

HomeWorld: A Unified Floorplan-to-Furnished Framework for Generating Controllable, Densely Interactive Whole-Home Scenes

Wenbo Li^{1,2*}, Xiaoliang Ju^{1,2*†}✉, Zipeng Qin^{2*},
Rongyao Fang², and Hongsheng Li^{2,1,3}✉

¹Ace Robotics, Shanghai, China ²CUHK MMLab, Hong Kong, China

³Shenzhen Loop Area Institute, Shenzhen, China

akira@link.cuhk.edu.hk, hsli@ee.cuhk.edu.hk

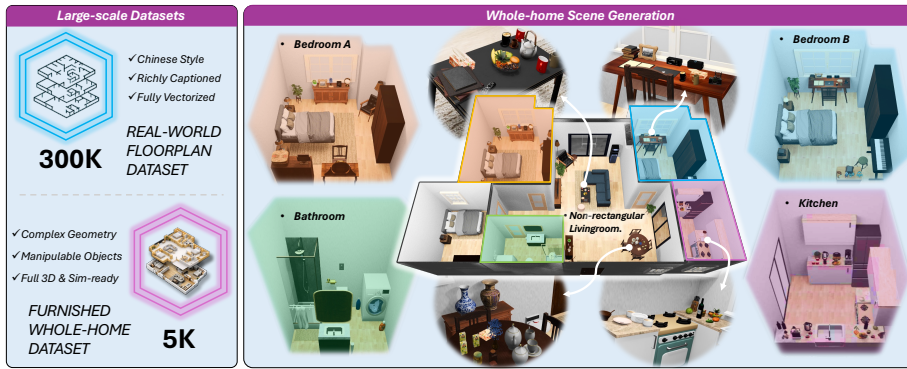


Fig. 1: We propose a large-scale indoor scene dataset (left) along with a whole-home scene generator; a representative generated example is shown on the right.

Abstract. Indoor scene generation is crucial for robot simulation and modern interior design. However, complex layouts together with scarce 3D scene data make learning-based generation challenging. Existing methods often rely on hand-crafted rules or focus on isolated sub-tasks (e.g., floorplan synthesis or single-room furnishing), producing whole-home scenes that lack global coherence, realism, and simulation readiness. To mitigate these limitations, we propose a unified hierarchical framework that decomposes indoor scene synthesis into controllable stages. First, we curate a large-scale dataset of 300K real residential floorplans to train a large language model for whole-home floorplan generation. With detailed descriptions and a K-D tree-based representation, our method enables fine-grained, controllable whole-home floorplan generation. Building upon the generated whole-home floorplan, we leverage image generation models to draft furniture layouts from multi-level roaming viewpoints, and then generate the layouts of small manipulable objects on different supporting surfaces (e.g. cabinets, desks, and dining tables) for embodied AI simulation. During furniture and object layout generation, a VLM-based refiner iteratively corrects furniture and object placement,

*Equal contribution. †Project lead. ✉Corresponding authors.

and a 3D generative model enables flexible replacement of individual assets. We further attach basic physical attributes and simple surface texture and lighting setups to complete the pipeline for embodied AI use. Experiments and user studies demonstrate that our pipeline produces indoor spaces with greater layout diversity and stronger 3D design appeal, outperforming prior methods on both quantitative and qualitative metrics. Finally, alongside our generation pipeline, we will release the floorplan dataset and 5K fully furnished scenes to the community. Project page: <https://kairos-homeworld.github.io/>.

Keywords: Whole-home indoor scene generation · Hierarchical 3D scene synthesis · Large-scale 3D scene dataset

1 Introduction

Indoor scene generation is critical for simulation in embodied AI and for modern interior design workflows. As embodied learning and generalist policies advance, agents are expected to navigate, interact, and complete increasingly complex tasks, raising the bar for virtual environments. The target is therefore shifting from isolated, single-room scenes to complete, multi-room homes with globally coherent structure and interactive surroundings. Meeting this demand calls for whole-home scene generation that is not only visually realistic, but also functionally plausible and physically feasible for simulation.

However, scaling to such whole-home generation is fundamentally constrained by data: large-scale, high-fidelity 3D residential scenes are still scarce in public datasets. Existing resources are dominated either by rule-based synthetic datasets [11, 22, 36], whose layouts and object distributions are biased by hand-designed priors, or by reconstruction-based datasets [6, 10], which often suffer from incomplete geometry and fragmental surfaces that make reliable layout extraction and asset grounding difficult. As a result, both the quantity and the quality of available 3D data are insufficient to effectively train learning-based generative models for diverse, realistic, and physically feasible multi-room homes.

Existing generative methods struggle under such data regime. Approaches that learn directly in 3D are typically limited by the quantity and fidelity of training scenes, and are computationally expensive, making it difficult to efficiently generate large-scale, whole-home environments [18, 29]. Meanwhile, rule-based pipelines like [11] can guarantee basic validity, but tend to produce repetitive layouts and biased object distributions. Another line of work exploits 2D priors from image generation model to lift room imagery to 3D such as [14]. However, without a strong 3D prior to anchor the generation, it is difficult to preserve geometric consistency when scaling to whole-home scenes. In practice, the lifted 3D outputs are often incomplete and fragmented, exhibiting artifacts commonly seen in reconstruction-based 3D data.

To overcome the data bottleneck while meeting the requirements of whole-home, interactive simulation, we propose a unified hierarchical generation framework that combines strong, scalable priors with explicit 3D grounding. Instead of

learning whole-home 3D synthesis end-to-end from scarce 3D datasets, we first learn to generate globally coherent 2D floorplan, and then progressively instantiate it into a physically feasible 3D environment with furniture and manipulable objects. A key observation is that even 2D floorplan data—despite being easier to obtain than full 3D scenes—is still relatively scarce in large scale and often lacks clean, standardized representations for learning. We therefore curate and annotate a large dataset of 300K real residential floorplans, and use it to train an LLM for fine-grained, controllable floorplan generation with caption supervision and a K-D tree-based representation.

Conditioned on the generated floorplan, we instantiate the 3D home with a view-roaming, prior-guided synthesis stage. We leverage the diversity and commonsense plausibility of foundation image generation models to propose new objects from roaming camera views, while using 3D constraints derived from the floorplan (e.g., room boundaries, walls, and free space) to anchor the proposals and enforce cross-view 3D consistency. To make the process stable and semantically structured, we adopt a hierarchical roaming policy: we first place large, function-defining furniture under a top-down view (e.g., beds and sofas), and then descend to egocentric views to progressively enrich the scene with smaller items. Finally, to better support embodied AI simulation, we distribute manipulable objects throughout the home. Throughout this process, a VLM-based recursive refiner corrects implausible placements and constraint violations, and a 3D generative module enables flexible asset replacement to increase object diversity and rendering variation without breaking scene coherence.

In summary, our contributions are two-fold:

- **A whole-home indoor scene generation pipeline.** We propose a unified hierarchical pipeline for controllable whole-home generation under limited 3D data. It (i) trains LLMs to generate precise and controllable floorplans via a K-D tree representation and detailed captions; (ii) leverages foundation image models as strong 2D priors to propose objects through hierarchical view roaming while maintaining 3D consistency and physical adherence; (iii) employs a VLM-based recursive refiner to correct construction errors; and (iv) supports flexible 3D asset replacement via a 3D generative module to increase object diversity and rendering variation.
- **A large-scale whole-home indoor scene dataset.** We curate and will publicly release a dataset of **300K** annotated real-world residential floorplans, together with **5K** fully furnished, high-quality whole-home 3D scene samples, providing a new dataset that directly mitigates the data scarcity bottleneck in whole-home layout generation. We plan to continuously expand this dataset in subsequent releases.

2 Related Work

Indoor scene dataset. We summarize representative indoor scene datasets in Tab. 1 according to data source, scene scope, simulator readiness, curation strategy, and scale. Early real-world floorplan datasets, including LI-FULL [19],

Table 1: Indoor Scene Resources Comparison. **Rec.** denotes reconstruction-based real-world datasets. **S./H.** denote the number of furnished **Scenes** (individual, often room-level areas) and **Homes** (unified, with multiple rooms). **Sim-ready** denotes whether a dataset provides fully 3D scenes that are directly instantiable in a simulator/rendering engine and supports object-level manipulation. **MObj.** reports manipulable objects per scene on average when available. **n/r** means not reported and “-” means not applicable. **Collection** indicates datasets aggregated from multiple sources.

PART A: STATIC / RELEASED DATASET							
Resource	Source / Training Data	Scope	Sim-ready	Curation	Data Scale / Scalability		
					Floorplan	Furnished S./H.	MObj.
LI-FULL [19]	Real	Home	✗	Raw	5M	-	-
RPLAN [28]	Real	Home	✗	Manual	80K	-	-
MSD [26]	Real	Home	✗	Manual	5.3K	-	-
ResPlan [1]	Real	Home	✗	Manual	17K	-	-
ScanNet [10]	Real(Rec.)	Partial	✗	Manual	-	1.5K S.	n/r
Matterport3D [6]	Real(Rec.)	Home	✗	Manual	90	90 H.	n/r
3D-FRONT [13]	Designed	Home	✓	Manual	6.8K	6.8K H.	n/r
Structured3D [36]	Designed	Home	✗	Programmatic	3.5K	21K S.	n/r
InternScenes [37]	Collection	Mixed	Partial	Hybrid	-	48K S.	8
SceneVerse [17]	Collection	Mixed	Partial	Hybrid	-	68K S.	n/r
ProcTHOR-10K [11]	Synthetic(Rules)	Home	✓	Programmatic	10K	10K H.	n/r*
Ours (Released)	FP.(Real) + Obj(Gen.)	Home	✓	Hybrid	300K	5K H.	>15
PART B: GENERATION PIPELINE							
Floorplan-LLaMA [35]	Collection	Home	✗	LLM	✓	✗	✗
Floorplan-Diffusion [30]	Real	Home	✗	Diffusion	✓	✗	✗
LayoutGPT [12]	None	Partial	Partial	LLM	✗	✓	✗
Holodeck [34]	Collection	Home	✓	LLM	✓	✓	✗
LayoutVLM [24]	Designed	Partial	Partial	VLM	✗	✓	✗
ProcTHOR [11]	Synthetic(Rules)	Home	✓	Programmatic	✓	✓	✗
Infinigen [21]	Synthetic(Rules)	Home	✓	Programmatic	✓	✓	✗
PhyScene [32]	Designed	Home	✓	Diffusion	✓	✓	✗
ChOrD [23]	Designed	Home	✓	Diffusion	✓	✓	✗
EmbodiedGen [27]	Collection	Mixed	✓	Diffusion	✗	✓	✗
Ours (Pipeline)	Real floorplan only	Home	✓	MLLM	✓	✓	✓

* ProcTHOR-10K does not report manipulable-object counts per scene; it provides an asset library containing 1633 interactive instances.

RPLAN [28], MSD [26], and ResPlan [1], provide large-scale residential layouts, but are mainly limited to 2D floorplans and are therefore not directly usable for embodied simulation. Reconstruction-based datasets, such as ScanNet [10] and Matterport3D [6], offer real 3D scans with semantic annotations, yet they often cover partial spaces and lack instance-separated, manipulable assets required by interactive simulators.

Designed or synthetic datasets partially address this limitation by providing furnished 3D scenes. 3D-FRONT [13] contains curated residential scenes with explicit object instances and renderable assets, while Structured3D [36] offers large-scale synthetic scenes with ground-truth structure, but is not primarily designed for interactive manipulation. Recent aggregation datasets, such as InternScenes [37] and SceneVerse [17], improve scene diversity by combining heterogeneous sources, although simulator readiness and manipulability remain inconsistent across subsets. ProcTHOR [11] provides procedurally generated houses directly usable in simulation, together with an interactable asset library of 1633

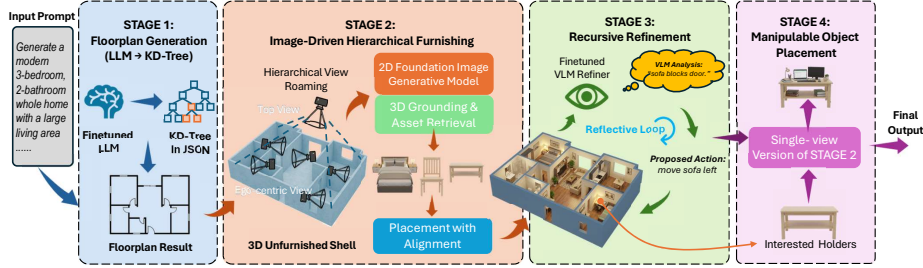


Fig. 2: Overview of the whole-home indoor scene generation pipeline. Our system turns a text prompt into a complete indoor scene in four stages: 1) build a large-scale floorplan dataset and train a prompt-conditioned generator; 2) given the floorplan, create an unfurnished 3D shell and progressively place furniture via hierarchical roaming with grounding/retrieval; 3) improve the scene using a finetuned VLM-based refiner to fix violations in a reflective loop; 4) add manipulable objects in feasible surface tops (e.g. dining tables, desks, cabinets, etc.).

instances. Overall, existing datasets still trade off scale, realism, and simulation readiness. Our dataset targets this gap by combining large-scale real floorplans with generated object assets to build furnished, sim-ready 3D houses with hybrid curation.

3D indoor scene generation. Existing 3D indoor scene generation methods can be roughly divided into four categories. Procedural methods rely on pre-defined grammars, priors, or handcrafted constraints [11, 21], which offer controllability but are limited by finite rules and asset libraries. Direct 3D-domain generation methods [2, 16, 18, 29] better preserve geometric constraints, yet scaling them to whole-home scenes remains difficult due to high computational cost and limited 3D training data. Compositional methods first predict layouts and then populate them with assets. Some recent works use LLMs for whole-home floorplan and asset layout generation [5, 20, 34], but their spatial reasoning ability remains limited, leading to imperfect plausibility and diversity. Many methods instead focus on room-level generation [3, 12, 24, 25, 33]; however, independently generated rooms often struggle to maintain global coherence in functional adjacency, style, and scale. Another line of work incrementally constructs scenes through roaming or 2D-lifting strategies [8, 9, 14], but the lack of explicit 3D constraints often causes geometric inconsistency and limits simulation deployment. Representative generation pipelines are also summarized in Tab. 1.

Our pipeline follows a hierarchical compositional paradigm while incorporating roaming-based generation. Unlike prior methods, we enforce explicit 3D constraints and introduce a reflective correction loop to improve geometric consistency. Moreover, we explicitly target simulation-ready environments with dense manipulable objects by generating complete homes from floorplans, to furniture layouts, and finally to small object arrangements.

3 Methodology

As illustrated in Fig. 2, our system enables users to describe a desired home layout with a natural-language prompt and automatically synthesizes a complete indoor scene. The workflow can be divided into four stages. First, we describe how we build a real-world large-scale floorplan dataset and train a prompt-conditioned floorplan generator. Second, conditioned on the floorplan, we introduce a hierarchical roaming strategy to populate the scene progressively. Before starting, we first build a 3D unfurnished shell as the explicit 3D constraint for the roaming. Then the system prioritizes the placement of large, room-defining furniture, and then adds more objects, using grounding and asset retrieval to ensure geometric and semantic consistency. Third, we employ a recursive refinement loop based on a finetuned VLM, which inspects the rendered scene to detect violations (e.g., collisions or blocked pathways), and then acts as a refiner agent that proposes and executes corrective actions. Finally, we use the similar strategy as Stage 2 to add manipulable objects in user-specified areas of interest.

3.1 Floorplan Dataset Curation and Floorplan Generation

Current floorplan datasets are limited in scale, complexity, diversity, and realism [1, 26, 28]. Yet for applications such as robot navigation, a large-scale and high-fidelity dataset is essential to enable robust generalization to real-world environments. In this section, we develop an automatic procedure for extracting vectorized data from carefully-curated large-scale real-world floorplan images from professional architectural portfolios. We also devise a K-D tree representation for effective training of floorplan generation model.

Floorplan dataset curation. As shown in Fig. 3(a), we construct the floorplan dataset through a five-stage pipeline. First, we collect 1.084M raw floorplan images from an online real-estate repository. Second, we train a geometry recognition model on 2K manually annotated samples to detect walls, doors, windows, and related architectural elements, and convert the detections into vectorized geometric primitives. Third, we apply OCR to extract auxiliary information, including room labels and dimensional annotations. Fourth, we perform filtering and post-processing to remove noisy samples, merge fragmented structures, and validate the consistency of room geometry, semantics, and architectural elements. Finally, we convert each validated floorplan into a unified JSON format and automatically generate structural captions describing room types, boundaries, connectivity, and attachments. The resulting dataset contains 314K fully vectorized floorplans paired with detailed textual descriptions, which are used to train the LLM-based floorplan generator.

LLM-based floorplan generator with K-D tree representation. We formulate floorplan generation as a text-to-structure prediction task. Given a natural-language design specification, including apartment boundary, room program, adjacency relations, positional constraints, and opening requirements, the LLM predicts a structured JSON representation rather than directly outputting room polygons. The output encodes the floorplan as a binary K-D tree together with doors, windows, and other attachments.

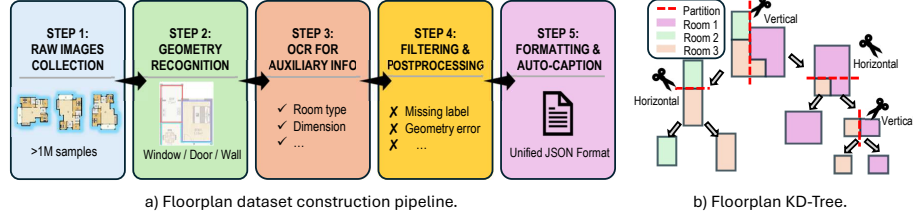


Fig. 3: (a) Floorplan dataset construction pipeline and (b) the K-D Tree representation for floorplan used in our generator.

Specifically, the interior space is recursively partitioned into axis-aligned sub-regions by vertical or horizontal cuts (Fig. 3(b)). Each internal node stores the split axis and coordinate, while each leaf node represents a spatial cell with its room identity and semantic type. This hierarchical serialization is naturally compatible with autoregressive LLM generation and provides a constrained output space that reduces geometric errors such as room overlap. The predicted tree can be deterministically converted into conventional floorplan representations for downstream use. We fine-tune the LLM on the curated floorplan dataset, with ablation results reported in Sec. 4.3.

3.2 Furniture Layout Generation by Hierarchical Roaming

We adopt a 2D-lifting approach realized through hierarchical indoor roaming, primarily because existing 3D furniture-layout datasets are insufficient in both quantity and quality as aforementioned. We leverage the strong semantic priors of 2D foundation generative models to enable open-vocabulary object placement with improved semantic plausibility. Our system then grounds the generated results into precise 3D bounding boxes, parameterized by semantic label, physical dimension and pose. The roaming consists of two stages: *top-down roaming* initializes coarse object placement at the scene level, while *ego-centric roaming* focuses on local regions to refine object geometry.

Unfurnished shell as 3D constraints for roaming. Before the roaming starts, we first instantiate an empty, unfurnished shell in Blender with basic lighting and material settings. This shell is used to impose explicit 3D constraints during roaming, which is especially important for irregular, non-rectangular room layouts as shown in Fig. 1. With the consistent geometry, it reduces perspective-induced artifacts and 3D inconsistencies caused by fully free roaming (e.g., drifting geometry in Text2Room [14]). In addition, the lighting-driven shading provides a natural prior for the subsequent 2D generation stage, leading to more coherent appearance and smoother view-to-view transitions.

Global layout generation via top-down view.

The key idea is to exploit a structure-aware top-down view as an intermediate representation, since major furniture placement is largely governed by global constraints: room boundary, doors, windows, and inter-room connectivity. Given the empty shell, we render a top-down view image I_{top} . We then annotate I_{top} with doors and windows, and for each door we also mark the adjacent room, because these cues strongly influence the furniture layout. Conditioned on these

structural priors, we apply an image inpainting model to I_{top} to synthesize a furnished top view $\hat{I}_{\text{top}}$, leveraging 2D foundation priors for more realistic, prompt-controllable furniture arrangements. After a lightweight validation step to discard infeasible generations, we use a VLM for open-vocabulary category recognition and SAM-3 [4] for instance masks and 2D boxes. This initial layout of major furniture will be further refined in the following stage.

Detail refinement via ego-centric view. While the global top-down view provides a reliable initialization for major furniture, it is inherently limited by occlusion and scale ambiguity, making it difficult to recover small objects and wall-attached items (e.g., bins, hooks, racks, and upper/lower cabinets in kitchens). We therefore introduce an ego-centric roaming stage to enrich the scene with fine-grained details.

Viewpoint selection. To cover the entire room with limited renders, we compute a compact set of viewpoints using a heatmap-based viewpoint selection strategy. Concretely, we discretize the floor plane into a grid and sample candidate camera poses along the room boundary and interior, subject to a fixed field-of-view and a safe distance to walls. Each candidate view contributes visibility to a subset of grid cells; we accumulate these contributions into a coverage heatmap. We then greedily select viewpoints by repeatedly choosing the view that covers the currently least-observed regions (equivalently, maximally reduces the uncovered heat), updating the heatmap after each selection.

Ego-centric synthesis and 3D grounding. For each viewpoint k , we render an ego-centric image I_k and apply inpainting to synthesize additional secondary objects, producing $\hat{I}_k$. We then use SAM-3 [4] to extract instance masks for newly introduced objects and employ SAM-3D [7] to reconstruct their 3D geometry. We apply a geometric alignment module to integrate these newly reconstructed 3D assets into the existing room environment. By moving the viewpoint, this iterative ego-centric roaming process provides 360-degree coverage, resulting in a densely populated and highly realistic 3D indoor environment.

3.3 Recursive layout refinement

Generated room layouts may contain physical, semantic, and structural violations, such as object collisions, boundary crossings, blocked doorways, implausible rotations, and floating objects. To address these errors, we introduce a lightweight VLM-based refiner that performs closed-loop verification and correction for each candidate layout. The module is trained to iteratively identify layout errors and predict structured corrective actions.

Iterative refinement formulation. For each candidate layout, we render a top-view image and pair it with a structured 3D layout representation containing object categories, positions, sizes, rotations, and architectural elements. Given these multimodal inputs, the VLM refiner evaluates the layout with respect to physical plausibility, semantic consistency, and structural validity. At each iteration, it predicts a corrective action specifying the target object and the required edit, such as translation, rotation, or a combined transformation. The action is

then deterministically applied to the 3D bounding-box layout, followed by re-rendering and re-evaluation. The loop terminates when all validation criteria are satisfied or a predefined iteration limit is reached.

Fine-tuning data construction. We construct a supervision corpus for multi-step layout correction from three sources. First, we perturb valid room layouts to synthesize corrupted states covering common generation errors, including collisions, boundary violations, doorway blocking, abnormal rotations, scale inconsistencies, and category errors. The recorded perturbation trajectories provide reverse supervision for repair. Second, for each corrupted state, an oracle generates candidate corrections, such as translation, rotation, combined transformation, and minimum-separation displacement. A verifier scores these candidates by residual errors, newly introduced violations, movement cost, and semantic plausibility, and selects the best one as the ground-truth next action. Third, to improve closed-loop robustness, we roll out an initial refiner, collect its intermediate states, relabel them with the oracle, and add them back to training. Each sample contains a top-view rendering, structured 3D layout description, current error state, optional action history, and oracle-labeled correction action.

3.4 Manipulable Objects Placement

Although the top-down and ego-centric roaming stages recover a coherent room layout, large supporting surfaces are often left unnaturally bare. Instead of randomly placing small objects, we synthesize surface clutter via image inpainting and then parse it into structured layouts, leveraging learned priors on realistic object co-occurrence and spatial arrangement.

Concretely, for each primary furniture item, we condition an inpainting model on the *room type* and *furniture name* to generate a surface image that hallucinates plausible small-object compositions. Given this synthesized image and the known furniture size, a VLM parses the surface into a layout of manipulable objects in the furniture-local frame, which is then transformed into global coordinates using the pose of the supporting furniture for seamless integration with the overall 3D layout.

4 Experiment

Fig. 4 provides an intuitive overview of our whole-home scene generation pipeline. Given a natural-language prompt, our method progressively generates a floorplan, instantiates an unfurnished 3D shell, produces a furnished 3D scene, and finally enriches the environment with manipulable objects. Following this pipeline, our experiments are organized into three parts. First, we evaluate floorplan generation to demonstrate the effectiveness of our proposed K-D Tree-based representation on our dataset. Second, we report results on floorplan-conditioned whole-home generation. Third, we conduct a comprehensive ablation study. Both quantitative and qualitative evaluations show that our method outperforms existing approaches.

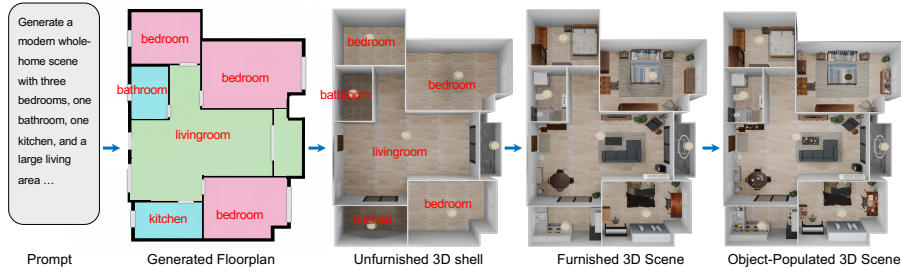


Fig. 4: Overview of the whole-home scene generation.

4.1 Floor Plan Generation

Experiment setup. We split our dataset into 281K training samples, 15K validation samples, and 15K test samples. We finetune Qwen3-4B-Instruct [31] using supervised finetuning with LoRA [15] on our training corpus.

We formulate floorplan synthesis as a constrained instruction-following task. The prompt specifies following factors of the target floorplan: a) dimensions; b) room count and types; c) door/window placement; d) a structural output format requiring a K-D Tree with axis-aligned (horizontal or vertical) partitions; e) spatial constraints regarding room adjacencies and connections.

We provide examples of our prompts in the supplementary.

Evaluation metrics. We introduce two quantitative metrics to assess the generated floorplans: 1) *graph validity*, the fraction of generated floorplans whose connectivity graph is fully connected, measuring basic topological navigability; 2) *graph diversity*: the number of distinct connectivity graph structures generated for a fixed room count, measuring topological variety. We report typed diversity (room types included) and untyped diversity (topology only).

Table 2: Quantitative comparison of floorplan generation. Our model generates more accurate and diverse room topologies than the selected baseline.

	Graph Validity	Graph Diversity	
		Typed	Untyped
Procthor-10K [11]	72.6%	6.2	2.3
Ours	87.7%	22.4	6.9

Quantitative results. We randomly sample 500 floorplans from our finetuned Qwen3 model and compare them against generated samples from Procthor-10K [11]. For graph validity, our model achieves a 15.1% improvement over Procthor-10K (see Tab. 2 column 1), demonstrating its ability to learn accurate opening placements (e.g., doors) that yield fully connected floorplan topology. For both typed and untyped graph diversity, we compute the numbers of distinct connectivity graphs for samples containing one to ten rooms - a range representative of typical residential layouts - and report the average. As shown in Tab. 2 columns 2 and 3, our model consistently outperforms the baseline across on the graph diversity, suggesting that our model successfully learns diverse, real-world strategies for constructing room connection topologies from the training data.

For a qualitative comparison of our results, we present it in the supplementary due to limited space provided.

User study. We conduct a human evaluation study with 24 participants to assess the perceived quality of generated floorplans. We choose the recently proposed FloorPlan-LLaMA [35] and Floorplan-Diffusion [30] as our compared methods. Participants are presented with three generated floorplans from the three methods and asked to rank them along two dimensions: a) *Reasonability* for practical feasibility, realistic room flows, and constructibility; and b) *Richness* for appropriate organizational richness without unnecessary complication to support multiple daily uses. To ensure consistent interpretation, all floorplans are rendered using a standardized color palette encoding room types and openings. Rankings are converted to numerical scores: 3, 2, and 1 points for first, second, and third place, respectively. These scores are aggregated across participants and averaged as shown in Tab. 3. Our generated results consistently achieve the highest quality across both evaluation dimensions.

Table 3: User study of floorplan generation. Our generated results achieve superior quality on both *Reasonability* and *Richness* dimensions.

	Reasonability	Richness	Average
FloorPlan-LLaMA [35]	1.85	2.00	1.93
Floorplan-Diffusion [30]	2.13	1.81	1.97
Ours	2.20	2.34	2.27

4.2 Furniture Layout Generation

Experiment setup. We exploit Blender to perform all 3D template initialization and perspective rendering in our automated 3D scene generation pipeline. For 2D image inpainting in both top-down and ego-centric view roaming, we use Gemini 3.0 Image Pro Preview (aka Nano Banana Pro). For instance-level detection and dense pixel-wise segmentation, we employ SAM3 [4]. We then use SAM 3D(Object) [7] to obtain the initial object geometry and pose.

We evaluate our method against the following baselines: LayoutGPT [12], Holodeck [34], and LayoutVLM [24], representing existing methods for open-universe layout generation.

Evaluation Metrics. We evaluate generated 3D layouts along two dimensions: **physical plausibility** and **distributional properties**. For physical plausibility, we report the *Collision Ratio* (CR) and the *Out-of-Boundary Ratio* (OOB). CR quantifies severe inter-object penetrations (excluding semantically valid contacts), and OOB measures the fraction of object footprints outside the room boundary. For distributional properties, we report *Volume Density* (VD) and *Footprint Object Density* (FOD), where VD is total 3D box volume normalized by floor area and FOD is the object count normalized by the union area of all furniture footprints, reflecting how densely objects occupy the floor plan. Detailed definitions and equations are provided in the supplementary material.

Qualitative results. We visualize predicted layouts in both whole-home and single functional room settings. Fig. 5 compares whole-home results; among the



Fig. 5: Comparison of whole-home generation.

evaluated baselines, only Holodeck [34] and our method support whole-home generation. For single-room generation, we provide additional visual comparisons for the living room/bedroom in Fig. 6, and kitchen / bathroom in Fig. 7, respectively. Note that LayoutGPT [12] is limited to living-room and bedroom generation, and thus is excluded from kitchen and bathroom comparisons.

Quantitative results. To evaluate controllable room-layout generation across common residential spaces, we generate four room categories: living room, bedroom, bathroom, and kitchen, with 25 samples per method per category. As shown in Tab. 4, our approach consistently outperforms prior methods on the metrics defined above. These quantitative gains align with qualitative observations: our generated layouts exhibit fewer severe overlaps and yield more diverse, natural spatial arrangements.

Table 4: Quantitative comparison of layout generation. Our generated results achieve superior quality.

Method	Physical Plausibility		Density & Diversity	
	CR↓	OOB↓	VD↑	FOD↑
LayoutGPT [12]	0.05	0.01	0.22	1.99
Holodeck [34]	0.07	0.02	0.22	2.10
LayoutVLM [24]	0.20	0.01	0.27	2.15
Ours	0.05	0.01	0.35	4.16

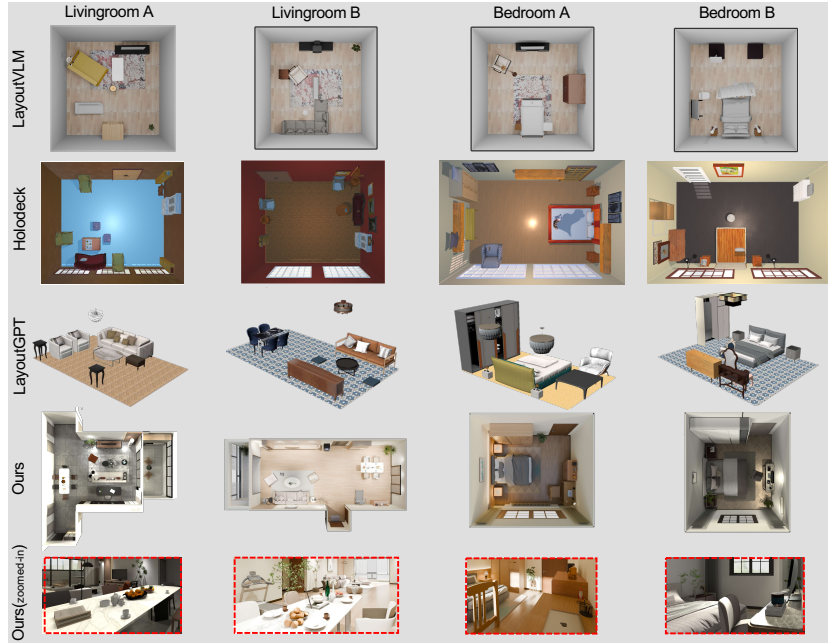


Fig. 6: Comparison of generated functional rooms: livingroom and bedroom.

User Study. Additionally, we conducted a user study with 30 users. For each method, we sampled 10 results spanning both the whole-home setting and four single-room categories: living room, bedroom, bathroom, and kitchen.

Participants were shown each sample generated by 4 methods, and asked to rank them from best to worst along three dimensions: (a) *Reasonability* for practical feasibility, realistic room flows, and constructibility; (b) *Aesthetics* for visual legibility, balance, and design coherence; and (c) *Complexity* for appropriate organizational richness without unnecessary complication. As reported in Tab. 5, our method is preferred by the majority of participants across all questions and scene categories.

Table 5: User study of layout generation. Our generated results achieve superior quality on *Reasonability*, *Aesthetics* and *Complexity*.

	Reasonability	Aesthetics	Complexity	Average
LayoutGPT [12]	0.202	0.184	0.153	0.180
Holodeck [34]	0.395	0.412	0.452	0.420
LayoutVLM [24]	0.252	0.260	0.283	0.265
Ours	0.807	0.827	0.797	0.811

4.3 Ablation Study

Ablation on Floorplan Generation(*w/o K-D Tree representation*). Empirically, under the canonical representation, 9.2% of the 500 generated floorplans contain overlapping room polygons, and 2.0% exhibit holes within the apartment outlines. Moreover, the proportion of floorplans with disconnected rooms

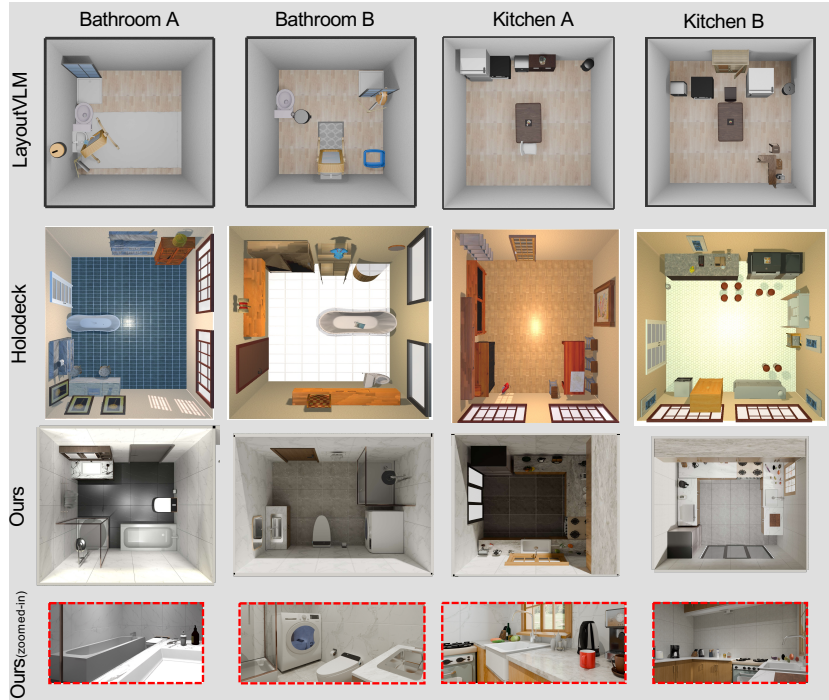


Fig. 7: Comparison of generated functional rooms: bathroom and kitchen.

increases from 12.3% using our K-D Tree formulation to 16.2% under the canonical representation, indicating a clear degradation in the reliability and structural coherence of floorplan generation.

Table 6: Ablation study on the Layout Generation.

Method	Physical Plausibility		Density & Diversity	
	CR↓	OOB↓	VD↑	FOD↑
w/o Top-down View Roaming	0.09	0.02	0.22	2.6
w/o Ego-centric View Roaming	0.07	0.02	0.20	3.44
w/o VLM-based Refiner	0.20	0.05	0.33	3.02
w/o Manipulable Object Placement	0.05	0.01	0.33	1.82
Ours	0.05	0.01	0.35	4.16

Ablation on Layout Generation. We conduct an ablation study by disabling one module at a time while keeping all remaining components, prompts, and the evaluation protocol unchanged. We compare four variants against the full system (**Ours**) in Tab. 6, reporting physical plausibility (**CR**↓, **OOB**↓) and distributional properties (**VD**↑, **FOD**↑).

1) *w/o top-down view roaming.* We remove the structure-aware top-down stage (top-view inpainting and the top-down 2D-to-3D initialization) and rely on the remaining pipeline. This reduces scene density (**VD** drops from **0.35** to

0.22), indicating that the top-down stage provides crucial global structure priors for both feasibility and coarse furniture placement.

2) *w/o ego-centric view roaming*. We disable ego-centric roaming and generate layouts only from the top-down initialization. The plausibility slightly degrades, while **VD** decreases (**0.35** $\rightarrow$ **0.20**), showing that ego-centric roaming is important for enriching furniture-level content.

3) *w/o VLM refiner*. We turn off the recursive refinement loop (no multi-view consistency checking or iterative box corrections). This significantly harms physical plausibility (**CR/OOB** rises to **0.20/0.05**), demonstrating that the refiner is critical for correcting structural and physical violations produced by synthesis and grounding.

4) *w/o manipulable object placement*. We remove manipulable surface-object placement while keeping the rest of the pipeline unchanged. Physical plausibility remains on par with **Ours**, but **FOD** drops sharply (**4.16** $\rightarrow$ **1.82**) and **VD** slightly decreases (**0.35** $\rightarrow$ **0.33**), indicating that object placement mainly increases the number of small, manipulable objects without sacrificing feasibility.

Overall, the top-down and ego-centric stages improve improves global feasibility and coarse density, object placement substantially increases object-level richness (especially FOD), and the VLM-based refiner is essential for ensuring physical plausibility.

5 Conclusion

In this work, we present HomeWorld, a unified hierarchical framework for controllable whole-home scene generation from natural-language prompts. It first synthesizes fine-grained residential floorplans with an LLM trained on large-scale real floorplans using a structured K-D Tree representation, and then progressively instantiates them into furnished 3D homes through hierarchical view roaming, recursive refinement, and surface-centric manipulable object placement. By combining scalable 2D generative priors with explicit 3D grounding and iterative correction, HomeWorld produces coherent furniture layouts, realistic small-object arrangements, and simulation-ready scenes with physical attributes, textures, and lighting. Experiments and user studies show that our method generates more valid and diverse floorplans, as well as more plausible and richer 3D layouts, than prior methods. We will also release 300K annotated real-world floorplans and 5K high-quality furnished whole-home scenes to support future research on controllable indoor scene generation and embodied AI simulation.

Acknowledgements

This work was partially supported by the National Key R&D Program of China (No. 2025ZD0122902), the NSFC-RGC Project N_CUHK498/24, the Guangdong Basic and Applied Basic Research Foundation (No. 2023B1515130008), the Shenzhen Loop Area Institute, and the Centre for Perceptual and Interactive Intelligence (CPII) Ltd under the Innovation and Technology Commission (ITC)’s InnoHK.

References

1. Abouagour, M., Garyfallidis, E.: Resplan: A large-scale vector-graph dataset of 17,000 residential floor plans. arXiv preprint arXiv:2508.14006 (2025) [4](#), [6](#)
2. Bokhovkin, A., Meng, Q., Tulsiani, S., Dai, A.: Scenefactor: Factored latent 3d diffusion for controllable 3d scene generation. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 628–639 (2025) [5](#)
3. Bucher, M.J., Armeni, I.: Respace: Text-driven 3d indoor scene synthesis and editing with preference alignment. arXiv preprint arXiv:2506.02459 (2025) [5](#)
4. Carion, N., Gustafson, L., Hu, Y.T., Debnath, S., Hu, R., Suris, D., Ryali, C., Alwala, K.V., Khedr, H., Huang, A., et al.: Sam 3: Segment anything with concepts. arXiv preprint arXiv:2511.16719 (2025) [8](#), [11](#)
5. 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) [5](#)
6. 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) [2](#), [4](#)
7. Chen, X., Chu, F.J., Gleize, P., Liang, K.J., Sax, A., Tang, H., Wang, W., Guo, M., Hardin, T., Li, X., et al.: Sam 3d: 3dfy anything in images. arXiv preprint arXiv:2511.16624 (2025) [8](#), [11](#)
8. Chen, Y., Nguyen, H.T., Voleti, V., Jampani, V., Jiang, H.: Housecrafter: Lifting floorplans to 3d scenes with 2d diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 28440–28450 (2025) [5](#)
9. Chung, J., Lee, S., Nam, H., Lee, J., Lee, K.M.: Luciddreamer: Domain-free generation of 3d gaussian splatting scenes. arXiv preprint arXiv:2311.13384 (2023) [5](#)
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) [2](#), [4](#)
11. Deitke, M., VanderBilt, E., Herrasti, A., Weihs, L., Ehsani, K., Salvador, J., Han, W., Kolve, E., Kembhavi, A., Mottaghi, R.: Proctor: Large-scale embodied ai using procedural generation. Advances in Neural Information Processing Systems **35**, 5982–5994 (2022) [2](#), [4](#), [5](#), [10](#)
12. 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) [4](#), [5](#), [11](#), [12](#), [13](#)
13. 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) [4](#)
14. Höllein, L., Cao, A., Owens, A., Johnson, J., Nießner, M.: Text2room: Extracting textured 3d meshes from 2d text-to-image models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7909–7920 (2023) [2](#), [5](#), [7](#)
15. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al.: Lora: Low-rank adaptation of large language models. Iclr **1**(2), 3 (2022) [10](#)
16. Huang, Z., Guo, Y.C., An, X., Yang, Y., Li, Y., Zou, Z.X., Liang, D., Liu, X., Cao, Y.P., Sheng, L.: Midi: Multi-instance diffusion for single image to 3d scene

- generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 23646–23657 (2025) 5
17. Jia, B., Chen, Y., Yu, H., Wang, Y., Niu, X., Liu, T., Li, Q., Huang, S.: Sceniverse: Scaling 3d vision-language learning for grounded scene understanding. In: European Conference on Computer Vision. pp. 289–310. Springer (2024) 4
 18. Ju, X., Huang, Z., Li, Y., Zhang, G., Qiao, Y., Li, H.: Diffindscene: Diffusion-based high-quality 3d indoor scene generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4526–4535 (2024) 2, 5
 19. LIFULL Co., Ltd., National Institute of Informatics (NII): LIFULL HOME’S High Resolution Floor Plan Image Data. <https://www.nii.ac.jp/dsc/idr/en/lifull/1.html> (2017), accessed: 2026-02-28 3, 4
 20. Liu, X., Tang, C.K., Tai, Y.W.: Worldcraft: Photo-realistic 3d world creation and customization via llm agents. arXiv preprint arXiv:2502.15601 (2025) 5
 21. 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) 4, 5
 22. Savva, M., Chang, A.X., Dosovitskiy, A., Funkhouser, T., Koltun, V.: Minos: Multimodal indoor simulator for navigation in complex environments. arXiv preprint arXiv:1712.03931 (2017) 2
 23. Su, C., Fu, Y., Hu, Z., Yang, J., Hanji, P., Wang, S., Zhao, X., Öztireli, C., Zhong, F.: Chord: Generation of collision-free, house-scale, and organized digital twins for 3d indoor scenes with controllable floor plans and optimal layouts. arXiv preprint arXiv:2503.11958 (2025) 4
 24. Sun, F.Y., Liu, W., Gu, S., Lim, D., Bhat, G., Tombari, F., Li, M., Haber, N., Wu, J.: Layoutvln: Differentiable optimization of 3d layout via vision-language models. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 29469–29478 (2025) 4, 5, 11, 12, 13
 25. 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) 5
 26. Van Engelenburg, C., Mostafavi, F., Kuhn, E., Jeon, Y., Franzen, M., Standfest, M., van Gemert, J., Khademi, S.: Msd: A benchmark dataset for floor plan generation of building complexes. In: European Conference on Computer Vision. pp. 60–75. Springer (2024) 4, 6
 27. Wang, X., Liu, L., Cao, Y., Wu, R., Qin, W., Wang, D., Sui, W., Su, Z.: Embodiedgen: Towards a generative 3d world engine for embodied intelligence. arXiv preprint arXiv:2506.10600 (2025) 4
 28. 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 (SIGGRAPH Asia) 38(6) (2019) 4, 6
 29. Wu, Z., Li, Y., Yan, H., Shang, T., Sun, W., Wang, S., Cui, R., Liu, W., Sato, H., Li, H., et al.: Blockfusion: Expandable 3d scene generation using latent tri-plane extrapolation. ACM Transactions on Graphics (ToG) 43(4), 1–17 (2024) 2, 5
 30. Xu, M., Lou, Y., Gao, X., Zhou, X.: Floorplan-diffusion: Automatic floor plan generation via pre-trained large latent diffusion model. In: Proceedings of the 2025 International Conference on Multimedia Retrieval. pp. 1617–1625 (2025) 4, 11

31. Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., Zheng, C., Liu, D., Zhou, F., Huang, F., Hu, F., Ge, H., Wei, H., Lin, H., Tang, J., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Zhou, J., Lin, J., Dang, K., Bao, K., Yang, K., Yu, L., Deng, L., Li, M., Xue, M., Li, M., Zhang, P., Wang, P., Zhu, Q., Men, R., Gao, R., Liu, S., Luo, S., Li, T., Tang, T., Yin, W., Ren, X., Wang, X., Zhang, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Zhang, Y., Wan, Y., Liu, Y., Wang, Z., Cui, Z., Zhang, Z., Zhou, Z., Qiu, Z.: Qwen3 technical report. arXiv preprint arXiv:2505.09388 (2025) [10](#)
32. 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) [4](#)
33. Yang, Y., Lu, J., Zhao, Z., Luo, Z., Yu, J.J., Sanchez, V., Zheng, F.: Llplace: The 3d indoor scene layout generation and editing via large language model. arXiv preprint arXiv:2406.03866 (2024) [5](#)
34. 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) [4](#), [5](#), [11](#), [12](#), [13](#)
35. Yin, J., Zeng, P., Sun, H., Dai, Y., Zheng, H., Zhang, M., Zhang, Y., Lu, S.: Floorplan-llama: Aligning architects’ feedback and domain knowledge in architectural floor plan generation. In: Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 6640–6662 (2025) [4](#), [11](#)
36. 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) [2](#), [4](#)
37. 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) [4](#)