Taming LLMs for Codematic Indoor Scene Generation

Yixun Liang[1,2,3], Qianyi Wu[3], Chuan Fang[1,2], Rui Chen[1,2,3]
Jiahang Liu[3], Jianfeng Zhang[3], and Ping Tan[1,2]

[1] Hong Kong University of Science and Technology
[2] Shenzhen Loop Area Institute
[3] ByteDance Seed

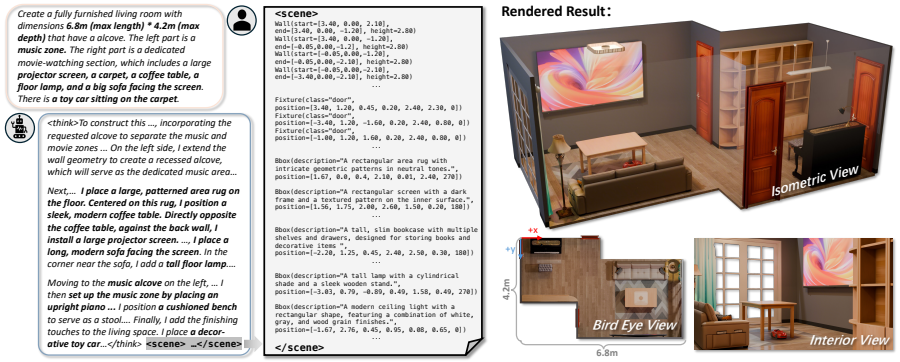

Fig. 1: Given prompts, SceneSpinner generates 3D scene like coding. It can infer arrangements from abstract concepts (music zone), plan spatial relations among objects (movie-watching section), and follow a specific request (toy car). Objects in rendered result are generated by in-house model. (Directions in our system: $+x$ right, $+y$ back.)

Abstract. The generation of plausible layouts is a critical step in text-driven 3D scene synthesis. While Large Language Models (LLMs) have shown promise in high-level organization in this task, their native fine-grained spatial placement remains a significant bottleneck, often requiring in-context guidance or complex agentic reflections to fix errors. This paper focuses directly on strengthening the core spatial ability by finetuning LLM to generate structured scene code (codematic indoor scene generation). We identify and address two fundamental issues hindering the LLM from doing this: data scarcity and poor instruction following. First, to mitigate data scarcity, we consolidate multiple heterogeneous datasets, spanning procedural, real-world, and professionally designed scenes, into a unified, large-scale training corpus. This dataset comprises **52K rooms** and **280k data pairs**, featuring diverse descriptions at both the scene and object levels. Second, even with abundant data, LLMs exhibit poor instruction following due to an information imbalance, where the token-heavy layout history overwhelms the concise user prompt. To resolve this, we propose SceneSpinner, a framework that introduces a language-based planning stage to provide high-level reasoning and a novel Conditional

Mutual Information (CMI) regularization to explicitly force the model to focus on user instructions. Experiments demonstrate that our approach significantly improves the ability of LLMs to generate plausible, diverse, and instruction-aligned 3D layouts.

Keywords: 3D Generation · Text-driven Scene Generation · Large Language Model

1 Introduction

Interactive generation of 3D indoor scenes is a cornerstone problem in computer graphics and artificial intelligence, with pivotal applications in embodied AI simulation, intelligent interior design, and game development. A fundamental challenge within this domain is the generation of plausible 3D object layouts from user prompts, a persistent goal that continues to drive research. Early works primarily relied on systems governed by explicit rules [12, 38, 47] to position objects under manual constraints. Later approaches like [15, 43, 53] shifted to a paradigm driven by data, training specialized models for layout synthesis. However, the limited scale and diversity of available training data constrain the effectiveness of such methods, preventing them from achieving layout synthesis with an open vocabulary and limiting the diversity of their outputs.

Powered by the rich common sense and knowledge of LLMs [1, 42, 46, 55], researchers have increasingly focused on leveraging these models to produce plausible layouts for scene generation. The structured representation of object layout, encompassing position, rotation, scale, and description, is naturally supported by the text output format of LLMs [4, 36], providing a solid foundation for this paradigm. To harness this potential, recent works have explored various strategies: some utilize learning from context to guide the model with few examples [6, 71], others incorporate external rules and physical constraints to prune invalid outputs [23, 35, 48], and sophisticated frameworks involving multi-modal agents employ iterative verification loops to refine the results [62, 69]. Some of them even directly use training-based methods as initializations tool [69]. However, the efficacy of these downstream refinement strategies is heavily dependent on the quality of the initial layout proposal. This dependency exposes a critical gap in the current methodology and raises a fundamental question: *What hinders the intrinsic ability of the LLM to generate a plausible object layout directly?*

To address this, we investigate fine-tuning LLMs to generate codematic layout representations—structured textual formats that LLMs can both interpret and produce [4, 36] as shown in Fig. 1. However, we identify two key obstacles. First, existing datasets are scarce and heterogeneous. Effective fine-tuning requires rich, multi-modal annotations, including open-vocabulary object descriptions and precise 7-DoF poses (covering position, size, and orientation), along with corresponding scene captions from multiple perspectives. No current dataset provides this combination of labels [7, 19, 36, 76]. To bridge this gap, we reprocess and unify diverse data sources, spanning real-world scans, procedurally generated environments, and professional designs, by annotating each object and

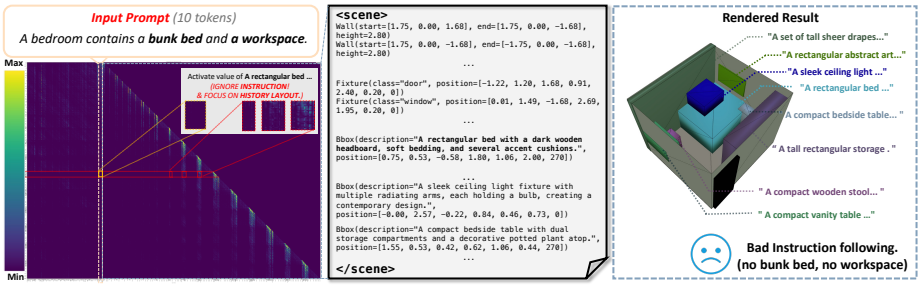

Fig. 2: Pilot Study. Finetuning LLMs in a vanilla way for codematic scene generation results in poor instruction adherence. The models tend to focus more on previous layout, rather than following the given instructions as shown in the left attention map.

scene with captions in a consistent format. This unified data foundation enables large-scale training for codematic scene generation.

Even with this extensive dataset, a second, more subtle challenge emerges: a critical imbalance of information inherent to the autoregressive generation process. As Fig. 2 illustrates, the model's generation is conditioned on both high-level user instructions (10 tokens) and low-level layout history for reasonable placement(782 tokens). This disparity leads to difficulties: since the layout history vastly outweighs the instructions and is statistically more similar to the target output, the model tends to forget the user instruction. As illustrated by the attention map in Fig. 2, the model focuses almost exclusively on previously generated code when generating a "bed" and ignores the "bunkbed" instruction. Consequently, the model struggles to capture user intent.

To address these dual challenges, we introduce SceneSpinner, a novel framework that enhances the capabilities of LLMs for codematic indoor scene generation. First, to mitigate the imbalance of tokens and inspired by the success of reasoning via chains of thought in LLMs [59], we introduce an intermediate planning stage. This forces the LLM to decompose the request of the user into a coherent, high-level natural language plan before generating code. This plan serves a dual purpose: it balances the distribution of tokens by providing a richer context, and it acts as a cognitive scaffold, bridging the semantic gap between the concise user prompt and the complex scene code. This guides the generation process with explicit and structured reasoning.

Second, to correct failures in attentional focus, we propose a method to quantify and regulate the reliance of the model on different conditions. Drawing from information theory [9], we derive a formulation based on conditional mutual information (CMI) to measure the contribution of the user instruction. Specifically, in the next token prediction paradigm, the conditional mutual information (CMI) of a predicted token is defined as the KL divergence between the model's prediction distribution given layout and user instructions and its distribution given the layout alone. This metric allows us to monitor whether the awareness

of the model to the user instructions is diminishing (as verified in Sec. 6). Building on this insight, we propose a training recipe that efficiently regulates this mutual information without significant computational overhead, preventing the model from being overwhelmed by the layout history.

In summary, our contributions are:

- We identify and address two fundamental challenges in codematic layout generation based on LLMs: limitations in datasets and imbalance of information during autoregressive generation.
- We propose a unified pipeline for data engineering that reprocesses and relabels 5 heterogeneous datasets, creating a large-scale dataset for fine-tuning LLMs on "codematic" scene representations.
- We introduce SceneSpinner, a finetuned LLM for codematic indoor scene generation featuring a planning stage based on language and trained by a training recipe based on CMI to resolve the information imbalance and ensure robust following of instructions.

2 Related Works

Generative Models for 3D Scene Synthesis. Early approaches [12, 47, 72] to 3D indoor scene generation relied heavily on procedural modeling and rule/optimization-based systems. These methods enforced heuristic constraints on object arrangement and asset retrieval to generate diverse scenes. However, a persistent bottleneck in these systems has been the *layout planning module*, which struggles to produce natural and varied arrangements without extensive manual engineering. The advent of large-scale 3D indoor scene datasets catalyzed a shift toward data-driven approaches [51, 56]. Early works explored model implicit layout distributions via adversarial training [39, 40, 67] or explicit distribution mapping [2, 45, 66], but still suffered from limited quality. These limitations motivated the community to explore more expressive architectures. Diffusion-based approaches, such as [15, 30, 33, 53, 60, 70, 73], formulate layout generation as a denoising process, while autoregressive models [24, 43, 49, 57, 61] treat it as a sequence-to-sequence task. Despite their advances, these methods are typically trained on limited datasets with closed vocabularies, restricting their ability to generalize to open-ended user instructions or generate diverse and complex scenes beyond the training distribution.

LLM-Driven Spatial Reasoning. A critical ingredient missing from purely geometric approaches is common-sense knowledge about spatial relationships and object semantics [13,21,22,27,52]. The emergence of Large Language Models (LLMs) has injected this vital capability into scene generation [18,34,41]. Frameworks like I-Design [6] and Holodeck [71] leverage LLMs to enrich the semantic diversity of generated layouts, though they often lack grounded 3D spatial understanding. Other works [23,35,58] combine LLMs with vision-guided feedback but are primarily limited to refining inaccurate poses rather than generating holistic layouts from scratch. Building on the agentic capabilities of LLMs, recent methods such as [62, 69] incorporate iterative verification and self-reflection

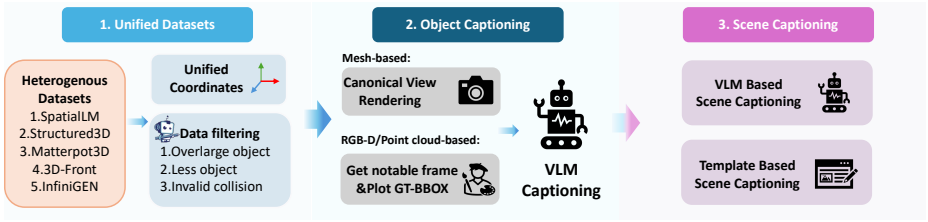

Fig. 3: Data process pipeline. We unifed heterogenous datasets for LLM finetuning.

loops to generate and correct layouts. However, the efficacy of these downstream refinement strategies is heavily dependent on the quality of the initial proposal generated by the LLMs [62, 69]. If the base model fails to provide a reasonable starting point, even robust verification loops struggle to converge. This dependency underscores a critical gap: *the need to improve the LLM's intrinsic ability to plan and generate plausible scene layouts directly.*

3D Indoor Scene Dataset. Beyond methodology, the quality and scale of training data are pivotal to the advancement of 3D scene generation. Early efforts utilized 3D scans of real-world environments [10, 25], such as Matterport3D [7]. While valuable for their realism, these datasets often suffer from noisy geometry, incomplete object coverage, and limited scene variety. To address these limitations, synthetic datasets [31, 76, 77] such as 3D-FRONT [19] were introduced, offering clean layouts and complete object assets, which also been widely used in indoor scene generation. However, these synthetic datasets often lack diversity and limited quality, further constrain the model performance. Recent studies have attempted to further improve data scale and quality by incorporating professional designs [16, 17, 36, 74, 75], providing a solid foundation for future research. Nevertheless, existing datasets remain fragmented and heterogeneous, often lacking unified instance-level annotations and broad diversity. This fragmentation motivates our work to process and unify five distinct datasets into a cohesive format, creating a robust, large-scale dataset to fuel the LLM training of layout prediction.

3 Dataset Processing

To finetune LLM model to do this task requires a large-scale dataset, which contains 7-DoF poses and textual descriptions for each object. As shown in Fig 3, to build this dataset, we relabel existing heterogeneous datasets, including professionally designed (e.g., SpatialLM [36], Structured3D [76]), real-world scans (e.g., Matterport3D [7]), and procedurally generated content (PCG) (e.g., 3D-FRONT [19], InfiniGen [47]). Our processing pipeline can also be applied to similar datasets for further scale-up.

Unified Coordinate System. We first normalize all scenes into a unified coordinate system where the Y-axis represents the vertical direction (aligned with

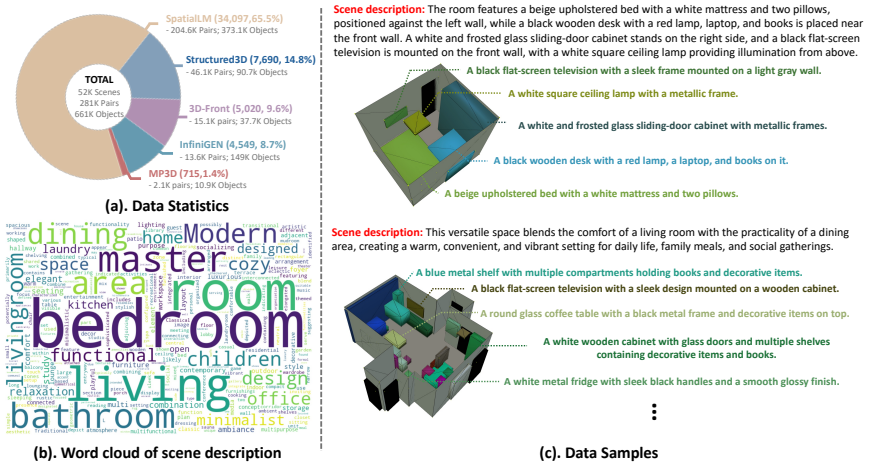

Fig. 4: Data statistics and samples. Our dataset contains 52K rooms with 281K data pairs.

gravity). The floor plane is aligned with y=0, and the scene's geometric center is set to the origin (0,0,0). For object orientation, we define the front face as the $+Z$ direction. Spatially, we partition the world coordinates $+X$ and $-X$ correspond to right and left, respectively. Conversely, along the Z-axis, $+Z$ denotes back, while $-Z$ denotes front.

Data Filtering. The second step is a rigorous data filtering process to ensure dataset quality. The criteria are based on the number of objects, the relative size of furniture, and the geometric quality of instances. We discard the room that contains fewer than 3 objects to prevent easy samples. We also remove scenes which contains oversize items that occupy more than one-third of the entire space. Finally, we remove the scene that contains invalid collisions between objects.

Object Captioning. To generate rich object descriptions, we employ a hierarchical strategy that adapts to the different data modality for each object: *1) Mesh-based.* For objects with available 3D meshes, such as PCG datasets [19,20], we render four canonical views and feed them to a Vision Language Model (VLM) [29] for direct captioning. *2) RGB-D/Point Cloud-based.* For datasets that did not provide the instance mesh, we prioritize the usage of rendered RGB images, either from the provided or the point-rendered results. We first obtain the projected bounding box of the interested object from the RGB frame, and then feed the highlighted objects into the VLM for detailed captions.

Scene Captioning. With all objects and their spatial relationships established in the unified space, we generate a holistic caption for the entire scene. To achieve this, our methodology incorporates two complementary captioning en-

gines: a template-based engine and a VLM-based engine. The primary goal of the template-based engine is to instill fundamental spatial reasoning. It systematically generates captions detailing the absolute positions and relative arrangements of objects, focusing on low-level concepts like direction and distance. Conversely, the VLM-based engine is similar to M3DLayout [75] but with different prompts, learns to connect natural language with the scene's semantic meaning. It produces captions that describe the scene from a holistic perspective, including its scene category, the function of its layout, the furniture placement, and the design purpose of this room, thereby teaching the model to understand the scene's high-level context. The specific rules for the template-based engine and the prompts used for VLM-based captioning are all detailed in the supplementary material.

Data Statistics and Comparison. Our training dataset contains 52,071 rooms after processing, yielding 281,586 scene-text pairs with VLM-based captions. Fig. 4 shows the data statistic and some examples from our VLM relabeled data. For our template-based captions, they are generated on-the-fly during training, making them theoretically infinite. We also provide the comparative advantages with existing datasets in Tab. 1. To the best of our knowledge, ours is the first dataset to feature open-set descriptions for objects in a scene, a key feature that directly supports LLM-driven layout generation.

Datasets.	Scenes	Objects	Layout Collection	Scene Description	Object Description
SUN3D [64]	254	N/A	RS	×	N/A
SceneNN [26]	100	N/A	RS	×	N/A
Matterport3D [8]	N/A	1,684	RS	×	N/A
ScanNet [11]	1,506	N/A	RS	×	N/A
Scan2CAD [3]	1,506	N/A	RS	×	N/A
OpenRooms [32]	1,068	97,607	RS	×	N/A
SceneNet [37]	57	N/A	RS	×	N/A
3DFront [19]	5,754	N/A	PD	×	N/A
Structured3D [76]	9,560	127,752	PD	×	CloseSet
SpatialLM [36]	49,963	469,516	PD	×	CloseSet
M3DLayout [75]	21,367	433,842	Mixture	✓	CloseSet
Ours	52,071	661,614	Mixture	✓	OpenSet

Table 1: Comparisons between existing 3D indoor scene datasets. "N/A" denotes "not available". "RS" means real scanned, and "PD" denotes professionally designed. "CloseSet" means that it only contains specific category labels, while "OpenSet" means specific text descriptions.

4 SceneSpinner

SceneSpinner is a finetuned LLM (Qwen3-4B-Instruct [65]) that can output scene code, and then use a 3D renderer to generate the overall 3D scene following the scene codes. As shown in Sec. 1, we find that directly finetuning LLM to generate scene codes is prone to collapse and makes the output focus only on scene codes. To handle these, we firstly introduce a planning phase like the LLMs'reasoning to extend the prompt token information while using the LLM's prior. Then we introduce a novel training strategy based on conditional mutual information to balance contributions of layout history and user instruction.

4.1 Planning Phase

To address the information density imbalance, we introduce a planning phase that reframes the generation task as a cascaded 'instruction -> plan -> code' process. The 'plan', a detailed natural language rationale, serves as an information-dense proxy for the user's intent. This balances the model's attention by creating an intermediate representation with a token count comparable to the layout code, ensuring instructions are not ignored.

To generate the "(*instruction, plan*)" training pairs, we built an automated pipeline. Inspired by [68], we avoid rule-based methods, which often lead to mode collapse. Instead, we prompt VLM [1,54] to generate diverse and coherent plans.

The process provides the VLM with comprehensive context for each sample: the user instruction, our coordinate system, the ground-truth scene code, and even rendered images. We use few-shot prompting to enhance its understanding of spatial relationships and the required format. The VLM is then prompted to simulate a designer's thought process, generating a CoT that has (1). all scene elements, (2). justifies each object's placement according to the instruction, and (3). logically infers the existence of necessary but unmentioned objects. The prompt for this is attached in the supplementary.

This pipeline yields diverse and semantically coherent planning labels. By training on the "instruction -> plan -> code" task, we enrich the conditional input, better leverage the LLM's intrinsic reasoning, and significantly mitigate performance collapse.

4.2 CMI for Condition Contributions Measurement

By increasing the token count of the conditional input, the planning phase serves as a data augmentation method that mitigates information imbalance. However, this merely provides two more balanced conditions (the "text prompt" noted as c_{text}, $c_{\text{text}} = c_{\text{instruction}} + c_{\text{think}}$ and the previous layout code, we note it as c_{code}), but it does not grant us direct control over how the model utilizes them during training. The model might still develop a preference for one condition over the other, ignoring the nuanced guidance in the "planning phase". Therefore, we require a method to quantify the contribution of c_{text}, enabling us to monitor and regularize its contribution throughout the training process.

To achieve this, we turn to conditional mutual information (CMI) from information theory [9]. Before proceeding with the derivation, we first abstract the action of generating layout code as a_{code}. The CMI between the generated action a_{code} and the text c_{text}, given the previous code context c_{code}, is defined as:

$$I(a_{\text{code}}; c_{\text{text}} | c_{\text{code}}) = H(a_{\text{code}} | c_{\text{code}}) - H(a_{\text{code}} | c_{\text{text}}, c_{\text{code}}), \tag{1}$$

where $H(X|Y)$ denotes the conditional entropy, which measures the remaining uncertainty of a random variable X given that Y is known. Intuitively, this equation quantifies the reduction in uncertainty about the a_{code} that is gained by introducing the condition c_{text}, given that c_{code} is already known. A high

CMI value indicates that the text provides significant influence for predicting this action.

By expanding the entropy terms and applying the chain rule of probability, the CMI can be rewritten as Eqn. (2) (We provide a detailed derivation in the supplementary). This reveals that the CMI is equivalent to the expected Kullback-Leibler (KL) divergence between the posterior distribution of the action with the text and the posterior without it:

$$I(a_{\text{code}}; c_{\text{text}}|c_{\text{code}}) = \mathbb{E}_{p(c_{\text{text}}, c_{\text{code}})} \left[D_{\text{KL}} \left(p(a_{\text{code}}|c_{\text{text}}, c_{\text{code}}) \parallel p(a_{\text{code}}|c_{\text{code}}) \right) \right], \tag{2}$$

Fortunately, this computation is tractable within the next token prediction (NTP) framework. Under the NTP paradigm, the set of all possible actions a_{code} is simply the discrete vocabulary of the model. Consequently, the conditional probability of any specific action is directly provided by applying the softmax function to the LLM's output logits. Thus, the both items in Eqn. (2) is formally expressed as:

$$p(a_{\text{code}}|c_{\text{text}}, c_{\text{code}}) = \text{softmax}(LLM(c_{\text{text}}, c_{\text{code}})) \tag{3}$$

$$p(a_{\text{code}}|c_{\text{code}}) = \text{softmax}(LLM(c_{\text{code}})) \tag{4}$$

Where LLM denotes the large language model finetuned for our task. The CMI for each training instance can be estimated by calculating the KL divergence between the model's output distribution when fully conditioned (with the text) and its distribution when conditioned only on the previous layout code. This allows us to monitor, or even constrain, the model's reliance on the text condition during the training process. Meanwhile, as shown in Sec. 6, the experimental observations are consistent with our hypothesis, confirming that this parameter can indeed quantify the model's awareness of instruction.

4.3 Training Recipe

Based on previous analysis, we aim to monitor/ regulate text contribution during LLM finetuning. However, a naive approach of creating a negative pair $p(a_{\text{code}}|c_{\text{code}})$ by omitting the instruction c_{text} is not trivial and inefficient. Unlike diffusion models that utilize dedicated unconditional placeholders, LLMs lack such a mechanism (unconditioned modeling), making simple omission problematic. Moreover, this "code only" branch would require a separate forward pass, which incurs additional computational overhead without contributing to the primary training objective (text to scene).

We propose an efficient alternative for generating conditioning pairs. Instead of ablating c_{text}, we replace it with a generalized, high-level categorical label, denoted as c_{label} (e.g., 'bedroom', 'living room'), which is also part of our training caption. This approach maintains efficiency because the forward passes for both the specific-condition (c_{text}) and the general-condition (c_{label}) compute a loss against the ground truth, fully contributing to the primary training objective. Meanwhile, the difference in information density between the two also makes it

Algorithm 1 Training Recipe of SceneSpinner

Input: Datasets $\mathcal{D}$, loss weights β, λ.
Initialization:
- $LLM()$: The Large Language Model.
- $CodeTemplate()$: Converts layout dictionary $Layout$ to string code.
- $PrepareIDs()$: Prepares input IDs I, labels L, and layout code indices A.
- sg(): stop gradient operation.

repeat
  Sample a room $\mathcal{R}$ from datasets $\mathcal{D}$
  Get layouts $Layout$, long description c_{text}, short description c_{label} from $\mathcal{R}$
  Convert layout to code: $a_{\text{code}} \leftarrow CodeTemplate(Layout)$
  Prepare data: $(I_{\text{long}}, L_{\text{long}}, A_{\text{long}}) \leftarrow PrepareIDs(a_{\text{code}}, c_{\text{text}})$
  Prepare data: $(I_{\text{short}}, L_{\text{short}}, A_{\text{short}}) \leftarrow PrepareIDs(a_{\text{code}}, c_{\text{label}})$
  Get logits: $[Z_{\text{long}}, Z_{\text{short}}] \leftarrow LLM([I_{\text{long}}, I_{\text{short}}])$
  Calculate CE loss: $\mathcal{L}_{\text{long}} \leftarrow CELoss(Z_{\text{long}}, L_{\text{long}})$
  Calculate CE loss: $\mathcal{L}_{\text{short}} \leftarrow CELoss(Z_{\text{short}}, L_{\text{short}})$
  Extract action logits: $z_{\text{long}} \leftarrow Z_{\text{long}}[A_{\text{long}}],\ z_{\text{short}} \leftarrow Z_{\text{short}}[A_{\text{short}}]$
  Calculate KL Divergence: $\mathcal{L}_{kl} \leftarrow D_{KL}(\ \text{softmax}(z_{\text{long}}) || \ \text{softmax}(\text{sg}(z_{\text{short}}))$
  Compute total loss: $\mathcal{L} \leftarrow \mathcal{L}_{\text{long}} + \beta\mathcal{L}_{\text{short}} - \lambda\mathcal{L}_{kl}$
  Update model parameters of LLM, by optimizing $\mathcal{L}$
until Converage

easy for us to quantify the residual's conditional contribution, which can then be monitored and regulated.

The overall training recipe is shown in Alg. 1. It should be noted that c_{text} contains both the initial instruction and a planning sequence. During the training process, we integrate this planning sequence into the data pair with a certain probability. In such cases, the total loss, $\mathcal{L}_{\text{long}}$, is composite. It includes not only the loss on the final a_{code} but also a loss on the "planning" sequence. This enables the simultaneous training of the model's ability to both planning and layout generation.

5 Experiements

Training Setting. We select Qwen3-4B-Instruct [65] as our base model for fine-tuning. Our training incorporates a diverse mixture of data. In addition to our proposed rule-based and VLM-based data, we include the Mixture of Thought datasets [5,28,44] and the Ultra-Chat [14] dataset to preserve the model's general reasoning and language understanding capabilities. Details about ratios of each dataset are posted in the supplementary.

Evaluation Settings. To evaluate the quality of our model, we curated a benchmark of 120 prompts covering various rooms from multiple perspectives, such as layout descriptions and design purposes.

For our evaluation metrics, we first follow the methodology from previous work [75], using CLIP to assess the semantic accuracy of the generated layouts. Additionally, inspired by SceneWeaver [69], we employ GPT-4o [29] for a

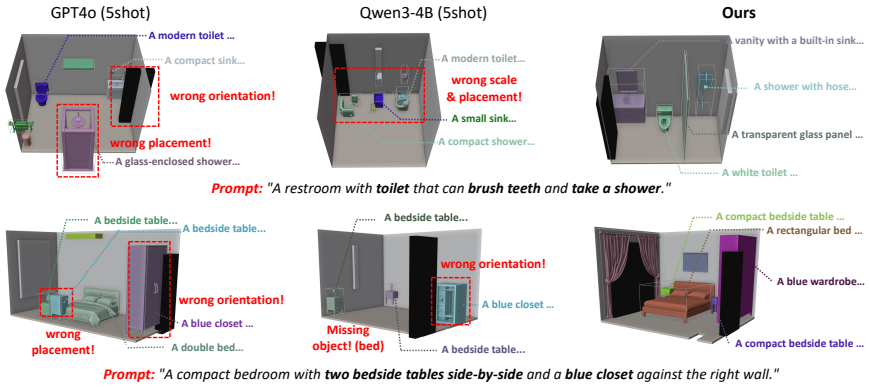

Fig. 5: In-context learning comparison. With few-shot prompting, the LLM generates compilable scene codes and infers necessary objects via reasoning. However, it struggles with fine-grained object placement, motivating the need for our proposed fine-tuning approach. White/black bboxes denote windows/doors.

more nuanced evaluation. However, our evaluation focus differs from that of SceneWeaver, which primarily assesses the reasonableness and functionality of generated states from short prompts. In contrast, our work requires a more rigorous assessment of how strictly the model adheres to detailed instructions. Therefore, we adapted their prompting strategy into our task, using GPT-4o to evaluate instruction-following capability and the fidelity and completeness of the outputs. Detailed evaluation prompts, the pipeline, and case studies demonstrating the validity of our proposed metrics are provided in the supplementary.

To ensure fairness, we uniformly employ cube [50] (an open-source text+bbox-to-3D generation framework) to generate all results when conducting comparisons through top-down view rendering, as our primary goal is to evaluate the accuracy of layout generation.

Baseline. In this paper, we first demonstrate that current Large Language Models (LLMs), such as Qwen3 [65] and GPT-4o [29], struggle to address this problem even with few-shot prompting. This finding highlights their limitations in precise spatial reasoning, thereby underscoring the necessity of our proposed training methodology. We then conduct a comparative analysis of our method against several state-of-the-art (SoTA) approaches. These include agent-based methods like HoloDeck [71] and I-design [6], as well as training-based SOTA methods such as Ctrl-Room [15] and M3DLayout [75]. Notably, M3DLayout provides both autoregressive and diffusion models.

5.1 Qualitative Comparision

In-context Learning for Layout Generation. As shown in Fig. 5, we compare our method with training-free in-context learning technique [59]. Guided by

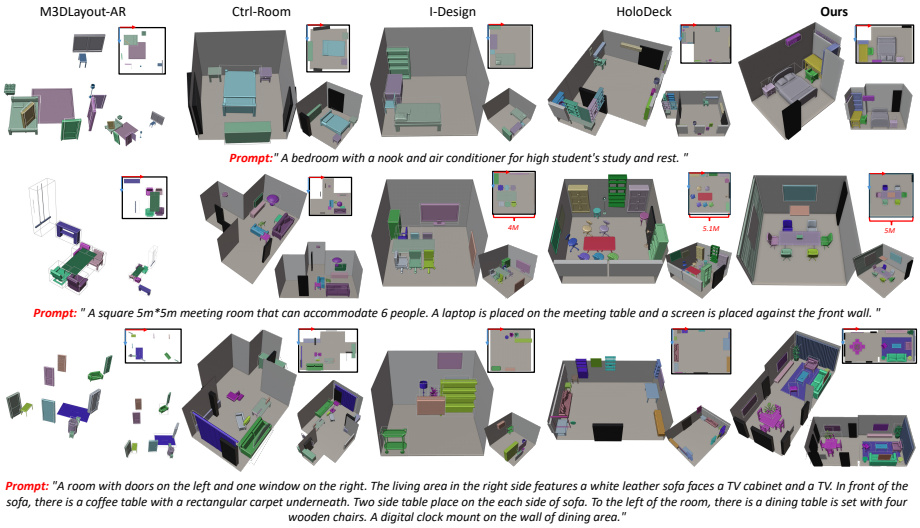

Fig. 6: Qualitative comparison. Our method consistently demonstrates superior generation quality and instruction following ability compared to SoTA methods across prompts of varying difficulty levels.

our custom system prompt and in-context learning setup (5-shot), our pretrained model, Qwen3 [65] and the GPT-4o [1] are capable of inferring and outputting scene code according to the specified format and successfully compiled. While the models can leverage their inherent knowledge to conceptualize the required objects (sink for brushing teeth, etc.), our experiments reveal a significant weakness of those pretrained models: a poor ability to place objects correctly. This finding validates the necessity of our fine-tuning process to enhance the LLM's capabilities in this area.

Baseline Comparision. We further conduct qualitative comparison across three scenarios by increasing the complexity of the prompt, as shown in Fig. 6. Task complexity ranges from basic compositional reasoning to instructions involving precise dimensions and implicit constraints. As this complexity grows, the performance gap between our method and the baselines widens. In the simplest task (first row in Fig. 6), our model correctly interprets the instruction to create a functional "nook", designing a dedicated study area that aligns with user intent. In the intermediate case with metric scale constraints, our method accurately respects fine-grained spatial requirements that both traditional and agent-based baselines fail to satisfy. In the most challenging scenario, where all baselines break down, our approach continues to generate coherent, high-quality layouts. These results highlight the robustness and practical effectiveness of our method. We provide more visual results in the supplementary.

Metrics.	M3DL_Diff [75]	M3DL_AR [75]	CtrlRoom [15]	I-Design [6]	HoloDeck [71]	Ours
Ins_Follow ↑	3.058	3.608	4.841	4.222	5.722	**7.852**
Lay_Fidelity&Comp ↑	3.691	3.908	5.291	4.533	6.037	**7.191**
Clip_Score ↑	0.182	0.187	0.187	0.167	0.259	**0.260**

Table 3: Quantitative comparison. We report Instruction Following score (Ins_Follow), Layout Fidelity and Competition score (Lay_Fidelity&Comp), and CLIP Score. Our method significantly outperforms current SoTA methods.

5.2 Quantitative Comparision

As shown in Tab. 3, our method consistently outperforms all baseline methods in the evaluated metrics, demonstrating its superior performance in the generation of instruction-guided layouts. We observe that existing agent-based methods often struggle with complex scenarios: Holodeck [71] frequently fails to follow instructions with limited physical plausibility, and I-Design [6] shows significant performance degradation in scenes with more than five objects. Similarly, diffusion-based approaches [15, 75] are limited by their reliance on fixed prompt patterns, which hinders their handling of open-vocabulary instructions. In contrast, our approach shows a commanding lead in adherence to user instructions and layout accuracy, achieving a score of 7.852 on "Instruction Following" and 7.191 on "Layout Fidelity and Competition". These results represent a significant improvement of 37.2% and 19.1%, respectively. Furthermore, in terms of high semantic alignment, our method also achieves the highest score of 0.260. These indicate that our generated scenes not only have a more accurate layout but also maintain a stronger correlation with the user instructions compared to all other baselines. These quantitative results robustly validate the effectiveness of our proposed approach.

6 Ablation Study

In this section, we first quantitatively evaluate the components of our proposed method. The ablation study, as shown in Tab. 2, confirms that each component is effective and contributes successfully to the overall performance. This result demonstrates the generalizability of the improvements achieved by our approach.

Planning Phase	CMI Recipe	Instruction Following	Layout Fidelity &Comp
×	×	6.371	6.557
×	✓	6.988	6.806
✓	✓	**7.631**	**7.157**

Table 2: Model component ablation. All of these parts are described in Sec. 6.

Planning Analysis We now further investigate the underlying mechanism of improvement brought from the planning phase. Specifically, we analyze the attention activation maps of the LLM layers during generation, drawing inspiration from [63]. We use the same prompt (A bedroom contains a bunk bed and a workspace.) in the pilot study in Fig. 2

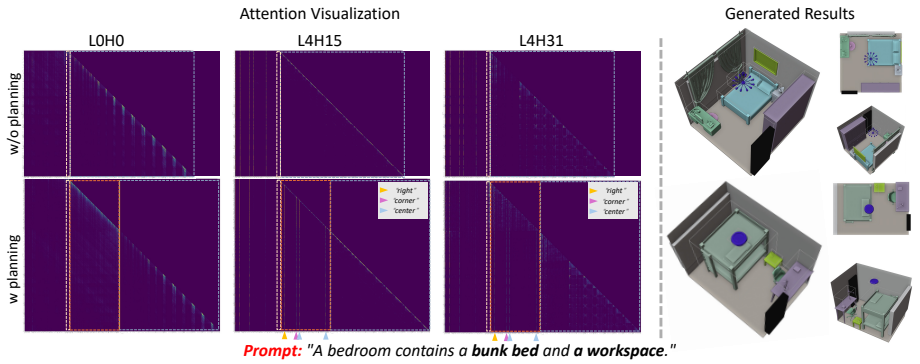

Fig. 7: Attention visualization comparison. The overall map consists of user instruction(same as Fig. 2), planning phase, and layout code; others are system prompts and templates. The model with planning maintains consistent attention on the instruction and planning tokens (specific spatial keywords as semantic anchors), whereas the baseline model's attention drifts to unrelated templates.

for comparison. A comparative analysis, illustrated in Fig. 7 (Left), reveals a stark contrast in attentional behavior. Without the planning stage, the model's attention drifts away from the user prompt as generation progresses, becoming dominated by the autoregressive history and unrelated tokens. However, with the introduction of planning, the model maintains robust attention on the user instruction/planning part throughout the entire generation process. Crucially, as highlighted in Fig. 7, we observe an emergent phenomenon: the model heavily attends to specific spatial keywords within the generated plan, such as 'right', 'corner', and 'center'. These keywords act as semantic anchors, grounding the subsequent layout code in explicit spatial concepts and leading to a more coherent and instruction-aligned 3D scene.

CMI Analysis This section empirically validates that KL divergence serves as a robust metric for quantifying the degradation in a model's instruction-following ability. As illustrated in Fig. 8 (top-left), the negative KL divergence exhibits a consistent upward trend during training. This rise is strongly correlated with a decline in instruction-following. Specifically, the qualitative examples in the remaining panels reveal a distinct failure mode: models with higher KL divergence can often produce a correct chain of thought, yet fail to adhere to this internal reasoning in their final generated output. This evidence confirms that an increasing KL divergence serves as a reliable indicator of this decoupling between instruction following and generation. In contrast, our proposed method effectively mitigates this rise, thereby maintaining instruction-following performance at an equivalent number of iterations.

7 Conclusion

In this work, we explore fine-tuning LLMs for codematic indoor scene generation, where 3D scenes are produced through structured, code-like text. To enable large-scale training, we first addressed the critical lack of suitable data by reprocessing and unifying five heterogeneous datasets to establish a robust and diverse

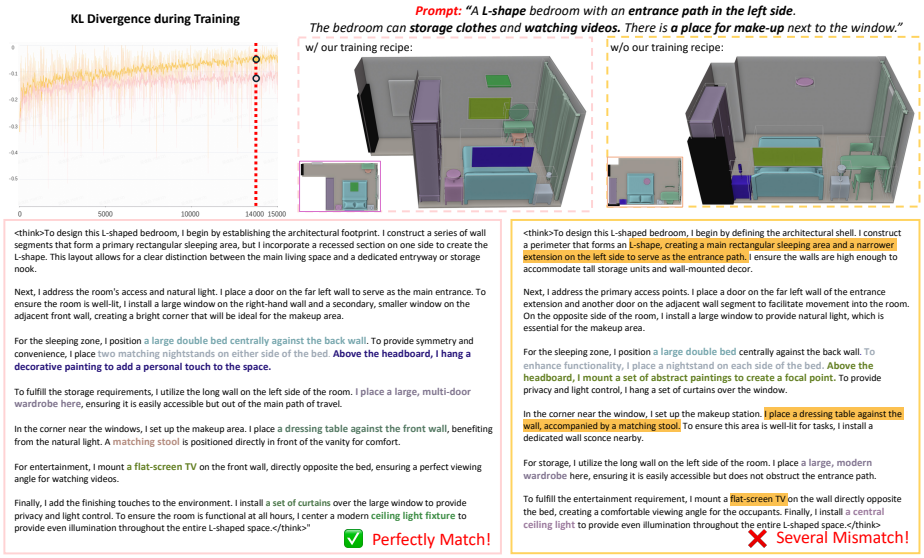

Fig. 8: Ablation study and training analysis. The top-left plot shows the KL training curve related to our proposed CMI (Sec. 4.2), where increasing negative KL implies the model learns to ignore instructions. The remaining panels present a case study: objects matching the thinking context are **bolded and colored**, while deviations are highlighted.

data foundation. While training on this dataset, we identified a more fundamental challenge: a critical information imbalance in the autoregressive process that causes models to ignore user instructions. To resolve this, we propose SceneSpinner, a finetuned LLM featuring a language-based planning stage trained on a CMI-based training recipe to enforce instruction adherence. As validated by our experiments, SceneSpinner achieves state-of-the-art (SOTA) performance in both fidelity and instruction following compared to previous methods. Our proposed method offers a novel paradigm for indoor scene generation that effectively leverages the prior knowledge of LLMs while resolving fine-grained placement issues, significantly pushing the boundaries of scene generation development.

8 Acknowledgement

We thank Rui Yang for helpful discussions. This work is supported by the Shenzhen Loop Area Institute under grant FPF10120260001.

References

1. Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023)
2. Arroyo, D.M., Postels, J., Tombari, F.: Variational transformer networks for layout generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13642–13652 (2021)
3. Avetisyan, A., Dahnert, M., Dai, A., Savva, M., Chang, A.X., Nießner, M.: Scan2cad: Learning cad model alignment in rgb-d scans. In: Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition. pp. 2614–2623 (2019)
4. Avetisyan, A., Xie, C., Howard-Jenkins, H., Yang, T.Y., Aroudj, S., Patra, S., Zhang, F., Frost, D., Holland, L., Orme, C., et al.: Scenescript: Reconstructing scenes with an autoregressive structured language model. In: European Conference on Computer Vision. pp. 247–263. Springer (2024)
5. Bercovich, A., Levy, I., Golan, I., Dabbah, M., El-Yaniv, R., Puny, O., Galil, I., Moshe, Z., Ronen, T., Nabwani, N., et al.: Llama-nemotron: Efficient reasoning models. arXiv preprint arXiv:2505.00949 (2025)
6. Çelen, A., Han, G., Schindler, K., Van Gool, L., Armeni, I., Obukhov, A., Wang, X.: I-design: Personalized llm interior designer. In: European Conference on Computer Vision. pp. 217–234. Springer (2024)
7. Chang, A., Dai, A., Funkhouser, T., Halber, M., Niessner, M., Savva, M., Song, S., Zeng, A., Zhang, Y.: Matterport3d: Learning from rgb-d data in indoor environments. arXiv preprint arXiv:1709.06158 (2017)
8. Chang, A., Dai, A., Funkhouser, T., Halber, M., Niessner, M., Savva, M., Song, S., Zeng, A., Zhang, Y.: Matterport3d: Learning from rgb-d data in indoor environments. arXiv preprint arXiv:1709.06158 (2017)
9. Cover, T.M.: Elements of information theory. John Wiley & Sons (1999)
10. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5828–5839 (2017)
11. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5828–5839 (2017)
12. Deitke, M., VanderBilt, E., Herrasti, A., Weihs, L., Ehsani, K., Salvador, J., Han, W., Kolve, E., Kembhavi, A., Mottaghi, R.: Procthor: Large-scale embodied ai using procedural generation. Advances in Neural Information Processing Systems **35**, 5982–5994 (2022)
13. Deng, J., Chai, W., Huang, J., Zhao, Z., Huang, Q., Gao, M., Guo, J., Hao, S., Hu, W., Hwang, J.N., et al.: Citycraft: A real crafter for 3d city generation. arXiv preprint arXiv:2406.04983 (2024)
14. Ding, N., Chen, Y., Xu, B., Qin, Y., Zheng, Z., Hu, S., Liu, Z., Sun, M., Zhou, B.: Enhancing chat language models by scaling high-quality instructional conversations. arXiv preprint arXiv:2305.14233 (2023)
15. Fang, C., Dong, Y., Luo, K., Hu, X., Shrestha, R., Tan, P.: Ctrl-room: Controllable text-to-3d room meshes generation with layout constraints. In: 2025 International Conference on 3D Vision (3DV). pp. 692–701. IEEE (2025)
16. Fang, C., Li, H., Liang, Y., Zheng, J., Mao, Y., Liu, Y., Tang, R., Zhou, Z., Tan, P.: Spatialgen: Layout-guided 3d indoor scene generation. arXiv preprint arXiv:2509.14981 **3** (2025)

17. Fang, C., Li, H., Liang, Y., Zheng, J., Mao, Y., Liu, Y., Tang, R., Zhou, Z., Tan, P.: Spatialgen: Layout-guided 3d indoor scene generation. arXiv preprint arXiv:2509.14981 **3** (2025)
18. Feng, W., Zhu, W., Fu, T.j., Jampani, V., Akula, A., He, X., Basu, S., Wang, X.E., Wang, W.Y.: Layoutgpt: Compositional visual planning and generation with large language models. Advances in Neural Information Processing Systems **36**, 18225–18250 (2023)
19. Fu, H., Cai, B., Gao, L., Zhang, L.X., Wang, J., Li, C., Zeng, Q., Sun, C., Jia, R., Zhao, B., et al.: 3d-front: 3d furnished rooms with layouts and semantics. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10933–10942 (2021)
20. Fu, H., Jia, R., Gao, L., Gong, M., Zhao, B., Maybank, S., Tao, D.: 3d-future: 3d furniture shape with texture. International Journal of Computer Vision **129**(12), 3313–3337 (2021)
21. Fu, R., Wen, Z., Liu, Z., Sridhar, S.: Anyhome: Open-vocabulary generation of structured and textured 3d homes. In: European Conference on Computer Vision. pp. 52–70. Springer (2024)
22. Gao, G., Liu, W., Chen, A., Geiger, A., Schölkopf, B.: Graphdreamer: Compositional 3d scene synthesis from scene graphs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21295–21304 (2024)
23. Gu, Z., Cui, Y., Li, Z., Wei, F., Ge, Y., Gu, J., Liu, M.Y., Davis, A., Ding, Y.: Artiscene: Language-driven artistic 3d scene generation through image intermediary. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 2891–2901 (2025)
24. Hu, S., Wu, W., Su, R., Hou, W., Zheng, L., Xu, B.: Raster-to-graph: Floorplan recognition via autoregressive graph prediction with an attention transformer. In: Computer Graphics Forum. vol. 43, p. e15007. Wiley Online Library (2024)
25. Hua, B.S., Pham, Q.H., Nguyen, D.T., Tran, M.K., Yu, L.F., Yeung, S.K.: Scenenn: A scene meshes dataset with annotations. In: 2016 fourth international conference on 3D vision (3DV). pp. 92–101. Ieee (2016)
26. Hua, B.S., Pham, Q.H., Nguyen, D.T., Tran, M.K., Yu, L.F., Yeung, S.K.: Scenenn: A scene meshes dataset with annotations. In: 2016 fourth international conference on 3D vision (3DV). pp. 92–101. Ieee (2016)
27. Huang, I., Bao, Y., Truong, K., Zhou, H., Schmid, C., Guibas, L., Fathi, A.: Fireplace: Geometric refinements of llm common sense reasoning for 3d object placement. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 13466–13476 (2025)
28. Hugging Face: Open r1: A fully open reproduction of deepseek-r1 (January 2025), `https://github.com/huggingface/open-r1`
29. Hurst, A., Lerer, A., Goucher, A.P., Perelman, A., Ramesh, A., Clark, A., Ostrow, A., Welihinda, A., Hayes, A., Radford, A., et al.: Gpt-4o system card. arXiv preprint arXiv:2410.21276 (2024)
30. Inoue, N., Kikuchi, K., Simo-Serra, E., Otani, M., Yamaguchi, K.: Layoutdm: Discrete diffusion model for controllable layout generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10167–10176 (2023)
31. Li, Z., Yu, T.W., Sang, S., Wang, S., Song, M., Liu, Y., Yeh, Y.Y., Zhu, R., Gundavarapu, N., Shi, J., et al.: Openrooms: An end-to-end open framework for photorealistic indoor scene datasets. arXiv preprint arXiv:2007.12868 (2020)

32. Li, Z., Yu, T.W., Sang, S., Wang, S., Song, M., Liu, Y., Yeh, Y.Y., Zhu, R., Gundavarapu, N., Shi, J., et al.: Openrooms: An open framework for photorealistic indoor scene datasets. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 7190–7199 (2021)
33. Lin, C., Mu, Y.: Instructscene: Instruction-driven 3d indoor scene synthesis with semantic graph prior. arXiv preprint arXiv:2402.04717 (2024)
34. Lin, T.Y., Lin, C.H., Cui, Y., Ge, Y., Nah, S., Mallya, A., Hao, Z., Ding, Y., Mao, H., Li, Z., et al.: Genusd: 3d scene generation made easy. In: ACM SIGGRAPH 2024 Real-Time Live! pp. 1–2 (2024)
35. Ling, L., Lin, C.H., Lin, T.Y., Ding, Y., Zeng, Y., Sheng, Y., Ge, Y., Liu, M.Y., Bera, A., Li, Z.: Scenethesis: A language and vision agentic framework for 3d scene generation. arXiv preprint arXiv:2505.02836 (2025)
36. Mao, Y., Zhong, J., Fang, C., Zheng, J., Tang, R., Zhu, H., Tan, P., Zhou, Z.: Spatiallm: Training large language models for structured indoor modeling. arXiv preprint arXiv:2506.07491 (2025)
37. McCormac, J., Handa, A., Leutenegger, S., Davison, A.J.: Scenenet rgb-d: 5m photorealistic images of synthetic indoor trajectories with ground truth. arXiv preprint arXiv:1612.05079 (2016)
38. McCormac, J., Handa, A., Leutenegger, S., Davison, A.J.: Scenenet rgb-d: Can 5m synthetic images beat generic imagenet pre-training on indoor segmentation? In: 2017 IEEE International Conference on Computer Vision (ICCV). pp. 2697–2706 (2017). https://doi.org/10.1109/ICCV.2017.292
39. Nauata, N., Chang, K.H., Cheng, C.Y., Mori, G., Furukawa, Y.: House-gan: Relational generative adversarial networks for graph-constrained house layout generation. In: European Conference on Computer Vision. pp. 162–177. Springer (2020)
40. Nauata, N., Hosseini, S., Chang, K.H., Chu, H., Cheng, C.Y., Furukawa, Y.: House-gan++: Generative adversarial layout refinement network towards intelligent computational agent for professional architects. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 13632–13641 (2021)
41. Öcal, B.M., Tatarchenko, M., Karaoğlu, S., Gevers, T.: Sceneteller: Language-to-3d scene generation. In: European Conference on Computer Vision. pp. 362–378. Springer (2024)
42. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al.: Training language models to follow instructions with human feedback. Advances in neural information processing systems **35**, 27730–27744 (2022)
43. Paschalidou, D., Kar, A., Shugrina, M., Kreis, K., Geiger, A., Fidler, S.: Atiss: Autoregressive transformers for indoor scene synthesis. Advances in neural information processing systems **34**, 12013–12026 (2021)
44. Penedo, G., Lozhkov, A., Kydlíček, H., Allal, L.B., Beeching, E., Lajarín, A.P., Gallouédec, Q., Habib, N., Tunstall, L., von Werra, L.: Codeforces cots. https://huggingface.co/datasets/open-r1/codeforces-cots (2025)
45. Purkait, P., Zach, C., Reid, I.: Sg-vae: Scene grammar variational autoencoder to generate new indoor scenes. In: European Conference on Computer Vision. pp. 155–171. Springer (2020)
46. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of machine learning research **21**(140), 1–67 (2020)
47. Raistrick, A., Mei, L., Kayan, K., Yan, D., Zuo, Y., Han, B., Wen, H., Parakh, M., Alexandropoulos, S., Lipson, L., et al.: Infinigen indoors: Photorealistic indoor

scenes using procedural generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21783–21794 (2024)
48. Ran, X., Li, Y., Xu, L., Yu, M., Dai, B.: Direct numerical layout generation for 3d indoor scene synthesis via spatial reasoning. arXiv preprint arXiv:2506.05341 (2025)
49. Ritchie, D., Wang, K., Lin, Y.a.: Fast and flexible indoor scene synthesis via deep convolutional generative models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6182–6190 (2019)
50. Roblox, F.A.T.: Cube: A roblox view of 3d intelligence. arXiv preprint arXiv:2503.15475 (2025)
51. Shi, Y., Shang, M., Qi, Z.: Intelligent layout generation based on deep generative models: A comprehensive survey. Information Fusion **100**, 101940 (2023)
52. Sun, F.Y., Liu, W., Gu, S., Lim, D., Bhat, G., Tombari, F., Li, M., Haber, N., Wu, J.: Layoutvlm: Differentiable optimization of 3d layout via vision-language models. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 29469–29478 (2025)
53. Tang, J., Nie, Y., Markhasin, L., Dai, A., Thies, J., Nießner, M.: Diffuscene: Denoising diffusion models for generative indoor scene synthesis. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 20507–20518 (2024)
54. Team, G.: Gemma 3 technical report. ArXiv **abs/2503.19786** (2025), `https://api.semanticscholar.org/CorpusID:277313563`
55. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023)
56. Wang, K., Lin, Y.A., Weissmann, B., Savva, M., Chang, A.X., Ritchie, D.: Planit: Planning and instantiating indoor scenes with relation graph and spatial prior networks. ACM Transactions on Graphics (TOG) **38**(4), 1–15 (2019)
57. Wang, X., Yeshwanth, C., Nießner, M.: Sceneformer: Indoor scene generation with transformers. In: 2021 International conference on 3D vision (3DV). pp. 106–115. IEEE (2021)
58. Wang, Y., Qiu, X., Liu, J., Chen, Z., Cai, J., Wang, Y., Wang, T.H., Xian, Z., Gan, C.: Architect: Generating vivid and interactive 3d scenes with hierarchical 2d inpainting. Advances in Neural Information Processing Systems **37**, 67575–67603 (2024)
59. Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q.V., Zhou, D., et al.: Chain-of-thought prompting elicits reasoning in large language models. Advances in neural information processing systems **35**, 24824–24837 (2022)
60. Wei, Q.A., Ding, S., Park, J.J., Sajnani, R., Poulenard, A., Sridhar, S., Guibas, L.: Lego-net: Learning regular rearrangements of objects in rooms. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19037–19047 (2023)
61. Wu, W., Fu, X.M., Tang, R., Wang, Y., Qi, Y.H., Liu, L.: Data-driven interior plan generation for residential buildings. ACM Transactions on Graphics (TOG) **38**(6), 1–12 (2019)
62. Xia, H., Li, X., Li, Z., Ma, Q., Xu, J., Liu, M.Y., Cui, Y., Lin, T.Y., Ma, W.C., Wang, S., et al.: Sage: Scalable agentic 3d scene generation for embodied ai. arXiv preprint arXiv:2602.10116 (2026)
63. Xiao, G., Tian, Y., Chen, B., Han, S., Lewis, M.: Efficient streaming language models with attention sinks. arXiv preprint arXiv:2309.17453 (2023)

64. Xiao, J., Owens, A., Torralba, A.: Sun3d: A database of big spaces reconstructed using sfm and object labels. In: Proceedings of the IEEE international conference on computer vision. pp. 1625–1632 (2013)
65. Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., et al.: Qwen3 technical report. arXiv preprint arXiv:2505.09388 (2025)
66. Yang, H., Zhang, Z., Yan, S., Huang, H., Ma, C., Zheng, Y., Bajaj, C., Huang, Q.: Scene synthesis via uncertainty-driven attribute synchronization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5630–5640 (2021)
67. Yang, M.J., Guo, Y.X., Zhou, B., Tong, X.: Indoor scene generation from a collection of semantic-segmented depth images. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15203–15212 (2021)
68. Yang, R., Zhu, Z., Li, Y., Huang, J., Yan, S., Zhou, S., Liu, Z., Li, X., Li, S., Wang, W., et al.: Visual spatial tuning. arXiv preprint arXiv:2511.05491 (2025)
69. Yang, Y., Jia, B., Zhang, S., Huang, S.: Sceneweaver: All-in-one 3d scene synthesis with an extensible and self-reflective agent. arXiv preprint arXiv:2509.20414 (2025)
70. Yang, Y., Jia, B., Zhi, P., Huang, S.: Physcene: Physically interactable 3d scene synthesis for embodied ai. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16262–16272 (2024)
71. Yang, Y., Sun, F.Y., Weihs, L., VanderBilt, E., Herrasti, A., Han, W., Wu, J., Haber, N., Krishna, R., Liu, L., et al.: Holodeck: Language guided generation of 3d embodied ai environments. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16227–16237 (2024)
72. Yu, L.F., Yeung, S.K., Tang, C.K., Terzopoulos, D., Chan, T.F., Osher, S.J.: Make it home: Automatic optimization of furniture arrangement. ACM Trans. Graph. **30**(4), 86 (2011)
73. Zhai, G., Örnek, E.P., Wu, S.C., Di, Y., Tombari, F., Navab, N., Busam, B.: Commonscenes: Generating commonsense 3d indoor scenes with scene graph diffusion. Advances in Neural Information Processing Systems **36**, 30026–30038 (2023)
74. Zhang, G., Wang, Y., Luo, C., Xu, S., Zhang, Z., Zhang, M., Peng, J.: Furniscene: A large-scale 3d room dataset with intricate furnishing scenes. arXiv preprint arXiv:2401.03470 (2024)
75. Zhang, Y., Cai, Z., Wang, M., Guo, M., Li, T., Lin, L., Wang, Y.: M3dlayout: A multi-source dataset of 3d indoor layouts and structured descriptions for 3d generation. arXiv preprint arXiv:2509.23728 (2025)
76. Zheng, J., Zhang, J., Li, J., Tang, R., Gao, S., Zhou, Z.: Structured3d: A large photo-realistic dataset for structured 3d modeling. In: European Conference on Computer Vision. pp. 519–535. Springer (2020)
77. Zhong, W., Cao, P., Jin, Y., Luo, L., Cai, W., Lin, J., Wang, H., Lyu, Z., Wang, T., Dai, B., et al.: Internscenes: A large-scale simulatable indoor scene dataset with realistic layouts. arXiv preprint arXiv:2509.10813 (2025)