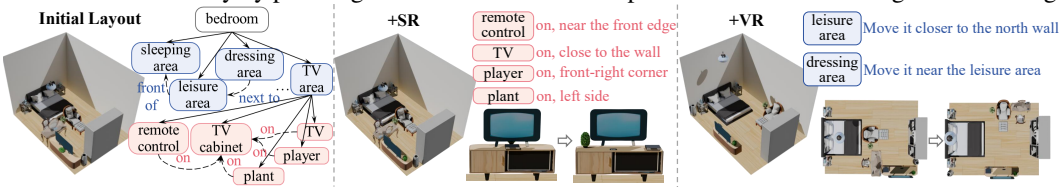

Figure 6: Visualization of SR and VR across multi-granularity spatial relations. SR enriches object spatial relations in accordance with user intent. VR improves regional layout from a visual perspective.

graph and a hierarchy. The relational graph setup performs layout inference via sequential traversal and global reflection without partitioning. Results are shown in Table 2.

1) ST-CoT improves semantic consistency and physical feasibility through explicit modeling. Table 2 shows that ST-CoT boosts performance in both graph and hierarchical configurations, with superior results in the latter. Figure 5 illustrates its effectiveness in reasoning about plausible placement.

Table 3: Averaged score of perceptual study.

Method	Semant.	Phys.	Ration.	Funct.
LayoutGPT	2.27	1.91	2.11	2.30
HOLODECK	2.72	3.40	3.29	3.52
LayoutVLM	2.83	1.96	2.14	2.68
HierScene	3.39	3.86	3.30	3.31
SceneWeaver	2.98	3.61	3.33	3.68
Ours	**4.83**	**4.86**	**4.78**	**4.68**

Table 4: Impact of the number of reflection iterations.

Iter.	↑ $OOR_{\%}$	↑ $OAR_{\%}$	↓ $OOB_{\%}$	↓ $COL_{ob\%}$	↓ $COL_{sc\%}$
0	39.83	57.94	2.54	0.50	1.00
1	40.11	60.75	1.75	0.52	2.00
3	41.55	62.62	0.79	0.25	1.00
5	41.55	64.02	0.79	0.24	1.00

2) SR enriches spatial relation descriptions, enabling ST-CoT to infer more reasonable object placement. Figure 6 shows that SR refines coarse spatial relations (e.g., objects are roughly on the TV cabinet) into more detailed descriptions without altering user intent, yielding more plausible results. 3) VR assesses and refines spatial relations from a visual perspective to optimize layouts, enhancing semantic consistency and visual realism. VR improves OOR, OAR, and Rat. by 4.3%, 8.1%, and 3.0% in Table 2. Figure 6 shows that it adjusts the positions of two areas for a more reasonable layout. 4) The hierarchy enables more efficient layout generation than graph structures. Through multi-level organization, reasoning and reflection comprehend spatial relations at multiple granularities, while each step is more focused, avoiding confusion from information overload.

5.4 In-depth Analysis of Reflection Iterations

To further analyze the impact of visual reflection iterations on the final layout, we evaluate performance across different iteration rounds, as shown in Table 4. We observe that after three iterations, both semantic consistency (OOR, OAR) and physical feasibility (OOB, COL) improve significantly. By five iterations, all metrics gradually converge and stabilize. In contrast to SceneWeaver, which iterates 10 times under its default settings, our method achieves superior performance with fewer iterations. Balancing generation quality and iteration cost, we set the number of iterations to three.

6 Conclusion

To address the semantic inconsistency caused by insufficient spatial relation understanding in open-vocabulary indoor scene synthesis, we propose HierRR, which organizes spatial relations through a hierarchy and performs spatial transformation chain-of-thought reasoning combined with semantic- and vision-guided reflection to explicitly model the reasoning process and refine spatial relations. Benefiting from the hierarchy, HierRR enhances the comprehension of multi-granularity spatial relations and enables attributable layout generation, thereby improving semantic consistency.

While HierRR is promising, it still has some limitations. First, to simplify the reasoning process, we assume rectangular room structures in the main paper experiments, but our method is also applicable to irregular room structures (see Appendix B.9). Second, HierRR represents objects as oriented bounding boxes without geometric details, which limits its performance on specific shapes (e.g., L-shaped sofas) and internal placements. Future work may incorporate more detailed shape representations and support-relation reasoning to further improve structural plausibility.

Acknowledgments and Disclosure of Funding

This work was supported by the National Natural Science Foundation of China (Grant No. 62677043); the Shandong Key Laboratory of Foundational Software (11150004040955); and the Engineering Research Center of Digital Media Technology, Ministry of Education.

References

[1] M. Deitke, E. VanderBilt, A. Herrasti, L. Weihs, K. Ehsani, J. Salvador, W. Han, E. Kolve, A. Kembhavi, and R. Mottaghi, "Procthor: Large-scale embodied ai using procedural generation," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, pp. 5982–5994, 2022.

[2] J. Huang, S. Yong, X. Ma, X. Linghu, P. Li, Y. Wang, Q. Li, S.-C. Zhu, B. Jia, and S. Huang, "An embodied generalist agent in 3d world," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2024.

[3] Y. Yang, B. Jia, P. Zhi, and S. Huang, "Physcene: Physically interactable 3d scene synthesis for embodied ai," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024, pp. 16 262–16 272.

[4] Y. Yang, F.-Y. Sun, L. Weihs, E. VanderBilt, A. Herrasti, W. Han, J. Wu, N. Haber, R. Krishna, L. 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 (CVPR)*, 2024, pp. 16 227–16 237.

[5] R. Fu, Z. Wen, Z. Liu, and S. Sridhar, "Anyhome: Open-vocabulary generation of structured and textured 3d homes," in *European Conference on Computer Vision (ECCV)*. Springer, 2024, pp. 52–70.

[6] A. Çelen, G. Han, K. Schindler, L. Van Gool, I. Armeni, A. Obukhov, and X. Wang, "I-design: Personalized llm interior designer," in *Computer Vision – ECCV 2024 Workshops*, 2025, p. 217–234.

[7] F.-Y. Sun, W. Liu, S. Gu, D. Lim, G. Bhat, F. Tombari, M. Li, N. Haber, and J. Wu, "Layoutvlm: Differentiable optimization of 3d layout via vision-language models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025, pp. 29 469–29 478.

[8] W. Sun, X. Li, M. Li, K. Xu, X. Meng, and L. Meng, "Hierarchically-structured open-vocabulary indoor scene synthesis with pre-trained large language model," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 7, 2025, pp. 7122–7130.

[9] W. Feng, W. Zhu, T.-J. Fu, V. Jampani, A. R. Akula, X. He, S. Basu, X. E. Wang, and W. Y. Wang, "Layoutgpt: Compositional visual planning and generation with large language models," *arXiv preprint arXiv:2305.15393*, 2023.

[10] Y. Yang, J. Lu, Z. Zhao, Z. Luo, J. J. Yu, V. Sanchez, and F. Zheng, "Llplace: The 3d indoor scene layout generation and editing via large language model," *arXiv preprint arXiv:2406.03866*, 2024.

[11] X. Ran, Y. Li, L. Xu, M. Yu, and B. Dai, "Direct numerical layout generation for 3d indoor scene synthesis via spatial reasoning," *arXiv preprint arXiv:2506.05341*, 2025.

[12] F. Berdoz, L. A. Lanzendörfer, N. Tuninga, and R. Wattenhofer, "Text-to-scene with large reasoning models," *arXiv preprint arXiv:2509.26091*, 2025.

[13] Z. Hu, A. Iscen, A. Jain, T. Kipf, Y. Yue, D. A. Ross, C. Schmid, and A. Fathi, "Scenecraft: An llm agent for synthesizing 3d scenes as blender code." in *Proceedings of the International Conference on Machine Learning (ICML)*, 2024, pp. 19 252–19 282.

[14] C. Wang, H. Zhong, M. Chai, M. He, D. Chen, and J. Liao, "Chat2layout: Interactive 3d furniture layout with a multimodal llm," *arXiv preprint arXiv:2407.21333*, 2024.

[15] Y. Yang, B. Jia, S. Zhang, and S. Huang, "Sceneweaver: All-in-one 3d scene synthesis with an extensible and self-reflective agent," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[16] R. Ma, H. Li, C. Zou, Z. Liao, X. Tong, and H. Zhang, "Action-driven 3d indoor scene evolution," *ACM Transactions on Graphics*, vol. 35, pp. 1 – 13, 2016.

[17] Q. Fu, X. Chen, X. Wang, S. Wen, B. Zhou, and H. Fu, "Adaptive synthesis of indoor scenes via activity-associated object relation graphs," *ACM Transactions on Graphics*, vol. 36, pp. 1 – 13, 2017.

[18] Q. Fu, H. Fu, H. Yan, B. Zhou, X. Chen, and X. Li, "Human-centric metrics for indoor scene assessment and synthesis," *Graphical Models*, vol. 110, p. 101073, 2020.

[19] S. Qi, Y. Zhu, S. Huang, C. Jiang, and S.-C. Zhu, "Human-centric indoor scene synthesis using stochastic grammar," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5899–5908, 2018.

[20] R. Ma, A. G. Patil, M. Fisher, M. Li, S. Pirk, B.-S. Hua, S.-K. Yeung, X. Tong, L. Guibas, and H. Zhang, "Language-driven synthesis of 3d scenes from scene databases," *ACM Transactions on Graphics*, vol. 37, no. 6, pp. 1–16, 2018.

[21] W. Sun, X. Li, M. Li, Y. Wang, Y. Zheng, X. Meng, and L. Meng, "Sequential fusion of multi-view video frames for 3d scene generation," *CICAI*, 2022.

[22] W. Sun, M. Li, P. Li, X. Cao, X. Meng, and L. Meng, "Sequential selection and calibration of video frames for 3d outdoor scene reconstruction," *CAAI Transactions on Intelligence Technology*, 2024.

[23] Y. Wang, X. Qiu, J. Liu, Z. Chen, J. Cai, Y. Wang, T.-H. Wang, Z. Xian, and C. Gan, "Architect: Generating vivid and interactive 3d scenes with hierarchical 2d inpainting," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 37, pp. 67 575–67 603, 2024.

[24] Z. Gu, Y. Cui, Z. Li, F. Wei, Y. Ge, J. Gu, M.-Y. Liu, A. Davis, and Y. Ding, "Artiscene: Language-driven artistic 3d scene generation through image intermediary," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025, pp. 2891–2901.

[25] B. Chen, Z. Xu, S. Kirmani, B. Ichter, D. Sadigh, L. Guibas, and F. Xia, "Spatialvlm: Endowing vision-language models with spatial reasoning capabilities," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 14 455–14 465.

[26] A.-C. Cheng, H. Yin, Y. Fu, Q. Guo, R. Yang, J. Kautz, X. Wang, and S. Liu, "Spatialrgpt: Grounded spatial reasoning in vision-language models," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 37, pp. 135 062–135 093, 2024.

[27] E. Zhou, J. An, C. Chi, Y. Han, S. Rong, C. Zhang, P. Wang, Z. Wang, T. Huang, L. Sheng *et al.*, "Roborefer: Towards spatial referring with reasoning in vision-language models for robotics," *arXiv preprint arXiv:2506.04308*, 2025.

[28] J. Wei, X. Wang, D. Schuurmans, M. Bosma, b. ichter, F. Xia, E. Chi, Q. V. Le, and D. Zhou, "Chain-of-thought prompting elicits reasoning in large language models," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, 2022, pp. 24 824–24 837.

[29] S. Yao, D. Yu, J. Zhao, I. Shafran, T. Griffiths, Y. Cao, and K. Narasimhan, "Tree of thoughts: Deliberate problem solving with large language models," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 36, 2023, pp. 11 809–11 822.

[30] M. Besta, N. Blach, A. Kubicek, R. Gerstenberger, L. Gianinazzi, J. Gajda, T. Lehmann, M. Podstawski, H. Niewiadomski, P. Nyczyk, and T. Hoefler, "Graph of Thoughts: Solving Elaborate Problems with Large Language Models," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 16, pp. 17 682–17 690, Mar 2024.

[31] Y. Liu, D. Chi, S. Wu, Z. Zhang, Y. Hu, L. Zhang, Y. Zhang, S. Wu, T. Cao, G. Huang *et al.*, "Spatial-cot: Advancing spatial reasoning through coordinate alignment and chain-of-thought for embodied task planning," *arXiv preprint arXiv:2501.10074*, 2025.

[32] L. Wang, C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang, Z. Chen, J. Tang, X. Chen, Y. Lin *et al.*, "A survey on large language model based autonomous agents," *Frontiers of Computer Science*, vol. 18, no. 6, p. 186345, 2024.

[33] Z. Xi, W. Chen, X. Guo, W. He, Y. Ding, B. Hong, M. Zhang, J. Wang, S. Jin, E. Zhou *et al.*, "The rise and potential of large language model based agents: A survey. arxiv 2023," *arXiv preprint arXiv:2309.07864*, vol. 10, 2025.

[34] N. Shinn, F. Cassano, A. Gopinath, K. Narasimhan, and S. Yao, "Reflexion: language agents with verbal reinforcement learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 36, 2023, pp. 8634–8652.

[35] X. Huang, S. Cheng, S. Huang, J. Shen, Y. Xu, C. Zhang, and Y. Qu, "Queryagent: A reliable and efficient reasoning framework with environmental feedback-based self-correction," *arXiv preprint arXiv:2403.11886*, 2024.

[36] L. Ling, C.-H. Lin, T.-Y. Lin, Y. Ding, Y. Zeng, Y. Sheng, Y. Ge, M.-Y. Liu, A. Bera, and Z. Li, "Scenethesis: A language and vision agentic framework for 3d scene generation," *arXiv preprint arXiv:2505.02836*, 2025.

[37] J. Gao, D. Zhou, M. Liang, L. Liu, C.-W. Fu, X. Hu, and P.-A. Heng, "Disco-layout: Disentangling and coordinating semantic and physical refinement in a multi-agent framework for 3d indoor layout synthesis," *arXiv preprint arXiv:2510.02178*, 2025.

[38] H. Xia, X. Li, Z. Li *et al.*, "Sage: Scalable agentic 3d scene generation for embodied ai," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026, pp. 22 358–22 368.

[39] N. Pfaff, T. Cohn, S. Zakharov *et al.*, "Scenesmith: Agentic generation of simulation-ready indoor scenes," *arXiv preprint arXiv:2602.09153*, 2026.

[40] M. Gumin, D. H. Han, S. J. Yoo, A. Ganeshan, R. K. Jones, K. Fu, R. Aguina-Kang, S. Morris, and D. Ritchie, "Procedural scene programs for open-universe scene generation: Llm-free error correction via program search," in *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, 2025, pp. 1–11.

[41] H. I. I. Tam, H. I. D. Pun, A. T. Wang, A. X. Chang, and M. Savva, "SceneEval: Evaluating semantic coherence in text-conditioned 3D indoor scene synthesis," in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2026.

[42] 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.

[43] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2023.

[44] Z. Zhao, Z. Lai, Q. Lin, Y. Zhao, H. Liu, S. Yang, Y. Feng, M. Yang, S. Zhang, X. Yang *et al.*, "Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation," *arXiv preprint arXiv:2501.12202*, 2025.

A Details of Our Framework

A.1 Hierarchical Structure Generation

This module guides the VLM to generate structured text describing the hierarchical structure of the scene based on the user's textual instructions. The prompt consists of three parts: (1) a description of the VLM's role and task, including a brief definition of the hierarchical structure and the meaning of nodes and edges; (2) the preferred data format; (3) a concrete example in the preferred format to help the VLM understand the task. The complete prompt is provided in Figure 7.

A.2 Layout Reasoning and Reflection Process

We provide a more detailed algorithmic workflow of hierarchical layout reasoning and reflection in Algorithm 1. The scene hierarchy is traversed in a bottom-up manner, progressively determining the precise placement of each node. For leaf nodes, we first invoke the spatial transformation chain-of-thought module to infer their initial placements, followed by fine-grained semantic refinement to improve the spatial descriptions. For intermediate nodes, we perform a visual reflection on the layout of all their child nodes before reasoning about the current node's placement. When the traversal reaches the root node, the complete scene layout is generated.

During the layout reasoning process, the spatial transformation chain-of-thought module takes the node's size s, its spatial relation description x_e, and the visual state v as input, and outputs its position and orientation. Here, the visual state v represents the visual contextual information within the parent node to which the current node belongs. Throughout this process, we guide the VLM to perform layout reasoning through designed prompts. The complete prompt templates are illustrated in Figures 8 and 9. Before traversing to intermediate or root nodes, we apply physical constraints to optimize the layout for physical plausibility. The details can be found in Appendix A.3.

During the reflection process, we apply different strategies according to node types. Upon traversing to a leaf node, we first prompt the VLM to evaluate whether the spatial relationship description x_e is sufficiently detailed by integrating the current layout. For nodes requiring further refinement, we process them according to their object types to obtain more precise spatial relation descriptions $\hat{x}_e$. The complete prompt templates are shown in Figure 10. Before traversing intermediate or root nodes, we conduct the visual reflection to iteratively evaluate the overall layout of all child nodes from two perspectives. Specifically, the layout is scored based on physical feasibility, such as collision and out-of-bounds detection, while the VLM assesses the rationality and semantic consistency of the child nodes' positions and orientations, providing supporting justifications. The total score for the current layout is then derived as follows:

$$S_{total} = \omega_p \cdot S_{phys} + \omega_v \cdot S_{vlm} \tag{6}$$

where S_{phys} and S_{vlm} represent the physical feasibility and VLM-based evaluation scores, respectively, with weights $\omega_p = \omega_v = 0.5$. This scoring mechanism allows for the identification of critical issues within the generated layout. For scenes falling below a certain threshold, we further prompt the VLM to diagnose problematic nodes based on the detected issues and to propose refinement suggestions. The complete prompt templates are provided in Figure 11 and Figure 12. Based on this, we extract a set of problematic nodes $\mathcal{Q}$ accompanied by detailed suggestions. For each node in $\mathcal{Q}$, the corresponding suggestion is fed back into the layout reasoning module to guide the re-reasoning step, followed by a subsequent VLM re-scoring. If the evaluation score decreases, the erroneous suggestion is fed back to the VLM to trigger a new reflection for the current step.

A.3 Layout Optimization with Physical Constraint

To ensure the physical plausibility of the synthesized scenes, we employ physical constraints, such as collision and out-of-boundary, to refine the layout reasoning results. Specifically, during the bottom-up traversal, when an intermediate or root node is reached and its child nodes have completed layout reasoning, we optimize the placement of these child nodes. We present the detailed formulation for the collision and out-of-bounds constraints in the following.

We conduct the optimization in the coordinate frame of the current intermediate or root node (i.e., the parent node). For a child node o_i, its center coordinates are defined as (x_i, y_i, z_i), and the dimensions of its oriented bounding box along the x-axis, y-axis, and z-axis are denoted by $(\Delta x_i, \Delta y_i, \Delta z_i)$.

Algorithm 1 Hierarchical Reasoning and Reflection

Input: Hierarchical structure $\mathcal{H}$, user textual instruction x, number of iterations T_{max}
Output: Scene layout $\mathcal{L}$

$v \leftarrow \emptyset$; $\mathcal{L} \leftarrow \emptyset$; $nodes \leftarrow$ BottomUpTraversal($\mathcal{H}$)
for each node $o_i \in nodes$ **do**
  $v, \mathcal{L} \leftarrow$ GetLayout(o_i)
  if o_i is a leaf node **then**
    $\mathcal{P}_i \leftarrow$ Spatial-CoT($s_i, x_{e,i}, v$)
    $v' \leftarrow$ UpdatedVisualState($\mathcal{P}_i, v$)
    if NeedReflect($x_{e,i}$, v') **then**
      $\hat{x}_{e,i} \leftarrow$ SemanticRefinement($o_i, x, x_{e,i}$)
      $\mathcal{P}_i \leftarrow$ Spatial-CoT($s_i, \hat{x}_{e,i}, v$)
    end if
    $v, \mathcal{L} \leftarrow$ UpdateLayout($\mathcal{P}_i, \mathcal{L}$)
  else
    $\mathcal{R} \leftarrow [\,]$; $t \leftarrow 0$
    $v_c, \mathcal{L}_c, \mathcal{P}_c, O_{c,i} \leftarrow$ GetChildren(o_i)
    $\mathcal{P}_{\text{optimized},c} \leftarrow$ PhysicalOptim($\mathcal{P}_c$)
    $v_c, \mathcal{L}_c \leftarrow$ UpdatedLayout($\mathcal{P}_{\text{optimized},c}, \mathcal{L}_c$)
    $S_0 \leftarrow$ GetScore(v_c)
    while $t < T_{max}$ **do**
      $\mathcal{R}_t \leftarrow$ VisualReflection($x, v_c, O_{c,i}, \mathcal{R}$)
      $Q \leftarrow$ GetProblemNodes($\mathcal{R}_t$)
      $v'_c \leftarrow v_c$; $\mathcal{L}'_c \leftarrow \mathcal{L}_c$
      for each problem node $q_i \in Q$ **do**
        $\mathcal{P}_{q_i} \leftarrow$ Spatial-CoT($s_{q_i}, x_{e,q_i}, v'_c, \mathcal{R}_t$)
        $v'_c \leftarrow$ UpdatedVisualState($\mathcal{P}_{q_i}, v'_c$)
      end for
      $S_t \leftarrow$ GetScore(v'_c)
      if $S_t >= S_{t-1}$ or $S_t >$ Threshold **then**
        $\mathcal{P}_{\text{optimized},c} \leftarrow$ PhysicalOptim($\mathcal{P}_c \cup \mathcal{P}_q$)
        $v'_c, \mathcal{L}'_c \leftarrow$ UpdateLayout($\mathcal{P}_{\text{optimized},c}, \mathcal{L}'_c$)
        $\mathcal{R}$.append($\{\mathcal{R}_t$: "right"$\}$);
      else if $S_t >$ Threshold **then**
        break
      else
        $\mathcal{R}$.append($\{\mathcal{R}_t$: "wrong"$\}$);
      end if
      $t \leftarrow t + 1$
    end while
    if o_i is not a root node **then**
      $\mathcal{P}_i \leftarrow$ Spatial-CoT($s_i, x_{e,i}, v$)
      $v, \mathcal{L} \leftarrow$ UpdateLayout($\mathcal{P}_i, \mathcal{L}$)
    end if
  end if
end for
return $\mathcal{L}$

The boundary dimensions of the parent node along the x-axis, y-axis, and z-axis are represented by (s_x, s_y, s_z). The optimization uses C_{coll} and C_{bound} constraints. C_{coll} prevents the oriented bounding boxes of any two child nodes under the same parent from overlapping. The formulation is as follows:

$$
\begin{aligned}
&C_{\text{coll}}(\mathcal{P}'_{o_i}, \mathcal{P}'_{o_j}) = \begin{cases} 0 & x_{ij} \cdot y_{ij} \cdot z_{ij} > 0 \\ 1 & x_{ij} \cdot y_{ij} \cdot z_{ij} \leq 0 \end{cases}, \\
&x_{ij} = (\Delta x_i + \Delta x_j)/2 - |x_i - x_j|, \\
&y_{ij} = (\Delta y_i + \Delta y_j)/2 - |y_i - y_j|, \\
&z_{ij} = (\Delta z_i + \Delta z_j)/2 - |z_i - z_j|.
\end{aligned} \tag{7}
$$

Table 5: User-User and User-GPT Alignment.

Method	Layout Rationality	Scene Functionality
User-User	0.669	0.590
User-GPT	0.578	0.563

C_{bound} prevents all child nodes from exceeding the boundary of their parent node. The formulation is as follows:

$$
\begin{aligned}
&C_{\text{bound}}(\mathcal{P}'_{o_i}, s_p) = m_i \cdot n_i \cdot h_i, \\
&m_i = \begin{cases} 0, & x_i < \Delta x_i/2,\ x_i > s_x - \Delta x_i/2 \\ 1, & \Delta x_i/2 \le x_i \le s_x - \Delta x_i/2 \end{cases}, \\
&n_i = \begin{cases} 0, & y_i < \Delta y_i/2,\ y_i > s_y - \Delta y_i/2 \\ 1, & \Delta y_i/2 \le y_i \le s_y - \Delta y_i/2 \end{cases}, \\
&h_i = \begin{cases} 0, & z_i < \Delta z_i/2,\ z_i > s_z - \Delta z_i/2 \\ 1, & \Delta z_i/2 \le z_i \le s_z - \Delta z_i/2 \end{cases}.
\end{aligned}
\tag{8}
$$

B Details of Our Experiments

B.1 Evaluation Details

In the quantitative evaluation of comparative experiments and ablation studies, we refer to Scene-Eval, using OOB and COL to measure the physical plausibility of the generated results, and CNT, OOR, and OAR to assess semantic consistency. For OOB and COL, we construct mesh cuboids based on the 3D bounding boxes of objects and perform ray casting and collision detection for evaluation. In the case of COL, we report two metrics separately: 1) COL_{ob}, the percentage of objects that are involved in collisions out of all objects; 2) COL_{sc}, the percentage of scenes containing collisions out of all scenes. For scene realism, we provide rendered images of the generated scenes, user instructions as input, and use GPT-4o to score the results from the perspectives of layout rationality and scene functionality. The prompt is detailed in Figure 13.

To further validate the effectiveness of GPT-4o for scoring, we employed the same evaluation dimensions of layout rationality and scene functionality in the perceptual study. Using the same test examples from the perceptual study, we converted both user and GPT-4o scores into rankings and calculated Kendall's Tau, as shown in Table 5. The results demonstrate strong user-user agreement and user-GPT agreement on both layout rationality and scene functionality metrics. They also demonstrate the validity of using GPT-4o for scoring.

B.2 Perceptual Study Details

We invited 20 participants for the perceptual study. For each participant, we present the generation results of five baseline methods and ours across six sets of user instructions, resulting in a total of 36 scenes. The scenes are displayed in random order. Figures 14 and 15 show the interface provided to the participants. They are asked to evaluate each generated scene along the following four dimensions:

1) Semantic Consistency: how well the generated layout aligns with the user's textual description, including object categories, quantities, relative positions, and orientations.

2) Physical Feasibility: the feasibility of the object arrangement in the real world, considering whether collisions or out-of-bounds objects exist, and whether sufficient space is provided for daily activities.

3) Layout Rationality: the overall plausibility and realism of the object organization and placement within the scene.

4) Scene Functionality: the richness of objects within the scene, i.e., whether the variety and quantity of objects meet the needs of daily life.

Table 6: Isolating the contribution of HierRR over HierScene on Scene-Eval-100. HierScene+STR denotes a HierScene-style hierarchy equipped with ST-CoT and reflection.

Method	Semantic Consistency			Physical Feasibility		
	↑ CNT$_{\%}$	↑ OOR$_{\%}$	↑ OAR$_{\%}$	↓ OOB$_{\%}$	↓ COL$_{ob\%}$	↓ COL$_{sc\%}$
HierScene	56.90	24.93	42.99	4.11	3.06	7.00
HierScene+STR	56.90	36.73	45.24	3.95	2.97	6.00
HierRR(Ours)	**72.80**	**41.55**	**62.62**	**0.79**	**0.25**	**1.00**

Table 7: Comparison of the overall agent budget with SceneWeaver, including reflection rounds, total token consumption and runtime per scene, together with the corresponding performance.

Method	Rounds	Tokens	Time	Semantic Consistency			Physical Feasibility		
				↑ CNT$_{\%}$	↑ OOR$_{\%}$	↑ OAR$_{\%}$	↓ OOB$_{\%}$	↓ COL$_{ob\%}$	↓ COL$_{sc\%}$
SceneWeaver	3	40k–70k	25 min	38.31	13.75	32.71	7.76	15.67	50.00
SceneWeaver(default)	10	130k–200k	64 min	39.46	16.91	34.11	6.63	8.47	39.00
HierRR(Ours)	3	60k–120k	35 min	**72.80**	**41.55**	**62.62**	**0.79**	**0.25**	**1.00**

For each scene, we provide the user instruction, a rendered image, and a top-view visualization with 2D bounding boxes to clearly show the number, positions, and orientations of objects. Ratings are given on a 5-point scale (5: Excellent; 4: Good; 3: Fair; 2: Poor; 1: Bad). We collect all participant ratings and calculate the average scores of HierRR and the five baseline methods across the four dimensions above. The results are reported in the paper.

B.3 Detailed comparison with HierScene

HierRR differs from HierScene [8] in both hierarchical depth and spatial relation modeling. HierScene adopts a fixed three-level hierarchy (scene-area-object), whereas HierRR employs a recursive multi-level hierarchy, in which each area can be further decomposed into several sub-areas according to the scene complexity, as shown in Figure 5. Moreover, HierScene models only parent-child and object-level relations, while HierRR additionally introduces explicit area-to-area spatial relations, providing richer structural constraints for spatial reasoning. More importantly, HierRR further demonstrates the role of hierarchy in agentic reasoning and reflection.

Since the two methods adopt different hierarchies, we further evaluate a HierScene-style hierarchy equipped with ST-CoT and reflection (HierScene+STR), in order to isolate the contribution of HierRR over HierScene. As shown in Table 6, two conclusions can be drawn. First, comparing the first two rows, keeping the HierScene hierarchy unchanged and only introducing ST-CoT and reflection significantly improves the performance, particularly on OOR, indicating that explicit multi-step spatial reasoning and iterative reflection enable a more accurate mapping from spatial relations to object layouts. Second, comparing the last two rows, keeping ST-CoT and reflection unchanged and replacing the HierScene hierarchy with the HierRR hierarchy yields further improvements, demonstrating that the recursive hierarchy and the explicit inter-area spatial relations provide richer structural constraints, which enable more effective layout reasoning and reflection.

B.4 Detailed comparison with SceneWeaver

Since both HierRR and SceneWeaver are agentic frameworks that employ iterative reflection, we provide a more detailed comparison of their inference efficiency and scalability.

Inference efficiency. Table 7 compares the overall agent budget with SceneWeaver in terms of reflection rounds, total token consumption, and runtime per scene, together with the corresponding performance. Although SceneWeaver consumes fewer tokens per reflection round, it requires substantially more rounds (10 by default), typically 130k–200k tokens and 64 minutes per scene, whereas HierRR requires only 60k–120k tokens and 35 minutes while achieving much better performance. Moreover, when SceneWeaver is restricted to the same number of reflection rounds as HierRR (3), its performance drops clearly and remains far below ours, confirming that the gains of HierRR do not stem from a larger agent budget.

Table 8: Inference time (in minutes) for different object counts.

Method	<=5	6-10	11-15	>=16
Scene Weaver	20.78	69.11	90.08	128.79
HierRR(Ours)	14.80	27.91	50.47	69.68

Table 9: Quantitative comparison on 50 scenes randomly sampled from the SAGE-10K dataset.

Method	Semantic Consistency			Physical Feasibility		
	$\uparrow$ CNT$_{\%}$	$\uparrow$ OOR$_{\%}$	$\uparrow$ OAR$_{\%}$	$\downarrow$ OOB$_{\%}$	$\downarrow$ COL$_{ob\%}$	$\downarrow$ COL$_{sc\%}$
HierScene	41.13	17.54	21.30	0.56	4.41	6.00
SceneWeaver	27.42	17.54	32.41	14.98	10.25	36.00
HierRR(Ours)	**79.03**	**52.63**	**58.33**	**0.39**	**3.61**	**6.00**

Scalability. We evaluate how the number of objects affects the inference time of SceneWeaver and HierRR, as shown in Table 8. As scene complexity increases, HierRR's inference time grows more slowly than that of SceneWeaver, indicating better scalability. This advantage stems from its more efficient hierarchical reasoning structure.

B.5 Performance on SAGE-10K datasets

To further evaluate the generalization of HierRR under a different data distribution, we additionally sample 50 scenes of different room types from the SAGE-10K dataset, and compare HierRR with HierScene and SceneWeaver. As shown in Table 9, HierRR achieves the best performance across all metrics, demonstrating that its improvements are not limited to Scene-Eval-100, but generalize well to scene descriptions from a different data source.

B.6 Performance across Varying Scene Complexity

We further group the results on Scene-Eval-100 by the number of objects in each scene. As shown in Table 10, HierRR performs stably on low- and medium-density scenes ($\leq$10 objects). As the scene complexity grows ($\geq$16 objects), some metrics decline moderately; however, the collision rate(COL), out-of-bound rate(OOB), and spatial relations between objects and architectural elements(OAR) remain at favorable levels. These results demonstrate that the proposed hierarchical reasoning framework generalizes well to increasingly cluttered scenes. The corresponding trend of inference time is analyzed in Section B.4.

B.7 Generalization across Foundation Models

To verify the generalization and robustness, we evaluate HierRR on multiple foundation models, including the open-source Mistral-3.2, Qwen-3.5 and Qwen-3.6, as well as the closed-source GPT-4o and GPT-5.5. As shown in Table 11, +HierRR consistently improves over the baseline of the same foundation model, confirming that the performance gains primarily stem from our algorithmic design rather than any specific model. Moreover, HierRR benefits from stronger foundation models, which yield further improvements in scene generation quality.

B.8 Computational Cost

We conduct a detailed analysis of our method's inference time. Generating a complete scene takes approximately 35 minutes on average, with the time consumption primarily attributed to

Table 10: Performance of HierRR on Scene-Eval-100 across scenes with different numbers of objects.

Number of Objects	$\uparrow$ OOR$_{\%}$	$\uparrow$ OAR$_{\%}$	$\downarrow$ OOB$_{\%}$	$\downarrow$ COL$_{ob\%}$	$\downarrow$ COL$_{sc\%}$
≤ 5	60.00	81.81	0.00	0.00	0.00
6–10	55.33	83.33	0.00	0.00	0.00
11–15	43.24	54.62	0.00	0.00	0.00
≥ 16	30.75	62.29	6.36	2.94	16.67

Table 11: Results from different foundation models. *Baseline* denotes directly generating the layout from the graph or hierarchy representation with a structured prompt, without ST-CoT or reflection.

Model		↑ OOR	↑OAR	↓OOB	↓ COL_{ob}	↓ COL_{sc}
Mistral-3.2	Baseline	37.83	10.14	43.14	54.5	83.00
	+HierRR	48.45	33.35	1.96	0.06	1.00
Qwen-3.5	Baseline	24.32	31.35	21.76	22.68	50.00
	+HierRR	40.54	44.42	3.92	0.05	1.00
Qwen-3.6	Baseline	29.73	33.25	23.53	8.47	33.00
	+HierRR	48.65	54.45	0.00	0.00	0.00
GPT-4o	Baseline	19.48	35.05	50.07	31.75	59.00
	+HierRR	41.55	62.62	0.79	0.25	1.00
GPT-5.5	Baseline	32.43	45.75	17.65	15.67	50.00
	+HierRR	54.05	66.64	0.00	0.00	0.00

Table 12: Comparison of approximate LLM token consumption for generating a single scene.

Method	LayoutGPT	HOLODECK	LayoutVLM	HierScene	SceneWeaver	HierRR(Ours)
Tokens / Scene	2k–5k	40k–60k	7k–12k	3k–8k	130k–200k	60k–120k

two factors. First, to obtain high-quality 3D assets, we employ Hunyuan3D for generation, which requires approximately 3 minutes per object—this is a necessary overhead rather than an algorithmic bottleneck. Second, the iterative reflection mechanism takes 4.7 minutes per iteration. While this introduces additional computational overhead, it significantly improves semantic consistency (OOR increases by 7.81% and OAR by 10.75%). Moreover, our method typically converges within 3 iterations, avoiding infinite iteration loops.

Regarding token consumption, Table 12 compares the approximate LLM token usage of different methods for generating a single scene. Compared with SceneWeaver, which also performs iterative refinement, HierRR reduces token consumption by nearly 50%. This reduction stems from the hierarchical representation, which allows each reflection step to target specific nodes, precisely localizing problems and reducing the number of required iterations. Compared with one-pass methods, HierRR incurs higher token costs. However, this additional cost leads to substantial improvements in semantic consistency and physical plausibility.

B.9 Extension to Complex Room Structures

Without modifying the core modules, our method can be extended to rooms that include doors and windows or feature irregular structures. Specifically, we introduce the position information of doors and windows and the room boundary information as additional attributes in the root node. These attributes can either be pre-specified according to the actual floor plan or automatically generated by the VLM based on user requirements. Each door or window is described by the following fields: "wall" (the wall on which it is located), "position" (the center point along that wall), "width", and "height". By incorporating these attributes, the VLM can take the physical structure of the room into account during the reasoning and reflection process, thereby producing more reasonable layouts. Furthermore, for rooms with irregular structures, we adopt simulated annealing for physical optimization, which avoids the need to explicitly model complex boundary constraints required by the Gurobi optimizer, thus flexibly accommodating irregular room contours. Figure 16 presents several generated results under both extended scenarios.

B.10 More Results

In Figure 17 and Figure 18, we provide additional visual comparison results of LayoutGPT, HOLODECK, LayoutVLM, HierScene, SceneWeaver, and our method.

C Additional Discussions

C.1 Differences from Agentic Scene Generation Frameworks

Recent agentic frameworks, such as SAGE [38], Scenethesis [36] and SceneSmith [39], also employ VLM agents with iterative refinement for scene generation. HierRR differs from these methods in both its technical focus and refinement mechanism.

Difference in technical focus. Existing methods primarily focus on the orchestration of external tools, whereas HierRR concentrates on hierarchical layout reasoning and reflection, aiming to understand multi-granularity spatial relations and explicitly ground them into the layout, thereby improving semantic consistency and layout plausibility.

Difference in refinement granularity and target. Existing methods typically evaluate the generated scene from a global perspective and directly modify the current scene or redesign the current stage. In contrast, HierRR adopts hierarchy-guided reflection, which refines the layout from local to global along the hierarchy. This enables precise localization of problematic nodes, thereby improving both reflection efficiency. Moreover, HierRR refines spatial relations and the reasoning process rather than directly editing the generated scene. Reflection results are fed back into ST-CoT to update the relational hierarchy, which then guides layout regeneration.

C.2 Analyzing ST-CoT and Strong Structured Prompting

Compared with strong structured prompts, which directly generate the layout from a structured scene representation, the key advantage of ST-CoT lies in explicitly modeling the step-by-step reasoning process from relational constraints to coordinate computation, rather than treating spatial reasoning as a one-shot parsing problem.

The ablation results in Table 2 also verify this advantage. Specifically, the first ablation directly generates layouts from graph/hierarchy representations through structured prompting, without ST-CoT or reflection, thereby serving as a strong structured-prompt baseline. The second ablation introduces only ST-CoT, while keeping all other components unchanged. It can be observed that, with the hierarchy representation fixed, replacing the strong structured-prompt baseline with ST-CoT alone improves OOR from 32.95% to 38.54% and OAR from 35.98% to 56.54%; a similar trend is observed under the graph representation (OOR from 19.48% to 20.92%, OAR from 35.05% to 51.87%). This indicates that the explicit step-by-step spatial reasoning of ST-CoT significantly improves layout quality, outperforming the way of directly generating layouts from structured prompts alone.

C.3 How to Handle Infeasible or Conflicting Spatial Constraints?

In principle, infeasible geometric solutions may arise in the fourth sub-step of ST-CoT (T4), particularly under highly dense or conflicting spatial constraints. In practice, however, such cases are uncommon in HierRR, for two reasons.

First, the hierarchy decomposes a complex scene into multiple local subproblems, which allows ST-CoT to reason over a much smaller set of spatial relations at each step. Most constraints are therefore resolved locally before being integrated into the global layout, substantially reducing geometric conflicts.

Second, reflection provides a closed-loop correction mechanism. When unreasonable layouts or inconsistent spatial relations are detected, reflection identifies the problematic nodes, updates the spatial relations in the hierarchy, and re-executes ST-CoT. This iterative reasoning-and-reflection process effectively resolves most conflicts caused by imperfect reasoning.

For intrinsically infeasible inputs, such as contradictory spatial relations or room-capacity violations, HierRR cannot guarantee a layout that satisfies all constraints. Instead, it prioritizes the primary spatial relations. In future work, we plan to incorporate more explicit geometric constraints into ST-CoT to provide stronger feasibility guarantees.

Prompt for Hierarchical Structure Generation

Task Description:
As an experienced layout designer, you need to analyze the user requirements and the scene space, then divide the space into multiple functional areas, and finally determine the objects and their placement.
Please note that each scene may have one or multiple functional areas designed for specific activities. Objects belonging to the same functional area usually have a close spatial relationship to complement their functionality. To simplify and facilitate the representation of spatial relationships within functional areas, each functional area can be divided into sub-areas as needed. After this division, a hierarchical structure representing the spatial relationships of the scene is formed. In this hierarchy, the root node is the scene, the intermediate nodes are functional areas/sub-areas, and the leaf nodes are the objects.

Your task is to identify possible objects based on user requirements. Further divide functional areas and sub functional areas as much as possible, ensuring that the object relationships present in each functional area/sub functional area are simple. Finally, design a hierarchical structure.

Output Format:
The hierarchical structure should be described in the following JSON format:
{'name': <scene name>, 'length': <scene length>, 'width': <scene width>, 'height': <scene height>,
'children':[
{'name': <area name> <id>, 'length': <area length>, 'width': <area width>, 'height': <area height>, 'relation position': <area relation position>,
'children':[
{'name': <obj name> <id>, 'length': <obj length>, 'width': <obj width>, 'height': <obj height>, 'description': <obj description>, 'relation position': <obj relation position>},
{'name': <sub area name> <id>,'relation position': <sub area relation position>, 'length': <sub area length>, 'width': <sub area width>, 'height': <sub area height>,
'children': [...]
},...
]},...
]}
Here are some guidelines for you:
1. To prevent confusion caused by multiple identical objects or area in the room, please indicate in <id> which of the current object or area is the first of the same object or area (starting from 1)
2. The <area relation position> describes the detailed spatial position of the current area in the scene.
3. The <sub area relation position> describes the detailed spatial position of the current sub area in the area corresponding to its parent node.
4. The <obj description> describes the detailed appearance of the object, <obj relationship position> describes the detailed spatial position of the object within the current area.

Example:
Here is an example representing the hierarchical structure: {EXAMPLE}

Please follow the desired format *strictly* to describe the hierarchical structure of the following user requirements.
User Requirement: {USER REQUIREMENT}

Figure 7: Prompt for hierarchical structure generation.

Prompt for Spatial Transformation Chain-of-Thought Reasoning (Task 1~3)

Task Description:
You are given information about the scene and areas/objects in the following format:
Scene State: <scene state>
Existing Layout: <existing layout>
Area/Object Spatial Relation Description: <position description>
<scene state>is the current state of the scene, rendered as a top-down image, where the positive x-axis represents the right side and the positive y-axis represents the front side.
<existing layout> is the information about existing areas/objects in the scene, and <position description> is the description of the position where the area/object will be placed.

Your task is to provide a more detailed description for the area/object relative position based on the <position description> and <existing layout> to infer the accurate position.

Suggestion:
{SUGGESTION AND WRONG REASONING} // *from reflection*

Reasoning Process:
Firstly, provide a reference coordinate system, which may be the entire scene, an area or a specific object.
Secondly, provide the relative position description of the current area/object along the x, y, and z axes in the reference coordinate system. If the reference coordinate system is an area or the entire scene, the origin is placed at the center of the area/scene . The maximum x, y, and z-coordinates correspond to the rightmost side, the forefront, and the ceiling of the area/scene, respectively. If the reference coordinate system is a specific object 'o', the origin is placed at the center of 'o'. The y-axis always faces in the same direction as object 'o'. The maximum x, y, and z-coordinates correspond to the rightmost side, the forefront, and the top of the area/scene, respectively.
In the reference coordinate system, providing the relative position description of an object/area consists of three steps:1) determine whether the object is located inside, on the surface, or outside of the reference object (e.g., a bed inside a bedroom, a book on a desk surface, a nightstand outside the bed); 2) determine its relative position along the x-, y- and z-axes with respect to the reference object, whether it is to the left, right, front, or exactly centered or below; 3) determine its distance from the reference object along the x, y, and z axes, whether it is match, slightly greater than or less than.
Finally, provide the orientation of the current area/object in the reference coordinate system. If the reference coordinate system is a scene/area, the available directions include x-axis positive direction, x-axis negative direction, y-axis positive direction, y-axis negative direction. If the reference coordinate system is an object, the available directions include facing the object (e.g., chairs facing a table), or consistent with the object (e.g., television in the same way as a television stand).

Output Format:
Please first think about the reasoning process in the mind and then provides the answer. The reasoning process is enclosed within <think></think>tags. The format of output is as follows:
<think> reasoning process </think>[<obj name>, <position description>, <reference coordinate system>, <x-axis relative position description>, <y-axis relative position description>, <z-axis relative position description>, <orientation description>]
Here are some guideline for you:

1. The output content must include <obj name>, <position description>, <reference coordinate system>, <x-axis relative position description>, <y-axis relative position description>, <z-axis relative position description>, <orientation description> all values. The format of these are strings.
2. Please provide the <x-axis relative position description>, <y-axis relative position description>, <z-axis relative position description>, and <orientation description> in one sentence.
Example:
Here is an example: {EXAMPLE}

Figure 8: Prompt for spatial transformation chain-of-thought layout reasoning (Task 1-3).

Prompt for Spatial Transformation Chain-of-Thought Reasoning (Task 4)

Task Description:
You are given information about the scene and objects in the following format:
Existing Layout: <existing layout>
New Area/Object: {'name': <name>, 'length': <length>, 'width': <width>, 'height': <height>, 'x-axis relative position description': <x-axis relative position description>, 'y-axis relative position description': <y-axis relative position description>, 'z-axis relative position description': <z-axis relative position description>, 'orientation description': <orientation description>}
Your task is to calculate the position and orientation of the new object on the x, y, and z axes based on the relative position description of the new object.

Calculation Rules:
Firstly, calculate the position in the x-axis based on the <x-axis relative position description>. For each object/area, determine its x-axis position using these three components: 1) <x result>: the numerical coordinate value; 2) <x note>: the reference point of the current object/area (e.g., max, min, or center); 3)<x relative object>: which object/area it is relative to.
Similarly, determine <y result>, <y note>, and <y related object>based on the <y-axis relative position description>, and determine <z result>, <z note>, and <z-relative object> based on the <z-axis relative position description>.
Then, determine the direction of the new object based on the <orientation description>. The <orientation> can be either 'X_negative', 'Y_negative', 'X_positive', 'Y_positive', '-<orientation_relative_object>'(using for facing <orientation_relative_object>) and '<orientation_relative_object>'(using for the orientation consistent with <orientation_relative_object>).

Output Format:
The format of the output content is as follows:
{'name': <name>, 'x_result': <x result>, 'x_note': <x note>, 'x_relative_object': <x relative object>, 'y_result': <y result>, 'y_note': <y note>, 'y_relative_object': <y relative object>, 'z_result': <z result>, 'z_note': <z note>, 'z_relative_object': <z relative object>, 'orientation': <orientation>}
Here are some guidelines for you:
1. The output content must include <new obj name>, <x result>, <x note>, <x relative object>, <y result>, <y_note>, <y relative object>, <z_result>, <z_note>, <z relative object>, <orientation> all values.
2. The <new obj name> is a string. The <x result>, <y result> or <z result> is a float.
3. The optional <x note>, <y note> or <z note> include 'center', 'min', or 'max'. The optional <x relative object>, <y relative object> or <z relative object> include 'scene' or <obj>. The optional <orientation> only include 'X_negative', 'Y_negative', 'X_positive', 'Y_positive', '-<orientation relative object>' or '<orientation relative object>'.
Example:
Here is an example: {EXAMPLE}

Figure 9: Prompt for spatial transformation chain-of-thought layout reasoning (Task 4).

Prompt for Fine-grained Sematic Extraction (Small Object)

Task Description:
You are given information about the scene and areas/objects in the following format:
Reference Image: <reference image>
In the <reference image>, the position of the general perspective is in front, and the position away from the perspective is in the back.
Your task is to describe the position of {OBJ} in the image and consider the relative position relationship between {OBJ} and {ANCHOR_OBJ}.

Processing Rules:
Firstly, provide an overall spatial position description of {OBJ} <spatial position description>.
Furthermore, consider the relative positional relationship between the {OBJ} and {ANCHOR_OBJ} in the front to back direction <front back relative position description>.
Then, consider the relative positional relationship between the {OBJ} and {ANCHOR_OBJ} in the left-right direction <left right relative position description>.
Finally, output the result.

Output Format:
Please first think about the reasoning process in the mind and then provides the answer. The reasoning process is enclosed within <think> </think>tags.
The format of output is as follows:
<think> reasoning process </think> [<spatial position description>, <front back relative position description>, <left right relative position description>]

Example:
Here is an example: {EXAMPLE}

Prompt for Fine-grained Sematic Extraction (Other Object)

Task Description:
You are given information about the scene and areas/objects in the following format:
User Requirement: <user requirement>
Existing Layout: <existing layout>
Area/Object Spatial Relation Description: <position description>
<scene state>is the current state of the scene, rendered as a top-down image, where the positive x-axis represents the right side and the positive y-axis represents the front side. The leftmost, rightmost, frontmost, and rearmost sides of the scene/area represent walls.
<existing layout> is the information about existing areas/objects in the scene, and <position description> is the description of the position where the area/object will be placed.

Your task is to check whether the<position description>is reasonable and accurate, supplement the<position description> based on the user requirement, and provide the <new position description>
Request a detailed and accurate description of the position of the object to be placed (preferably using words such as front, back, left, right, etc.) and the orientation of the object (preferably using words 'positive x-axis direction', 'positive y-axis direction', 'negative x-axis direction', 'negative y-axis direction', 'face to', 'consistent with').

Output Format:
Please first think about the reasoning process in the mind and then provides the answer. The reasoning process is enclosed within <think> </think>tags.
The format of output is as follows:
<think> reasoning process </think> <new position description>

Example:
Here is an example: {EXAMPLE}

Figure 10: Prompt for fine-grained sematic refinement.

Prompt for Vision Reflection (Score and Reason)

Task Description:
You are given a top-down rendering of the scene state, information about the current object within the scene, and the user's requirements in the following format:
Scene State: <scene state>
Existing Layout: <existing layout>
User Requirement: <user requirement>
<scene state>is the current state of the scene, rendered as a top-down image, where the positive x-axis represents the right/east side and the positive y-axis represents the front/north side. The boundary of the floor represents the boundary of the scene, which is the wall.

Your task is to evaluate how well the current scene state matches the user's requirements based on the two criteria listed below. For each criterion, please give a score from 0 to 10 and briefly explain your rationale. Scoring must be rigorous.

Evaluation Criteria:
There are some criteria for judgment as follows:
1. Placement: Is the distribution of objects reasonable and consistent with user preferences?
• Good (8-10): Each object is a reasonable size and neatly arranged. The distribution and relative positions of the objects are consistent with user preferences. The large objects are placed against the wall, such as sofas, TV cabinets, beds.
• Poor (0-3): Objects are unevenly distributed, inconsistent with user preferences (for example, a user prefers a painting against the wall above the bed, but the painting is only suspended above the bed, not close to the headboard or the boundary of the scene).
• Note: If there is a missing object, the score should not exceed 4. If objects are cluttered, large objects are not placed against the wall, bedside tables are not placed close to the bed, or hanging objects are not close to the scene boundaries, the score should not exceed 5. If the objects do not conform to the daily arrangement rules, even if they meet the user's preferences, the rating should not exceed 6 points.
2. Orientation: Is the orientation of the object reasonable? Usually, objects against the wall should point towards the interior of the room.
• Good (8-10): Each object has a reasonable orientation (for example, chairs face desks, sofa face tv, the direction of the bedside table is consistent with the bed), and all objects against the wall face the interior of the room (for example, the bookshelf is close to the north wall of the scene, and its direction faces south (negative y-axis)).
• Poor (0-3): The orientation of the objects is confusing, and all objects against the wall face the exterior of the room. For example, the bookshelf is close to the left wall of the scene, and its direction still faces the left (negative x-axis). The direction of the chair is consistent with the table.
• Note： If the user wants an object to be close to a corner of the scene, its orientation can be in two situations. For example, if the wardrobe is located in the upper right corner of the scene, its orientation can be to the left (negative x-axis) or to the bottom (negative y-axis).

Output Format:
Please analyze the scene status before giving a rating. The output format is as follows:
{
"Placement": {"grade": your grade as int, "comment": "your comment and reasons for the score."},
"Orientation": {"grade": your grade as int, "comment": "your comment and reasons for the score."}
}

Figure 11: Prompt for vision reflection to obtain scores and reasons.

Prompt for Vision Reflection (Suggestion)

Task Description:
Please provide modification suggestions for the layout based on the following information:
Scene State: <scene state>
Existing Layout: <existing layout>
<scene state>is the current state of the scene, rendered as a top-down image, where the positive x-axis represents the right/east side and the positive y-axis represents the front/north side. The boundary of the floor represents the boundary of the scene, which is the wall.
<existing layout> is the information about existing areas/objects in the scene.

You have identified the following unreasonable situation in the current scene:
{PROBLEM}
To solve this problem, your task is:
1. Identify which objects need to be adjusted to improve the scene layout. For each object that requires modification, only provide the most direct and effective suggestions from the following two aspects (not all are required):
• Move Object Location: If the object is out of bounds, not against the wall, missing or its spatial relationship with user requirements is inconsistent, consider moving the object. If you want to move the object, please describe in detail where the object should be moved, such as closer to a wall/corner, or to the right/left/in front of/behind another object. Note: This suggestion requires consideration of available space and guarantees that the object will not overlap or collide with other objects after the move.
• Change Object Orientation: If the object's orientation is unreasonable, specify the direction it should be adjusted to (i.e., positive/negative x-axis, positive/negative y-axis, face to other objects or consistent with other objects).
2. Only include the items that require modification in your output, with clear and specific descriptions. Parts that are not involved can be omitted.

Wrong Suggestion in the Preliminary Reflections:
Do not give the following suggestion:{WRONG SUGGESTION}

Output Format:
The output format is as follows:

```
{
  <object name>: {
  "Move Object Location": "Your suggestion.",
  "Change Object Orientation": "Your suggestion.",
  },
      <object name>: {
      "Change Object Orientation": "Your suggestion.",
       },
      <object name>: {
       "Move Object Location": "Your suggestion.",
       },...
   }
```

Example:
Here is an example: {EXAMPLE}

Figure 12: Prompt for vision reflection to obtain suggestions.

Prompt for Scene Realism Evaluation

Task Description:
You are provided with the following information: a rendered image of the scene and the user's requirements, in this format:
Rendered Image : <rendered image>
User Requirement: <user requirement>
<rendered image> is presented in a front view. The boundary of the floor represents the boundary of the scene, which is the wall.

Your task is to evaluate how well the scene state meets two criteria listed below. For each criterion, please assign a score from 1 to 5 and briefly explain your rationale.

Evaluation Criteria:
1. Layout Rationality: Evaluate whether the placement of objects is realistic and reasonable.
• Excellent(5): Objects are neatly arranged with correct positions and orientations, fully conforming to daily life habits.
• Good(4): Objects are neatly arranged, but their positions and orientations are slightly unreasonable. • Fair(3): The arrangement is moderately tidy and shows some logic, but the position or orientation of certain objects is noticeably unreasonableness.
• Poor(2): Objects are placed in a rather cluttered manner, with multiple obvious issues in position and orientation.
• Bad(1): Objects are chaotically placed with illogical positions and orientations, resembling random distribution.

2. Scene Functionality: Evaluate the richness of objects in the scene, that is, whether the categories and quantity of objects can meet daily life needs.
• Excellent(5): The scene is rich in objects with strong logical relationships, containing all necessary furniture and items, fully meeting daily living needs.
• Good(4): The scene is rich in objects, lacking only a few non-essential small items, which does not affect daily use.
• Fair(3): The scene lacks sufficient object variety and is missing some items.
• Poor(2): The scene has insufficient objects and lacks a few essential items, affecting daily use.
• Bad(1): The scene suffers from a severe shortage of objects and lacks a large number of essential items, failing to meet daily living needs.
Please note that "Layout Rationality" assesses only the placement and orientation of objects, while "Scene Functionality" assesses only the categories and quantities of objects. The evaluation of these two metrics does not involve visual dimensions such as appearance, material, or style.

Output Format:
Please carefully analyze the scene status and score strictly according to the corresponding criteria. The output format is as follows:
{
"Layout Rationality": {"grade": your grade as int, "comment": "your comment and suggestion."},
"Scene Functionality": {"grade": your grade as int, "comment": "your comment and suggestion."},
}

Figure 13: Prompt for scene realism evaluation.

User Evaluation Survey on Scene Layout Generation

Hello! We are conducting a survey on indoor scene layout generation to evaluate the quality of layouts generated by different algorithms. In this survey, you will be presented with 36 scene results generated based on 6 different sets of user requirements. To ensure fair evaluation, all results have been shuffled. Please provide your objective ratings based on your intuitive perception.

For each result, the following references are provided:
1. User Requirements: Describes the user's requirements for the room (e.g., room type, object categories, quantities, and spatial relationships).
2. Top-view Diagram: Provides a clear view of the number, positions, and orientations of objects.
3. Rendered Images: Provides an intuitive 3D visual reference for spatial atmosphere.

Please rate each result across the following four dimensions. (using a 1-5 scale, where higher scores indicate better performance):

1. Semantic Alignment: Evaluates how well the generated layout aligns with the user's textual description.
- Excellent-5: Perfect match. Object categories, quantities, relative positions, and orientations are all highly consistent with the text description.
- Good-4: Basic match. Object categories and quantities match the text description, but there are slight inconsistencies in relative positions/orientations.
- Fair-3: Partial match. Some objects are missing or incorrect, with certain deviations in positions/orientations, but the overall layout remains recognizable.
- Poor-2: Low match. Many objects are missing or incorrect, and the layout significantly deviates from the text description.
- Bad-1: No match. The layout is almost completely irrelevant to the text description.

2. Physical Feasibility: Evaluates the feasibility of the object arrangement in the real world, considering whether there are collisions or out-of-bounds objects, and whether there is sufficient space for daily activities.
- Excellent-5: No overlapping or out-of-bounds objects. Space is sufficient for convenient daily activities.
- Good-4: No overlapping or out-of-bounds objects in the scene, but some items are too close together, which may affect daily activities.
- Fair-3: Slight overlapping or out-of-bounds exists, but no objects are out-of-bounds.
- Poor-2: Noticeable overlapping or out-of-bounds issues are present, affecting normal use.
- Bad-1: Severe overlapping and out-of-bounds problems exist; the layout is chaotic and physically infeasible.

3. Layout Rationality: Evaluates whether the placement of objects is realistic and reasonable.
- Excellent-5: Objects are neatly arranged with correct positions and orientations, fully conforming to daily life habits.
- Good-4: Objects are neatly arranged, but their positions and orientations are slightly unreasonable.
- Fair-3: The arrangement is moderately tidy and shows some logic, but the position or orientation of certain objects is noticeably unreasonableness.
- Poor-2: Objects are placed in a rather cluttered manner, with multiple obvious issues in position and orientation.
- Bad-1: Objects are chaotically placed with illogical positions and orientations, resembling random distribution.

4. Scene Functionality: Evaluate the richness of objects in the scene, that is, whether the categories and quantity of objects can meet daily life needs.
- Excellent-5: The scene is rich in objects with strong logical relationships, containing all necessary furniture and items, fully meeting daily living needs.
- Good-4: The scene is rich in objects, lacking only a few non-essential small items, which does not affect daily use.
- Fair-3: The scene lacks sufficient object variety and is missing some items.
- Poor-2: The scene has insufficient objects and lacks a few essential items, affecting daily use.
- Bad-1: The scene suffers from a severe shortage of objects and lacks a large number of essential items, failing to meet daily living needs.

Figure 14: User interface of perceptual study.

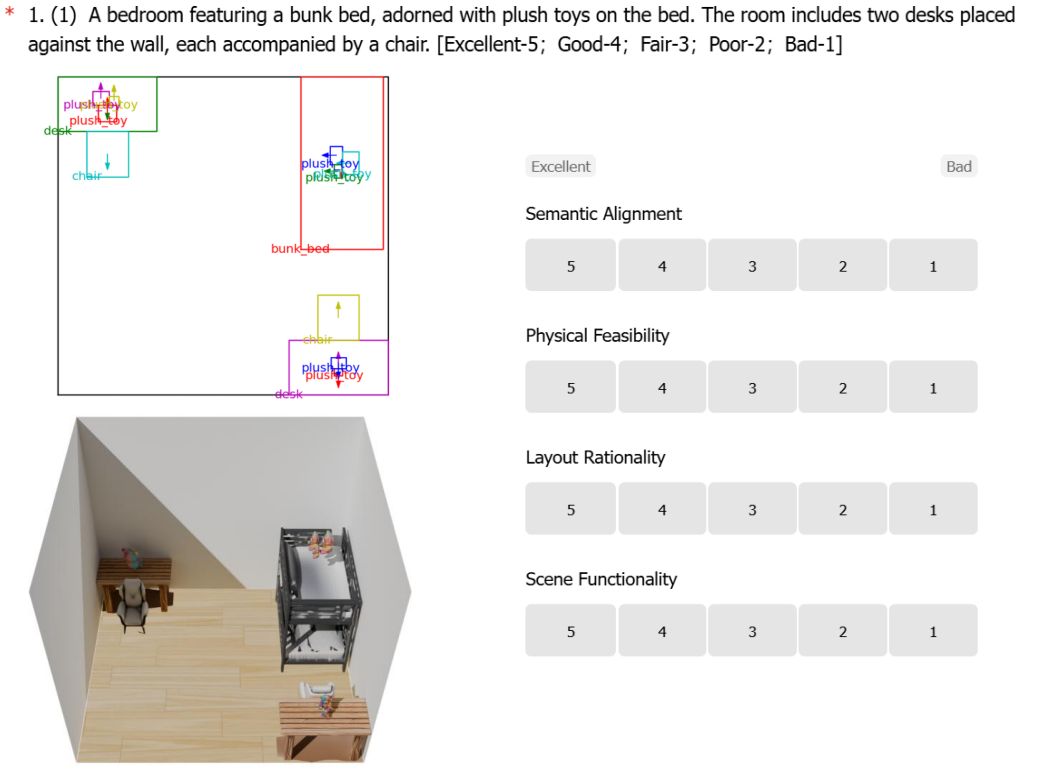

Figure 15: User interface of perceptual study.

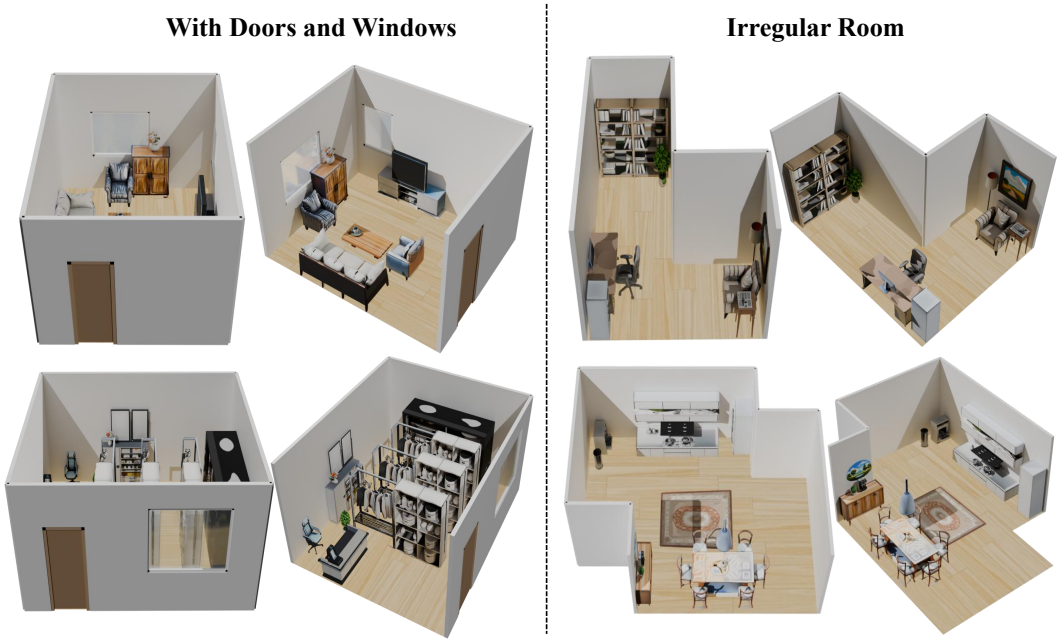

Figure 16: Results in rooms with doors, windows, and irregular structures.

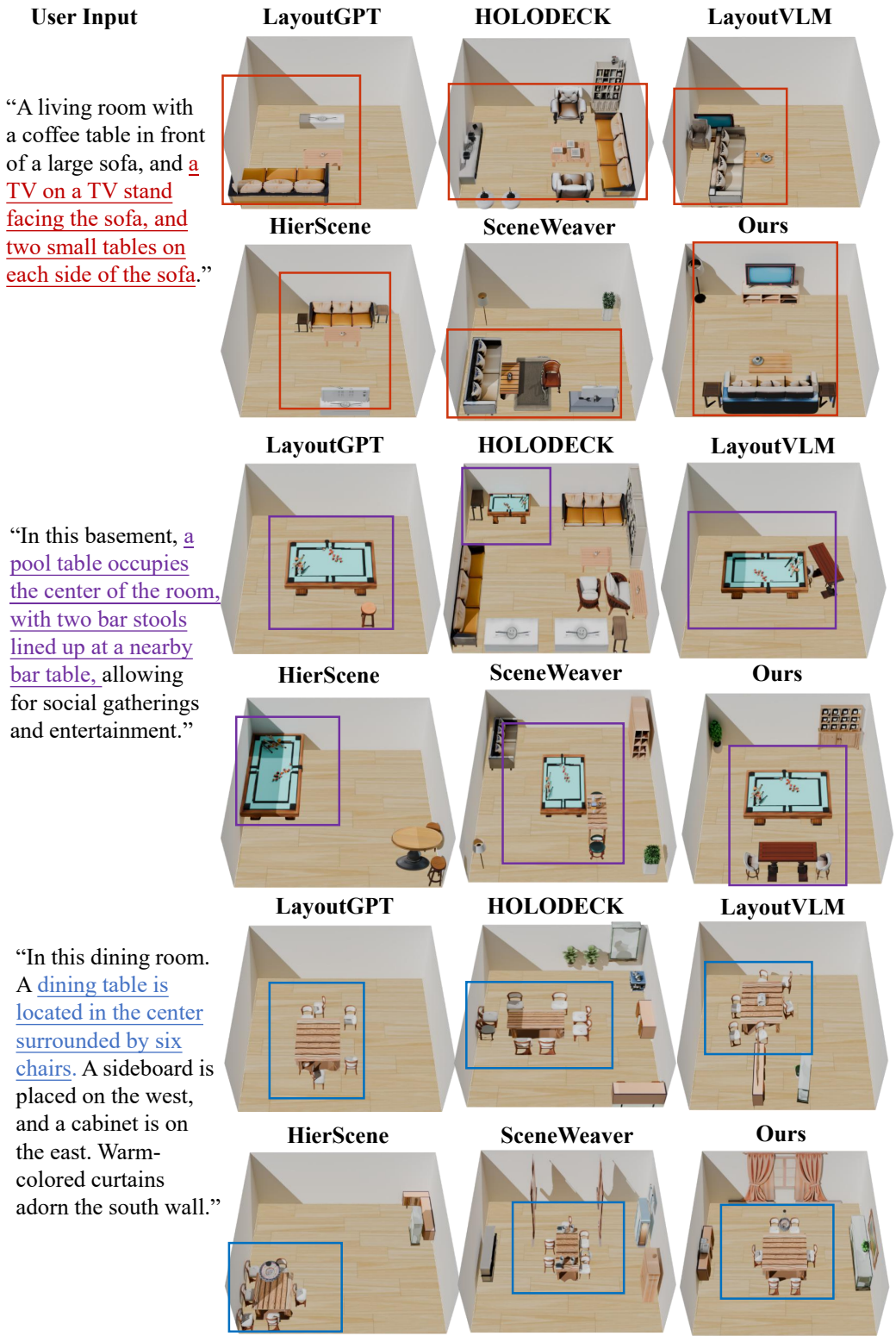

Figure 17: More qualitative comparison results generated from different methods.

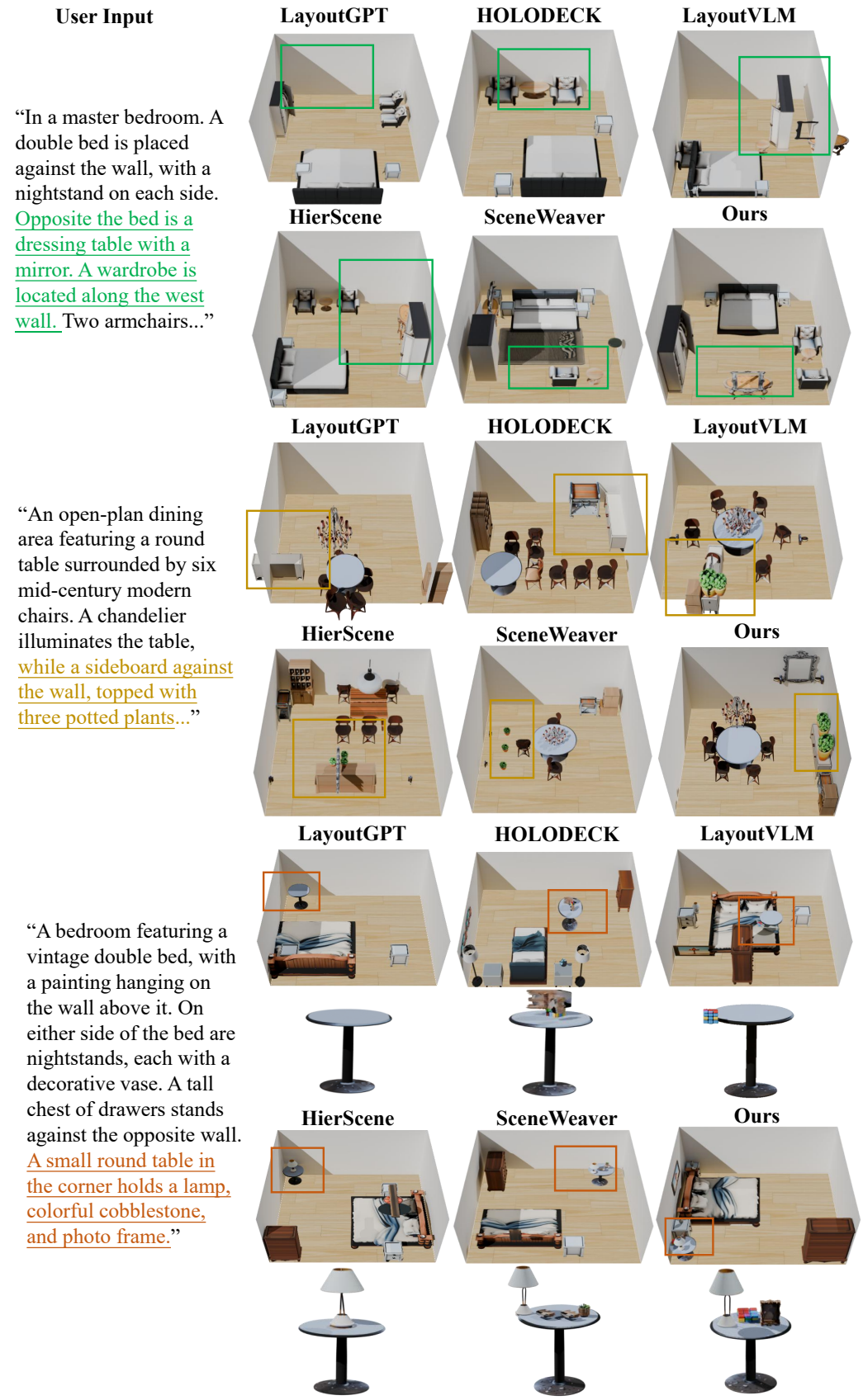

Figure 18: More qualitative comparison results generated from different methods.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The research problem and the main contributions of this study are clearly articulated in the abstract and introduction sections.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The conclusion section include a discussion of the work's limitations.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: This paper does not involve theoretical assumptions.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: This paper provides a detailed description of the experimental setup to ensure reproducibility.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer:[Yes]

Justification: This code is provided as supplementary material.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The experimental setup is clearly detailed in both the main experimental section and appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: This paper reports the mean of results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The computational resources utilized are described in the experimental implementation details section and appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This work fully adheres to the NeurIPS Code of Ethics in all aspects.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The introduction highlights the importance of the indoor scene synthesis task.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper uses pre-trained language models, but they do not pose such misuse risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All relevant works are properly cited, and all open-source assets are used in accordance with their licensing terms.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The anonymized code implemented in this work is included in the supplementary materials.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Justification: Details of the human perceptual study, including the complete instructions and screenshots, are available in both the main paper and appendix.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper only involves the harmless human perceptual study with no potential risks to participants, and thus IRB approval is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: This paper adopts LLMs as the base model for the core methods, and provides a clear and detailed description of their usage.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.