Hierarchically-Structured Open-Vocabulary Indoor Scene Synthesis with Pre-trained Large Language Model

Weilin Sun[1], Xinran Li[1], Manyi Li[1]*, Kai Xu[2]*, Xiangxu Meng[1], Lei Meng[1, 3]*

[1]School of Software, Shandong University, China
[2] School of Computer Science, National University of Defense Technology, China
[3] Shandong Research Institute of Industrial Technology, China
{sunweilin, xinranli}@mail.sdu.edu.cn, manyili@sdu.edu.cn, kevin.kai.xu@gmail.com, {mxx, lmeng}@sdu.edu.cn,

Abstract

Indoor scene synthesis aims to automatically produce plausible, realistic and diverse 3D indoor scenes, especially given arbitrary user requirements. Recently, the promising generalization ability of pre-trained large language models (LLM) assist in open-vocabulary indoor scene synthesis. However, the challenge lies in converting the LLM-generated outputs into reasonable and physically feasible scene layouts. In this paper, we propose to generate hierarchically structured scene descriptions with LLM and then compute the scene layouts. Specifically, we train a hierarchy-aware network to infer the fine-grained relative positions between objects and design a divide-and-conquer optimization to solve for scene layouts. The advantages of using hierarchically structured scene representation are two-fold. First, the hierarchical structure provides a rough grounding for object arrangement, which alleviates contradictory placements with dense relations and enhances the generalization ability of the network to infer fine-grained placements. Second, it naturally supports the divide-and-conquer optimization, by first arranging the sub-scenes and then the entire scene, to more effectively solve for a feasible layout. We conduct extensive comparison experiments and ablation studies with both qualitative and quantitative evaluations to validate the effectiveness of our key designs with the hierarchically structured scene representation. Our approach can generate more reasonable scene layouts while better aligned with the user requirements and LLM descriptions. We also present open-vocabulary scene synthesis and interactive scene design results to show the strength of our approach in the applications.

1 Introduction

Indoor scene design requires a comprehensive consideration of space partition, functional arrangement, and aesthetic creativity to determine the object selection and placement to form the scene layout. There has been a vast amount of research on indoor scene synthesis, ranging from layout optimization (Yu et al. 2011; Merrell et al. 2011; Qi et al. 2018) to various conditional scene synthesis (Wang et al. 2018; Paschalidou et al. 2021; Gao et al. 2023; Tang et al. 2023). The goal is to automatically produce plausible, realistic, and diverse 3D indoor scenes, especially given arbitrary user requirements. However, due to the complexity of indoor scenes, most of them are limited within the scope of training data and cannot be generalized to arbitrary conditions.

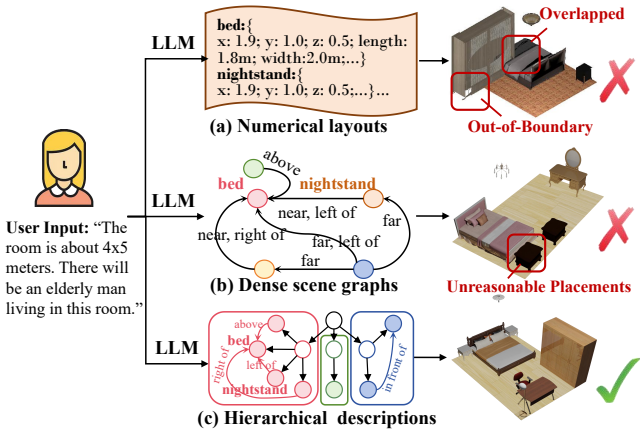

Figure 1: Compared to LLM-generated (a) numerical layouts and (b) scene graphs with dense relations, we use (c) LLM-generated hierarchical scene descriptions, whose internal nodes represent functional areas with compact and generalizable prior, to generate more reasonable and physically feasible scene layouts aligned with the descriptions.

Some pioneer works (Feng et al. 2023; Fu et al. 2024; Yang et al. 2024b) make use of the promising generalization ability of pre-trained large language models (LLM) to address the open-vocabulary scene synthesis task, where the LLM is responsible for interpreting any textual requirement into detailed scene configurations. The challenge lies in obtaining reasonable and physically feasible scene layouts from LLM outputs. Directly using LLM to output numerical layouts (Feng et al. 2023) causes unreliable results with heavy object overlap and out-of-boundary, since LLM fails to understand the spatial relationship with numerical layouts. On the other hand, LLM shows good performance in generating detailed text descriptions of various scenes, but still require an approach to convert the textual descriptions into numerical layouts while maintaining the generalization of the entire pipeline. The existing methods (Fu et al. 2024; Yang et al. 2024b) have to pre-define textual phrases and numerical rules for several types of spatial relations to obtain

*Corresponding Author.

the layouts. However, dense spatial relations often lead to incompatible object arrangements while coarse relations fail to capture diverse spatial placements, causing misalignment between LLM-generated configuration and the results.

In this paper, we propose to use hierarchical scene descriptions as the intermediate representation in the LLM-assisted scene synthesis pipeline. The hierarchical structure has three levels, with the entire scene as root node, functional areas as internal nodes, and objects as leaf nodes, as illustrated in Figure 2. Our approach contains three stages. First, we prompt the pre-trained LLM to generate the hierarchical structure with text descriptions. Second, we train a hierarchy-aware network to further infer the fine-grained relative placements between objects with textual spatial relations. Taking the hierarchical structure as grounding, the network can infer reasonable relative placements in an open-vocabulary setting. Third, we develop a divide-and-conquer optimization, which optimizes each functional area separately and then arranges them to form the entire scene, to solve for the physically feasible scene layouts effectively.

The advantages of using hierarchically structured scene representation are two-fold. First, the hierarchical structure provides a rough grounding for object arrangement, which alleviates contradictory placements with dense relations and enhances the generalization ability of the network to infer fine-grained placements. Second, it naturally supports the divide-and-conquer optimization to more effectively solve for a feasible layout that matches with the LLM-generated descriptions. We perform extensive comparison experiments and ablation studies with both qualitative and quantitative evaluations. Our approach generates more reasonable and physically feasible scenes that align better with the user requirements and LLM arrangements. In addition, we show the results of open-vocabulary scene synthesis and interactive scene design as practical applications of our approach.

Our contributions are summarized as follows:

- We propose an LLM-assisted hierarchically-structured scene synthesis pipeline, which uses a three-level hierarchical structure to infer reasonable object arrangements.
- We develop the hierarchy-aware network and the divide-and-conquer optimization, which takes advantage of the hierarchical structure for effective layout synthesis.
- We conducted extensive comparison and ablation study experiments to validate the effectiveness of our approach, as well as two applications to show its practical usage.

2 Related Work

Indoor Scene Synthesis. The common practice is to produce a set of objects and their placements (Patil et al. 2023), i.e. scene layouts. Early works rely on the pre-defined rules (Yu et al. 2011; Merrell et al. 2011; Ma et al. 2016; Fu et al. 2017, 2020) to generate interpretable and feasible scene layouts. To further capture diverse spatial arrangement, the data-driven approaches (Fisher et al. 2012; Qi et al. 2018; Xu et al. 2014; Ma et al. 2018b; Sun et al. 2022, 2024b) learn the object relationship from datasets (Fu et al. 2021; Song et al. 2017). The researchers have developed all kinds of networks to learn the scenes represented as

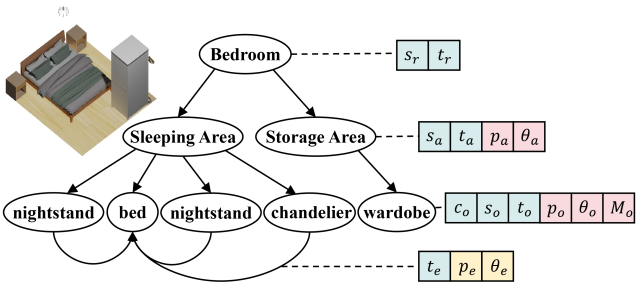

Figure 2: Our three-level hierarchical scene structure with the functional area as internal nodes.

different data structures, including sequences (Wang et al. 2018), graphs (Zhou, While, and Kalogerakis 2019; Wang et al. 2019), hierarchies (Li et al. 2019; Gao et al. 2023), sets (Paschalidou et al. 2021; Wei et al. 2023; Tang et al. 2023; Zhai et al. 2024a), etc. However, due to the inherent complexity, it is difficult to capture the essential relationship from the observed layouts and generalize to other categories.

Incorporating additional knowledge enhances scene prior learning. Graph-to-3D (Dhamo et al. 2021) and CommonScenes (Zhai et al. 2024b) synthesize the layouts and object shapes for coherent scenes. Some methods (Ye et al. 2022; Yi et al. 2023) take human motion trajectory as conditions to populate the objects. Haisor (Sun et al. 2024a) uses reinforcement learning with human interaction and space area consideration for scene synthesis. External expert knowledge (Leimer et al. 2022; Yang et al. 2024a) can also be incorporated during training to enhance network performance. These works refer to the same observation that indoor scene synthesis involves a comprehensive consideration of space partition, functional arrangement, and aesthetic creativity, thus requiring a generative model with extensive knowledge.

Text-to-Scene Synthesis. The challenges include semantic understanding of user requirements and scene synthesis. Given a natural language description, early works (Chang, Savva, and Manning 2014; Chang et al. 2015, 2017; Savva, Chang, and Agrawala 2017; Ma et al. 2018a; Yang, Hu, and Ye 2021) parse the input as scene templates, where the nodes represent objects and edges for spatial relations, and then sample the corresponding object models and placements. With the development of deep learning, some works (Paschalidou et al. 2021; Tang et al. 2023; An et al. 2023) take latent vectors such as textual embeddings as conditions, and train conditional scene synthesis model to output scenes. However, these works rely on detailed descriptions as input to specify the setting of target scenes, e.g. "there is a desk and there is a notepad on the desk", rather than reasoning the scene configuration from abstract instructions. Moreover, generalizing to diverse scene categories and open-vocabulary settings is a long-standing problem.

LLM-Assisted Scene Synthesis. Recent works have investigated utilizing the pre-trained LLMs to handle the multi-objects in the scenes, most of which focus on the 2D layout for controllable scene image synthesis (Lian et al. 2023; Gani et al. 2023; Nie et al. 2024). LayoutGPT (Feng et al.

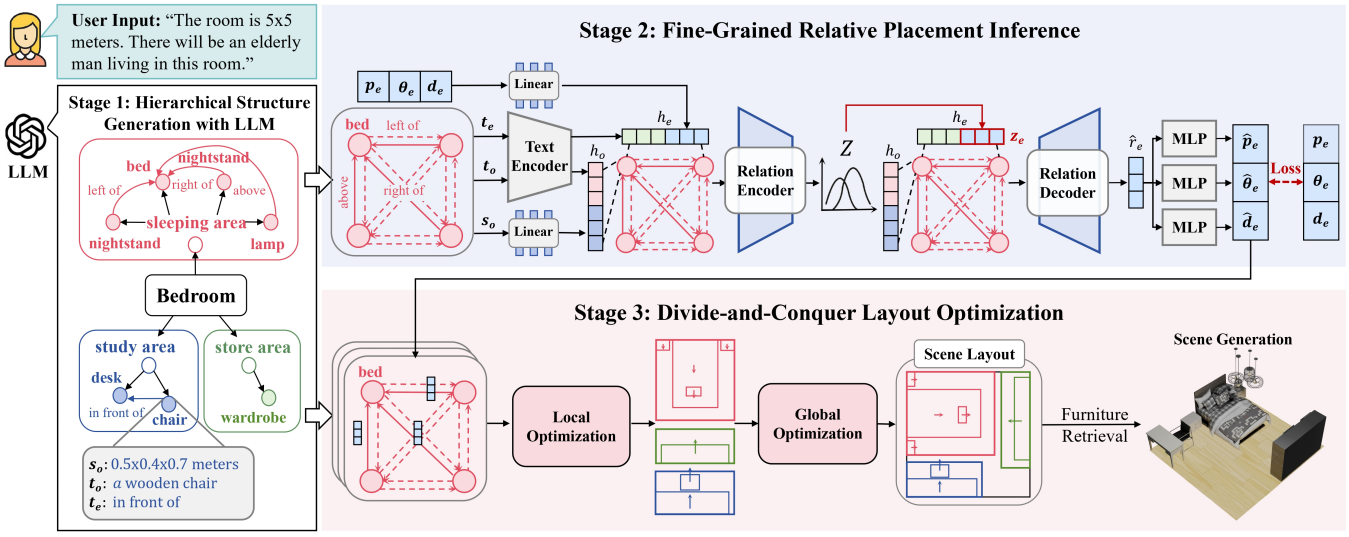

Figure 3: Our hierarchically-structured scene synthesis pipeline involves three stages. First, given a user requirement, we prompt the LLM to generate a hierarchical structure with text descriptions. Second, we train a hierarchy-aware network to infer the fine-grained relative placements between objects. Third, we use a divide-and-conquer optimization algorithm to arrange each functional area separately and then organize them into the entire scene. z_e indicates the random sampling from the Gaussian distribution Z, and $\hat{r}_e$ indicates the relative position information after decoding.

2023) retrieves scene layouts in CSS format, with LLMs outputting numerical bounding boxes for each object. However, due to the lack of spatial reasoning ability, LLM cannot handle the complex relationships of 3D scenes, causing heavy object overlap and out-of-boundary problems.

Some others use LLM to generate a textual scene description and convert it to 3D scenes. Aladdin (Huang et al. 2023) introduced a pipeline to sample and generate 3D textured assets from an abstract description and manually organize them to construct a scene. Recent works (Fu et al. 2024; Yang et al. 2024b) require LLM to describe the object relations using pre-defined atom relations, which are interpreted as fixed relative positions between objects and refined using a rule-based optimization algorithm. However, our experiments show that defining a compact yet informative atom relations is difficult. Dense and detailed object relations are able to provide precise spatial arrangement but often cause self-contradiction in the LLM outputs, while a sparse set of coarse relations leads to coherent arrangements but fails to capture the diverse spatial placements among objects.

3 Problem Formulation

Problem. Given an arbitrary textual description t as the input condition, e.g. "a small bedroom for a young student", our goal is to generate plausible, realistic, and physically feasible scenes corresponding to the requirement. The generated scene includes the selected 3D models and the arranged placements of the objects therein, i.e. $S = \{(M_i, P_i)|i = 1, ..., n\}$, where M_i and P_i denote the selected 3D model and spatial placement respectively for the ith object. The placement of each object contains its center position coordinates, orientation angle, and size.

Hierarchical Scene Representation. We propose to use the hierarchical scene representation throughout our pipeline. It is a three-level hierarchical structure, as shown in Figure 2. The first level is the root node representing the entire scene, the second level is the internal nodes each representing a rectangular functional area, and the third level is the leaf nodes representing the objects belonging to the corresponding area. The nodes are connected with two types of edges, i.e. the parent-child relation indicating the hierarchical structure and the pairwise relation between objects to represent their spatial relationship. Specifically, to reduce the redundancy, we set one anchor object for each functional area and only allow the pairwise relations between the anchor object and other objects belonging to the same functional area.

Each node in the hierarchy contains some attributes. Assuming axis-aligned rectangular floorplans for the scenes, the root node r has a size attribute s_r and a text description t_r of the scene, i.e. $r = \{t_r, s_r\}$. Each internal node a, which represents an axis-aligned functional area, carries the text description t_a, the size attributes s_a, as well as a center position p_a and an orientation θ_a, i.e. $a = \{t_a, s_a, p_a, \theta_a\}$. The position is a 2D coordinate while the orientation is a binary value representing either horizontal or vertical direction. Each object node o contain the text description t_o, the category label c_o, the corresponding 3D model M_o, as well as the size s_o, center position p_o, orientation θ_o of the oriented bounding box of the object, i.e. $o = \{t_o, c_o, M_o, s_o, p_o, \theta_o\}$. Note that the size attributes are 2D vectors for room and functional areas, but 3D vectors for objects, since we also care about their heights. In addition, the pairwise spatial relationship e stores the coarse text description t_e such as "in front of" and the fine-grained relative placement coordinates including position p_e and orientation θ_e of one object w.r.t. the anchor object, i.e. $e = \{t_e, p_e, \theta_e\}$.

4 Hierarchical Scene Synthesis with LLM

Given a text description t_r and the scene size s_r as conditions, our approach is composed of three stages, as illustrated in Figure 3. First, we prompt the pre-trained LLM to generate the hierarchical structure with text descriptions. Second, we train a hierarchy-aware graph neural network to infer the relative placement coordinates between objects. Third, we design a divide-and-conquer optimization which optimizes the sub-layout for each functional area and then arranges their placements to form the entire scene.

4.1 Hierarchical Structure Generation with LLM

Given the user requirement, the pre-trained LLM takes the constructed prompt as input and outputs structured text to describe the hierarchical scene representation, including the node attributes. The key challenge is to generate reasonable and informative spatial relations to specify the scene layout.

Although existing works define dense object relations to describe layouts, the more detailed the descriptions are, the more incorrect or self-contradictory results they make, due to the lack of spatial reasoning ability of the LLM. Therefore, in our approach, we require the LLM to generate a hierarchical structure to ground the objects and only the spatial relations between objects belonging to the same area, only to roughly specify their arrangements.

We construct the input prompt with three components: 1) a description of the LLM's role and task, including a brief definition of the hierarchical structure with the meaning of the nodes and connections; 2) a description of the preferred data format and pre-defined constraints, including the types of functional areas, possible anchor objects, and spatial relations; and 3) an example of a simple scene in the preferred format and the specific user requirements. We don't require the example to be selected corresponding to the user requirement, but only to demonstrate the output format. In this stage, the LLM generates textual descriptions and size attributes of the functional areas and objects, as well as the textual descriptions of spatial relations.

4.2 Fine-Grained Relative Placement Inference

We propose a hierarchy-aware graph neural network to infer the fine-grained relative placements between correlated objects. The relative placements within each functional area exhibit a more compact and generalizable prior, allowing us to train a network to infer the placements for various scenes.

As illustrated in Figure 3, given the LLM-generated hierarchy, we construct the input graph $G = (O, E)$ with the nodes as objects and edges connecting all objects belonging to the same functional area. Although the input includes all objects in the scene, the functional areas are isolated from each other. We use Linear embeddings for the object sizes s_o and the ground truth relative placement coordinates $[p_e, \theta_e, d_e]$, where d_e is a binary indicator of the alignment between two objects, and the pre-trained CLIP text encoder (Radford et al. 2021) for descriptions of objects t_o and spatial relations t_e. They are organized as node features h_o and edge features h_e, i.e.

$$\begin{aligned} h_o &= [\mathrm{CLIP_t}(t_o), \mathrm{LINEAR}(\mathrm{s_o})], \\ h_e &= [\mathrm{CLIP_t}(t_e), \mathrm{LINEAR}(p_e, \theta_e, d_e)], \end{aligned} \tag{1}$$

which forms the contextual graph for the following network processing. Note that since we only have textual spatial relations between the anchor object and the others, we use all-zero vectors as the text embeddings for the edges without corresponding textual spatial relations (dotted arrows).

We adopt the variational graph neural network (Zhai et al. 2024b) for the contextual graph with the h_o and h_e. Both the encoder and decoder are composed of several MLPs for 5 rounds of message passing, including $g_e^{(k)}$ for updating the edge features with connected node features in the kth round and $g_o^{(k)}$ for updating the node features with the 1-ring neighbor nodes, i.e.

$$\begin{aligned} h_{e_i \to j}^{(k+1)} &= g_e^{(k)}(h_{o_i}^{(k)}, h_{e_i \to j}^{(k)}, h_{o_j}^{(k)}) \\ h_{o_i}^{(k+1)} &= h_{o_i}^{(k)} + g_o^{(k)}(\mathrm{AVG}(h_{o_j}^{(k)} | o_j \in N_{\mathcal{G}}(o_i))), \end{aligned} \tag{2}$$

where $e_{i \to j}$ represents an edge connecting two objects o_i and o_j, $N_{\mathcal{G}}(o_i)$ represents the set of neighbor nodes connected with object o_i. The encoder takes the contextual graph as input and outputs the graph with updated features, where the edge features (specifically the relative placement components of edge features, as shown in Figure 3) are parameterized as a Gaussian distribution. The decoder takes the updated graph as input and randomly samples from the Gaussian distribution. Finally, we use separate MLPs to decode the relative placement $[\hat{p}_e, \hat{\theta}_e, \hat{d}_e]$.

During training, we freeze the CLIP text encoder and update all other network layers. The loss function is

$$L = L_{KL} + L_{ep} + L_{e\theta} + L_{ed}, \tag{3}$$

where L_{KL} is the Kullback-Liebler divergence between the Gaussian distribution and posterior distribution of the edge feature components. L_{ep} is L1 loss on the relative positions p_e. $L_{e\theta}$ and L_{ed} are cross-entropy loss on the discretized relative orientation angles and the binary alignment indicator.

4.3 Divide-and-Conquer Layout Optimization

Given the hierarchical scene with the relative placements between correlated objects, we develop a divide-and-conquer optimization to solve for the final layout. Our solution includes a local optimization for each functional area and then a global optimization to organize the areas into scenes. This optimization produces reasonable and physically feasible layouts more effectively than a simple global optimization or iteratively optimizing each object's placements.

Local optimization. For each functional area, we use local optimization to solve the object placements w.r.t. the bounding box of the functional area. The local optimization is formulated to minimize the objects' relative placements and those inferred by the network, with constraints to avoid object overlap and out-of-boundary, i.e.

$$\begin{aligned} &\min_{o'_i \in O} \sum_{o_i \in N_{\mathcal{G}}(o_a)} |\mathrm{REL}(o'_i, o'_a) - [p_{e_i \to a}, \theta_{e_i \to a}]|, \\ &s.t. \quad C_{overlap}(o'_i, o'_j), \quad \forall o_i, o_j \in A \\ &\qquad C_{OOB}(o'_i, s_a), \quad \forall o_i \in A \end{aligned} \tag{4}$$

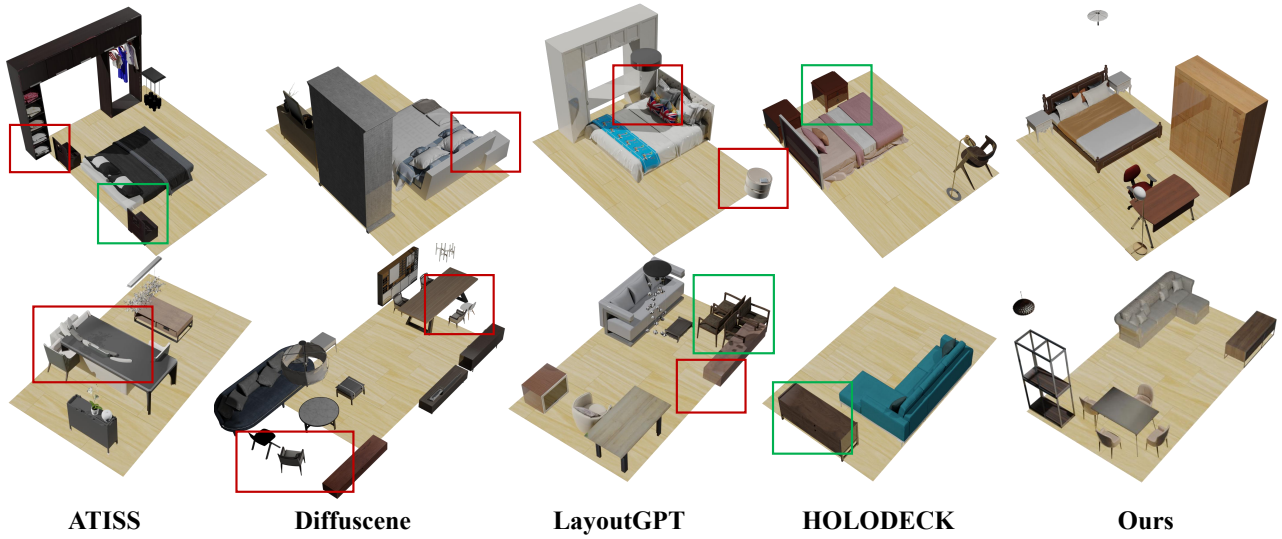

Figure 4: The scenes generated from different approaches. The end-to-end data-driven approaches often produce infeasible object placements including overlap and out-of-boundary cases (red boxes). On the other hand, HOLODECK sometimes produces unreasonable results such as two nightstands on the same side of the bed (top row) or a tv stand on the left side of the sofa (second row). By contrast, our approach is able to more effectively produce reasonable and physically feasible scene layouts.

where o_i' and o_a' refers to the placements (center positions and orientations) w.r.t. the functional area of an object o_i and the anchor object o_a, respectively. REL computes the relative placements between two objects and $[p_{e_i \to a}, \theta_{e_i \to a}]$ is the relative positions between object o_i and o_a predicted by the network. A represents the set of objects within the area. $C_{overlap}$ constrains the overlap between oriented bounding boxes of any two objects as small as possible, and C_{OOB} aims to avoid the object boxes lying out of the area boundary, whose size s_a is generated from pre-trained LLM in stage 1.

Global optimization. We then organize the areas to form scenes with global optimization. Each functional area takes the orientation of its anchor object as its own orientation. Based on observations in our daily life, the optimization is formulated to place the functional areas against the walls and far from each other with orientations pointing inside the scene, while avoiding object overlap and out-of-boundary:

$$\begin{aligned} \min_{a_i} & \sum_{a_i} |\mathrm{D_w}(a_i, s_r)| - \sum_{a_i, a_j} |\mathrm{D_a}(a_i, a_j)|, \\ s.t. \quad & C_{overlap}(a_i, a_j), \quad \forall a_i, a_j \in S \\ & C_{OOB}(a_i, s_r), \quad \forall a_i \in S \end{aligned} \tag{5}$$

where a_i denotes area placement (center position and orientation). D_w is the distance between back side of the area and the boundary of the scene. D_a is the distance between two areas' bounding boxes. S is the set of areas within the scene. $C_{overlap}$ and C_{OOB} are same as in local optimization.

After the optimizations, we transform the coordinate systems to obtain object positions and orientations in the frame of scenes. Finally, we retrieve 3D object models from Objaverse (Deitke et al. 2023) and 3D-Front datasets (Fu et al. 2021) based on the CLIP scores, i.e. cosine similarity between object images and text embeddings. The object models are then scaled and placed according to the scene layouts.

5 Experiment and Results

5.1 Experiment Settings

Dataset. We conduct the comparison and ablation study on the 3D-Front dataset (Fu et al. 2021). That is, we train the deep-learning-based approaches on this dataset and constrain the LLM-assisted methods to synthesize scenes with object categories within this dataset, for a fair comparison. Following LayoutGPT (Feng et al. 2023), we take the room category and floor, i.e. its width and height, as input conditions and filter out the scenes with irregular floors. The sizes of the training sets are 3397 and 690 for bedrooms and living rooms, while the corresponding test sets are 60 and 53.

Metrics. We evaluate the generated scenes from two perspectives. One is the physical feasibility of the scenes, estimated by the overlap between oriented bounding boxes and out-of-boundary metrics, i.e. overlap and OOB. The other is the reasonable organization of scenes, for which we select some common object pairs, i.e. bed-nightstand, table-chair, table-sofa, and measure the averaged KL-divergence between the relative placement distributions of the ground-truth scenes in the test sets and the generated scenes.

Implementation. We use GPT-4(Achiam et al. 2023) for all the LLM-assisted approaches for the evaluation (the open-source LLaMA also works well with our approach). We train the hierarchy-aware neural network with 500 epochs using the Adam optimizer, where the batch size is 4 and the learning rate is 1e-4. The network is trained on the combination of the bedroom and living room training sets, which takes about 8 hours on a Nvidia 4090 GPU. Our divide-and-conquer optimization is implemented with the GUROBI solver (Gurobi Optimization, LLC 2023). Our approach takes about 2 minutes to synthesize a scene with 8 objects.

Models	Bedrooms			Living Rooms		
	OOB↓	Overlap↓	KL Div.↓	OOB↓	Overlap↓	KL Div.↓
ATISS (Paschalidou et al. 2021)	0.48	0.18	0.19	0.50	0.34	0.23
Diffuscene (Tang et al. 2023)	0.77	0.14	0.29	0.95	0.30	0.28
LayoutGPT (Feng et al. 2023)	0.70	0.16	0.25	0.64	0.21	0.36
HOLODECK (Yang et al. 2024b)	0.00	0.00	0.27	0.00	0.00	0.34
Ours	**0.00**	**0.00**	**0.09**	**0.00**	**0.00**	**0.13**

Table 1: Quantitative evaluation of the generated scene layouts of different approaches.

Models	Bedrooms		Living Rooms	
	#Rel.	#Obj.	#Rel.	#Obj.
HOLODECK	0.77	0.93	0.69	0.97
Our	**0.88**	**0.99**	**0.86**	**1.00**

Table 2: Semantic alignment between LLM-generated descriptions and the resulting scenes of HOLODECK and ours.

Models	Scene Effect.	Physical.	Layout.
ATISS	3.55	2.86	3.10
DiffuScene	3.38	2.55	2.57
LayoutGPT	3.04	2.45	2.23
Holodeck	3.12	3.75	3.23
Ours	**4.73**	**4.69**	**4.55**

Table 3: Averaged score of perceptual study on the generated scenes. The participants give scores between 1-5 w.r.t. scene effectiveness, physical feasibility and layout rationality.

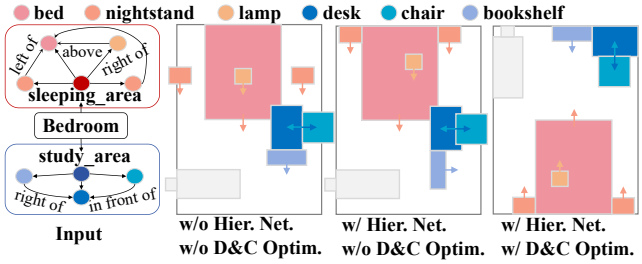

Figure 5: Topview visualizations of the ablation study.

5.2 Comparisons

We compare with two types of state-of-the-art approaches, including those training deep neural networks from scratch, i.e. ATISS (Paschalidou et al. 2021) and DiffuScene (Tang et al. 2023), and LLM-assisted indoor scene synthesis, i.e. LayoutGPT (Feng et al. 2023) and HOLODECK (Yang et al. 2024b). We re-train the deep networks using their released code on the same train/test split. For the LLM-assisted approaches, we use their implementations of the pipelines and invoke the same version of GPT for the inference.

Qualitative Evaluation. Figure 4 presents generated scenes of different methods. The deep learning methods ATISS and DiffuScene generate results with reasonable placements of objects. But the networks are not guaranteed to ensure the physical feasibility of the scene layouts and sometimes cause object overlap and out of the floor boundary. LayoutGPT, which uses in-context learning to infer numerical layouts based on the demonstrated examples, generates many incorrect orientations and positions. HOLODECK produces relatively better results in terms of physical feasibility, but some objects are not placed in the optimal position as specified by the LLM. By contrast, our approach is able to produce more reasonable and feasible scene layouts.

Quanlitative Evaluation. Table 1 validates the observations from the visual results. Among all the methods, we achieve the best in terms of both the physical feasibility (overlap and OOB) and the reasonable relative placements (KL Div.). It is interesting to see that the data-driven approaches are good at objects' relative positions and LLM-assisted optimization, i.e. HOLODECK, obtains more feasible results, while ours takes the merit of both and won the best on all the metrics.

We further provide an additional quantitative comparison with HOLODECK, the closest work to ours, as both use the LLM to generate textual scene descriptions and then solve for the scene layouts. The difference is that HOLODECK requires dense and detailed spatial relations while ours uses hierarchical structures with sparse relations as well as a neural network to infer the fine-grained relative placements. Table 2 reports the semantic alignment between the LLM-generated descriptions and the generated scenes. Specifically, #Rel. counts the percentage of relative placements matching with LLM-generated spatial relations and #Obj. counts the existence of LLM-specified objects. Obviously, our results align better with LLM arrangements, implying the advantages of using hierarchical scene representation with our approach.

Perceptual Study. The perceptual study evaluates the quality of scenes generated by different methods (those used in the comparison experiments). We present 25 generated scenes with the input user requirements to 30 participants, who rate them on a 5-point Likert scale from three aspects: scene effectiveness, physical feasibility, and layout rationality. Since both the rendered scenes and the topview visualizations are presented, the participants are sensitive to unreasonable placements which might be covered by the occlusion in the renderings. Table 3 shows that our results get the highest scores w.r.t. all three aspects, i.e. high scene effectiveness as we prompt the LLM with functional area considerations, high feasibility and layout rationality as our approach can effectively generate feasible scenes aligned with LLM arrangement. Otherwise, although HOLODECK prevents object overlap or out-of-boundary, since its results may break the LLM arrangements, it may affect possible human activity, such as shown in the top row case in Figure 4.

Baselines		Bedrooms			Living Rooms		
Hierarchy-Aware Net.	D&C Optim.	OOB↓	Overlap↓	KL Div.↓	OOB↓	Overlap↓	KL Div.↓
×	×	0.80	0.09	0.23	0.98	0.10	0.24
✓	×	0.73	0.18	0.09	0.93	0.21	0.16
×	✓	0.00	0.00	0.23	0.00	0.00	0.23
✓	✓	**0.00**	**0.00**	**0.09**	**0.00**	**0.00**	**0.13**

Table 4: Quantitative evaluation of ablation study to validate our key designs: the hierarchy-aware network (Hierarchy-Aware Net.) to infer fine-grained relative placements and divide-and-conquer optimization (D&C optim.) to solve the final layouts.

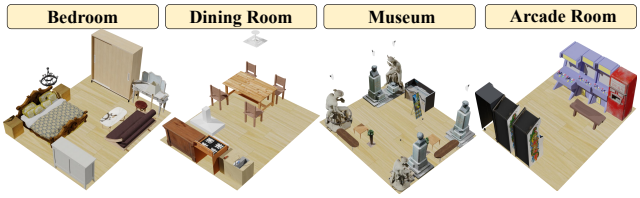

Figure 6: Diverse results in the open-vocabulary task.

5.3 Ablation Study

The ablation study validates the key designs of our approach, i.e. the hierarchy-aware network and the divide-and-conquer optimization, by removing the corresponding stage from our pipeline. When removing the hierarchy-aware network, we pre-define the relative placement coordinates for the textual spatial relations to replace the predictions of the network. When removing the divide-and-conquer optimization, we directly require the LLM to generate coordinates of anchor objects and transform the relative placements of the other objects into global coordinates without optimization refinement. The results are shown in Table 4 and Figure 5. It indicates that our hierarchy-aware network is of vital importance in capturing the reasonable relative placements between objects, and the divide-and-conquer optimization ensures the physical feasibility of the generated scene layouts.

6 Applications

Open-vocabulary scene synthesis. To demonstrate the generalization of our approach, we show the open-vocabulary scene synthesis results in Figure 6. By removing object category constraints in the LLM prompt, we generate scenes of a dining room, museum, and arcade room, not trained on by our model, to validate its practical usage. The results show that the LLM generates reasonable hierarchical structures for arbitrary requirements, while our generalizable hierarchy-aware network and divide-and-conquer optimization produce realistic scenes with various descriptions.

Interactive scene editing. Our approach also supports user-friendly language-guided interactive scene editing. Specifically, we describe the current state of the scene and an editing instruction as the input to LLM, and add an additional constraint to the divide-and-conquer optimization to maintain the placements of unchanged objects as much as possible. As shown in Figure 7, the LLM can modify the scene by adding and removing objects. Moreover, with the hierarchical scene structure and our approach, the edited scenes exhibit minimal changes from the original scenes while satisfying the LLM arrangments and the expectations of users.

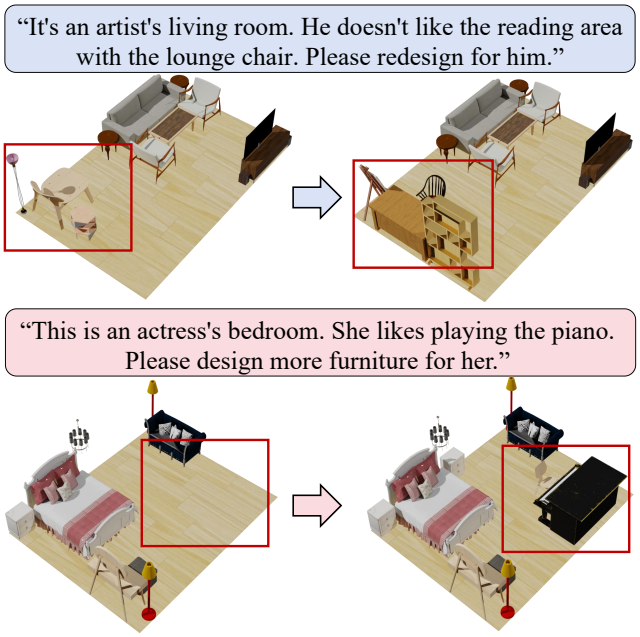

Figure 7: Language-guided interactive scene editing.

7 Conclusions

We present an LLM-assisted hierarchical indoor scene synthesis approach to produce customized and diverse scenes. Our approach fully takes advantage of the three-level hierarchical structure, where the LLM generates the descriptions of hierarchical scenes, a hierarchy-aware network infers the fine-grained relative placements, and a divide-and-conquer optimization solves the feasible layout.

Our approach still holds some limitations. First, for the simplicity of optimization, we assume rectangular floors for the generated scene. It is possible to utilize the spatial-aware optimization with simulated annealing algorithms (Yu et al. 2011) for irregular floors. Second, LLM sometimes generates infeasible configurations with too many or large objects and we randomly remove some areas or objects to address this, affecting scene quality. Third, since our hierarchical scene representation takes each object as its oriented bounding boxes without geometric details, our generated scenes are not sensitive to object shapes, such as L-shaped sofa.

Acknowledgments

This work is supported in part by the Shandong Province Excellent Young Scientists Fund Program (Overseas) (Grant No. 2022HWYQ-048 and 2023HWYQ-034), the TaiShan Scholars Program (Grant No. tsqn202211289), the National Natural Science Foundation of China (Grant No. 62325211 and 62132021).

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

An, V. D.; Vu, M. N.; Nguyen, T.; Huang, B.; Nguyen, D.; Vo, T.; and Nguyen, A. 2023. Language-driven Scene Synthesis using Multi-conditional Diffusion Model. *NeurIPS*.

Chang, A. X.; Eric, M.; Savva, M.; and Manning, C. D. 2017. SceneSeer: 3D Scene Design with Natural Language. *CoRR*, abs/1703.00050.

Chang, A. X.; Monroe, W.; Savva, M.; Potts, C.; and Manning, C. D. 2015. Text to 3D Scene Generation with Rich Lexical Grounding. *ACL-IJCNLP*, 53–62.

Chang, A. X.; Savva, M.; and Manning, C. D. 2014. Learning Spatial Knowledge for Text to 3D Scene Generation. *EMNLP*, 2028–2038.

Deitke, M.; Liu, R.; Wallingford, M.; Ngo, H.; Michel, O.; Kusupati, A.; Fan, A.; Laforte, C.; Voleti, V.; Gadre, S. Y.; VanderBilt, E.; Kembhavi, A.; Vondrick, C.; Gkioxari, G.; Ehsani, K.; Schmidt, L.; and Farhadi, A. 2023. Objaverse-XL: A Universe of 10M+ 3D Objects. *arXiv preprint arXiv:2307.05663*.

Dhamo, H.; Manhardt, F.; Navab, N.; and Tombari, F. 2021. Graph-to-3D: End-to-End Generation and Manipulation of 3D Scenes Using Scene Graphs. *ICCV*, 16332–16341.

Feng, W.; Zhu, W.; Fu, T.-J.; Jampani, V.; Akula, A. R.; He, X.; Basu, S.; Wang, X. E.; and Wang, W. Y. 2023. LayoutGPT: Compositional Visual Planning and Generation with Large Language Models. *arXiv preprint arXiv:2305.15393*.

Fisher, M.; Ritchie, D.; Savva, M.; Funkhouser, T.; and Hanrahan, P. 2012. Example-based synthesis of 3D object arrangements. *ACM Transactions on Graphics*, 31(6): 1–11.

Fu, H.; Cai, B.; Gao, L.; Zhang, L.-X.; Wang, J.; Li, C.; Zeng, Q.; Sun, C.; Jia, R.; Zhao, B.; et al. 2021. 3d-front: 3d furnished rooms with layouts and semantics. *ICCV*, 10933–10942.

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

Fu, Q.; Fu, H.; Yan, H.; Zhou, B.; Chen, X.; and Li, X. 2020. Human-centric metrics for indoor scene assessment and synthesis. *Graphical Models*, 110: 101073.

Fu, R.; Wen, Z.; Liu, Z.; and Sridhar, S. 2024. AnyHome: Open-Vocabulary Generation of Structured and Textured 3D Homes. *ECCV*.

Gani, H.; Bhat, S. F.; Naseer, M.; Khan, S.; and Wonka, P. 2023. Llm blueprint: Enabling text-to-image generation with complex and detailed prompts. *arXiv preprint arXiv:2310.10640*.

Gao, L.; Sun, J.; Mo, K.; Lai, Y.-K.; Guibas, L. J.; and Yang, J. 2023. SceneHGN: Hierarchical Graph Networks for 3D Indoor Scene Generation With Fine-Grained Geometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45: 8902–8919.

Gurobi Optimization, LLC. 2023. Gurobi Optimizer Reference Manual.

Huang, I. Y.; Krishna, V.; Atekha, O. E.; and Guibas, L. J. 2023. Aladdin: Zero-Shot Hallucination of Stylized 3D Assets from Abstract Scene Descriptions. *arXiv preprint arXiv:2306.06212*.

Leimer, K.; Guerrero, P.; Weiss, T.; and Musialski, P. 2022. LayoutEnhancer: Generating good indoor layouts from imperfect data. *SIGGRAPH Asia*, 1–8.

Li, M.; Patil, A. G.; Xu, K.; Chaudhuri, S.; Khan, O.; Shamir, A.; Tu, C.; Chen, B.; Cohen-Or, D.; and Zhang, H. R. 2019. GRAINS: Generative Recursive Autoencoders for INdoor Scenes. *ACM Transactions on Graphics*, 38(2): 12:1–12:16.

Lian, L.; Li, B.; Yala, A.; and Darrell, T. 2023. Llm-grounded diffusion: Enhancing prompt understanding of text-to-image diffusion models with large language models. *arXiv preprint arXiv:2305.13655*.

Ma, R.; Li, H.; Zou, C.; Liao, Z.; Tong, X.; and Zhang, H. 2016. Action-driven 3D indoor scene evolution. *ACM Transactions on Graphics*, 35: 1 – 13.

Ma, R.; Patil, A. G.; Fisher, M.; Li, M.; Pirk, S.; Hua, B.; Yeung, S.; Tong, X.; Guibas, L. J.; and Zhang, H. 2018a. Language-driven synthesis of 3D scenes from scene databases. *ACM Transactions on Graphics*, 37(6): 212.

Ma, R.; Patil, A. G.; Fisher, M.; Li, M.; Pirk, S.; Hua, B.-S.; Yeung, S.-K.; Tong, X.; Guibas, L.; and Zhang, H. 2018b. Language-driven synthesis of 3D scenes from scene databases. *ACM Transactions on Graphics*, 37(6): 1–16.

Merrell, P. C.; Schkufza, E.; Li, Z.; Agrawala, M.; and Koltun, V. 2011. Interactive furniture layout using interior design guidelines. *SIGGRAPH*.

Nie, W.; Liu, S.; Mardani, M.; Liu, C.; Eckart, B.; and Vahdat, A. 2024. Compositional Text-to-Image Generation with Dense Blob Representations. *arXiv preprint arXiv:2405.08246*.

Paschalidou, D.; Kar, A.; Shugrina, M.; Kreis, K.; Geiger, A.; and Fidler, S. 2021. ATISS: Autoregressive Transformers for Indoor Scene Synthesis. *NeurIPS*.

Patil, A. G.; Patil, S. G.; Li, M.; Fisher, M.; Savva, M.; and Zhang, H. 2023. Advances in Data-Driven Analysis and Synthesis of 3D Indoor Scenes. *arXiv preprint arXiv:2304.03188*.

Qi, S.; Zhu, Y.; Huang, S.; Jiang, C.; and Zhu, S.-C. 2018. Human-Centric Indoor Scene Synthesis Using Stochastic Grammar. *CVPR*, 5899–5908.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable Visual Models From Natural Language Supervision. *arXiv preprint arXiv:2103.00020*.

Savva, M.; Chang, A. X.; and Agrawala, M. 2017. SceneSuggest: Context-driven 3D Scene Design. *CoRR*, abs/1703.00061.

Song, S.; Yu, F.; Zeng, A.; Chang, A. X.; Savva, M.; and Funkhouser, T. 2017. Semantic scene completion from a single depth image. *CVPR*, 1746–1754.

Sun, J.-M.; Yang, J.; Mo, K.; Lai, Y.-K.; Guibas, L.; and Gao, L. 2024a. Haisor: Human-aware indoor scene optimization via deep reinforcement learning. *ACM Transactions on Graphics*, 43(2): 1–17.

Sun, W.; Li, M.; Li, P.; Cao, X.; Meng, X.; and Meng, L. 2024b. Sequential selection and calibration of video frames for 3D outdoor scene reconstruction. *CAAI Transactions on Intelligence Technology*.

Sun, W.; Li, X.; Li, M.; Wang, Y.; Zheng, Y.; Meng, X.; and Meng, L. 2022. Sequential Fusion of Multi-view Video Frames for 3D Scene Generation. *CICAI*.

Tang, J.; Nie, Y.; Markhasin, L.; Dai, A.; Thies, J.; and Nießner, M. 2023. DiffuScene: Scene Graph Denoising Diffusion Probabilistic Model for Generative Indoor Scene Synthesis. *arXiv preprint arXiv:2303.14207*.

Wang, K.; Lin, Y.-A.; Weissmann, B.; Savva, M.; Chang, A. X.; and Ritchie, D. 2019. PlanIT: planning and instantiating indoor scenes with relation graph and spatial prior networks. *ACM Transactions on Graphics*, 38: 132:1–132:15.

Wang, K.; Savva, M.; Chang, A. X.; and Ritchie, D. 2018. Deep convolutional priors for indoor scene synthesis. *ACM Transactions on Graphics*, 37: 1 – 14.

Wei, Q.; Ding, S.; Park, J. J.; Sajnani, R.; Poulenard, A.; Sridhar, S.; and Guibas, L. J. 2023. LEGO-Net: Learning Regular Rearrangements of Objects in Rooms. *CVPR*, 19037–19047.

Xu, K.; Ma, R.; Zhang, H.; Zhu, C.; Shamir, A.; Cohen-Or, D.; and Huang, H. 2014. Organizing heterogeneous scene collections through contextual focal points. *ACM Transactions on Graphics*, 33(4): 1–12.

Yang, X.; Chang, T.; Zhang, T.; Wang, S.; Hong, R.; and Wang, M. 2024a. Learning Hierarchical Visual Transformation for Domain Generalizable Visual Matching and Recognition. *International Journal of Computer Vision*, 1–27.

Yang, X.; Hu, F.; and Ye, L. 2021. Text to Scene: A System of Configurable 3D Indoor Scene Synthesis. *MM*, 2819–2821.

Yang, Y.; Sun, F.-Y.; Weihs, L.; VanderBilt, E.; Herrasti, A.; Han, W.; Wu, J.; Haber, N.; Krishna, R.; Liu, L.; Callison-Burch, C.; Yatskar, M.; Kembhavi, A.; and Clark, C. 2024b. Holodeck: Language Guided Generation of 3D Embodied AI Environments. *CVPR*, 16227–16237.

Ye, S.; Wang, Y.; Li, J.; Park, D.; Liu, C. K.; Xu, H.; and Wu, J. 2022. Scene synthesis from human motion. *SIGGRAPH Asia*, 1–9.

Yi, H.; Huang, C.-H. P.; Tripathi, S.; Hering, L.; Thies, J.; and Black, M. J. 2023. MIME: Human-aware 3D scene generation. *CVPR*, 12965–12976.

Yu, L.-F. C.; Yeung, S.-K.; Tang, C.-K.; Terzopoulos, D.; Chan, T. F.; and Osher, S. 2011. Make it home: automatic optimization of furniture arrangement. *SIGGRAPH*.

Zhai, G.; Örnek, E. P.; Chen, D. Z.; Liao, R.; Di, Y.; Navab, N.; Tombari, F.; and Busam, B. 2024a. EchoScene: Indoor Scene Generation via Information Echo over Scene Graph Diffusion. *arXiv preprint arXiv:2405.00915*.

Zhai, G.; Örnek, E. P.; Wu, S.-C.; Di, Y.; Tombari, F.; Navab, N.; and Busam, B. 2024b. Commonscenes: Generating commonsense 3d indoor scenes with scene graphs. *NeurIPS*, 36.

Zhou, Y.; While, Z.; and Kalogerakis, E. 2019. SceneGraphNet: Neural Message Passing for 3D Indoor Scene Augmentation. *ICCV*, 7383–7391.