RoomPlanner: Reachability-Aware View Sampling for Text-to-Room 3D Gaussian Splatting

Wenzhuo Sun[1*], Mingjian Liang[1*], Wenxuan Song[2], Xuelian Cheng[1,3✉], and Zongyuan Ge[1]

[1] Monash University, Clayton VIC, Australia
wenzhuo.sun@monash.edu, mingjian.liang2000@gmail.com, Zongyuan.Ge@monash.edu
[2] Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China
songwenxuan0115@gmail.com
[3] Monash Suzhou Research Institute, Suzhou, China
Xuelian.Cheng@monash.edu

Abstract. In this paper, we propose *RoomPlanner*, the first fully automatic 3D room generation framework for painlessly creating realistic indoor scenes with only short text as input. Without any manual layout design or panoramic image guidance, our framework can generate explicit layout criteria for rational spatial placement. We begin by introducing a hierarchical structure of language-driven agent planners that can automatically parse short and ambiguous prompts into detailed scene descriptions. These descriptions include raw spatial and semantic attributes for each object and the background, which are then used to initialize 3D point clouds. To position objects within bounded environments, we implement two arrangement constraints that iteratively optimize spatial arrangements, ensuring a collision-free and accessible layout solution. In the final rendering stage, we propose a novel ReachView Sampling strategy for camera trajectory, along with the Interval Timestep Flow Sampling (ITFS) strategy, to efficiently optimize the coarse 3D Gaussian scene representation. These approaches help reduce the total generation time to under 30 minutes. Extensive experiments demonstrate that our method can produce geometrically rational 3D indoor scenes, surpassing prior approaches in both rendering speed and visual quality while preserving editability. The code will be available at https://kaitlina-s.github.io/RoomPlanner/.

1 Introduction

Text-to-3D scene generation aims to convert textual descriptions into 3D environments. This paradigm is intended to reduce cost and lower the expertise barrier, and it supports a range of downstream applications, such as embodied

[*] Equal contribution.
[✉] Corresponding author.

AI [2,34], gaming [43], filmmaking, and augmented/virtual reality [27]. Benefiting from the success of generative models [13,32] and implicit neural representations [17,24], many methods have advanced text-to-3D object generation and achieved substantial progress. However, extending from individual 3D assets to complete 3D scenes remains challenging. One key reason is that scene layout requires additional context and often involves substantial human effort. According to the sources from which layout information is obtained, existing text-to-3D scene generation methods can be broadly categorized as **I.** *visual-guided* methods and **II.** *rule-based* methods.

Visual-guided methods extract implicit spatial information from visual representations, *e.g.,* multi-view RGB images or panoramas. These methods typically adopt a two-stage framework in which image generation serves as an intermediate step, followed by an implicit neural representation network that reconstructs a 3D scene from the generated images. Some methods [15,20] provide the implicit rendering network with objects and background as an indivisible whole. Consequently, the resulting continuous representations lack object-level decoupling, which substantially limits practical use. These approaches can also produce foggy or distorted geometry, as shown in Fig. 1 (b). Rather than being directly constrained, the layout configuration is learned through a text-to-image generation model, which is optimized by aligning generated images with the input text using a similarity score. Therefore, the optimization of the scene configuration is bounded by the capability of the image generation model, which makes direct control of layout through textual descriptions difficult.

Rule-based methods construct layout information for 3D environments using predefined rules derived from physical relationships and constraints. Early methods [3, 8, 14, 23] rely on manually crafted prompt templates or human-designed layouts for initialization, which facilitates the generation of physically plausible scenes. This procedure requires specialized expertise and often involves a tedious trial-and-error process to refine predefined spatial attributes of objects, *e.g.,* position, size, and rotation *etc.* This human-in-the-loop paradigm increases the demand for human labor and can hinder end-to-end training. In a separate line of research that aims to reduce human effort in layout design, data-driven approaches formulate scene synthesis as a 3D asset arrangement problem within a bounded environment. Recently, LLMs have been introduced to automate the creation of 3D scenes [1,10,39,41,42]. LLMs translate unstructured language into structured spatial representations, which reduces dependence on rigid templates. However, reliance on existing 3D assets limits the versatility and flexibility required to generate diverse and novel scenes.

A simple yet effective approach is to couple layout composition with generative rendering [19, 38]. High-quality rendering requires effective view selection and reliable camera-trajectory sampling, but a semantically reasonable and collision-free layout can still prevent camera traversal within a room and limit approaches to target objects. This mismatch yields low-quality or missing object-centric views, degrades the observation set, and induces sticky or merged artifacts during downstream optimization. Furthermore, object-level editing requires

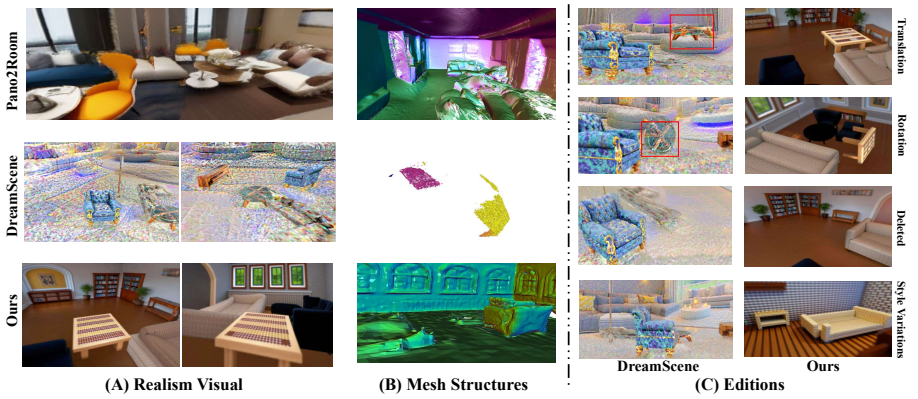


Fig. 1: Compared to previous methods, *i.e.,* the visual-guided method Pano2Room [31] and the rule-based method DreamScene [19], our approach generates indoor scenes with (a) higher realism, (b) smoother mesh structures, and (c) support for diverse editing operations, including rotation, translation, importing/deleting 3D assets, and style variations.

object-level decoupling and stable scene organization, which increases the need for explicit and controllable layout constraints. These considerations motivate a unified design that optimizes layout for rendering performance while maintaining editing flexibility, and couples camera planning with render-time scheduling.

In this paper, we present RoomPlanner, a fully automatic framework that generates differentiable 3D indoor scenes and explicit layout constraints from short textual prompts, without manual layout authoring or panoramic-image guidance. RoomPlanner is motivated by a central observation: high-quality rendering depends on effective camera-trajectory sampling, yet layouts produced from text or handcrafted rules often fail to guarantee camera traversal and coverage of key objects. This mismatch degrades the observation sampling and frequently leads to low-quality views for some objects and sticky or merged artifacts during downstream optimization. RoomPlanner addresses this issue by coupling layout design with rendering requirements and incorporating renderability as a first-class objective throughout generation.

First, RoomPlanner translates short prompts into an executable scene specification and a renderability-aware initial layout. The high-level planner identifies the target scene type and key objects from contextual semantics, while the low-level planner infers object and background scales and provides layout rules aligned with real-world dimensions. Given this specification, RoomPlanner uses Text to 3D Assets model [16,36] to create point-cloud 3D assets and places object assets in the scene via iterative optimization with continuous constraint checking and iterative updates. In addition to collision constraints that resolve overlaps, a reachability constraint incorporates traversability of camera trajectories into the layout objective, since spacing or occlusion relations can block traversal, reduce close-up coverage, and degrade the observation set, lowering identifiability for

downstream optimization. ReachView therefore constrains traversable camera space during layout optimization via A* path planning with agent-based collision checks, improving observation coverage before rendering optimization and reducing degradation from non-traversable layouts, as shown in Fig. 1 (c).

Next, RoomPlanner couples camera-trajectory sampling with optimization scheduling to improve local geometric detail and global semantic consistency in a single rendering pass. A rendering module integrates Interval Timestep Flow Sampling (ITFS) with reflection flow matching and hierarchical multi-timestep sampling. By leveraging a 2D diffusion prior and fine-grained scene descriptions produced by the LLM planners, the module produces continuous 3D representations with realistic appearance and rich semantics. ITFS and ReachView are coupled: close object-centric views prioritize earlier timestep intervals to sharpen geometry, while global views prioritize later intervals to improve cross-object semantic consistency. Finally, object–environment decoupling is achieved by inserting optimized objects into the scene to supplement environment generation, which reduces duplicated, non-editable artifacts and enables flexible editing of object affine parameters and text descriptions for objects and environments (e.g., repositioning, insertion or deletion, and style changes). Consequently, the proposed pipeline produces high-fidelity scenes in a single optimization stage and reduces overall runtime by nearly 2× compared to prior work.

2 Related Work

2.1 Text-to-3D Object Generation

Existing 3D scene generation methods can be broadly classified into two categories: **I.** *Visual-Guided*, and **II.** *Direct regression* methods. Visual-Guided methods [21, 35] rely on pretrained 2D diffusion models, which can lead to error accumulation during the subsequent image-to-3D conversion process. While MagicTailor [45] addresses the issue of semantic pollution in the text-to-image stage, the resulting continuous representations still struggle with object-level decoupling. Direct regression methods [16, 26] focus on generating 3D assets directly from input text. Although 3D object generation has made significant advancements, directly applying a similar philosophy to solve 3D scene generation is still very challenging. Due to the lack of high-quality text-to-3D scene datasets, it hinders the ability to implement a feed-forward network directly. Furthermore, incorporating layout information adds an additional layer of complexity to the scene synthesis process.

2.2 LLM driven 3D Scene Generation

As discussed in §Introduction, rule-based methods can utilize explicit layout information to generate accurate and diverse 3D layouts. However, designing an effortless mechanism for layout arrangement remains an open question in 3D scene synthesis. With the reasoning capabilities of LLMs, works such as LayoutGPT [7] and SceneTeller [28] can generate visual layouts and place existing 3D

assets for scene synthesis. However, because of the single-step planning strategy, their output often exhibits physical implausibilities, *e.g.,* floating objects or intersecting geometries. Additionally, they tend to perform suboptimally in complex scenes with numerous objects, mainly due to the absence of spatial constraints.

On the other hand, LLM-driven frameworks aim to reduce human effort in the 3D layout arrangement task. To improve layout rationality, recent reinforcement learning-based methods utilize constraint solvers to enhance physical validity. Holodeck [42] and its extensions [1, 11, 41] translate relational phrases into geometric constraints, which are optimized using gradient-based methods to eliminate collisions. Diffuscene [33] and Physcene [40] incorporate physics engines to simulate object dynamics, iteratively adjusting parameters derived from LLMs to ensure stability. Despite advances in realism, their reliance on existing 3D assets limits the versatility and flexibility needed for generating diverse and innovative scenes. Beyond these layout arrangement approaches, our model provides a fully automated pipeline for generating complete and controllable 3D scenes, including individual decoupling objects.

3 Preliminary

3D Layout Arrangement Given layout descriptions, the goal is to arrange unordered 3D assets from existing datasets within a bounded 3D environment. Formally, given a layout criterion φ_{layout}, a room space defined by four walls along cardinal directions $\{w_1, \ldots, w_4\}$, and a set of N 3D meshes $\{m_1, \ldots, m_N\}$, the objective is to synthesize a 3D scene that best satisfies the spatial relationships in the layout specification. Previous methods [1, 6, 42] formulate indoor layout as a 3D reasoning task and place objects according to open-ended language instructions. These methods typically assume upright input objects and use an off-the-shelf vision-language model (VLM), *e.g.,* GPT-4 [29], to estimate front-facing orientation. As a representative example, LayoutVLM [1] annotates each object with a concise text description s_i, and defines axis-aligned bounding-box size $b_i \in \mathbb{R}^3$ after rotating the object to face $+x$. The target output is object pose $p_i = (x_i, y_i, z_i, \theta_i)$, which contains 3D position and rotation around the z-axis.

Neural Representation Unlike explicit representations such as meshes or point clouds, implicit neural representations optimize a 3D model through differentiable rendering. Two common formulations are Neural Radiance Fields [24] and 3D Gaussian Splatting (3DGS) [17]. 3DGS represents a scene as a set of 3D Gaussians $\{\mathcal{G}_i\}_{i=1}^{M}$, where each Gaussian has center $\mu_i \in \mathbb{R}^3$, covariance $\mathbf{\Sigma}_i \in \mathbb{R}^{3\times3}$, color $c_i \in \mathbb{R}^3$, and opacity $\alpha_i \in \mathbb{R}^1$. For an offset Δx from center μ, the Gaussian kernel is written as $\mathcal{G}(\Delta x) = \exp\big(-\frac{1}{2}\Delta x^T \Sigma^{-1} \Delta x\big)$. The Gaussian parameters are optimized through differentiable rasterization by comparing rendered projections with training views using image-based losses. In text-to-3D settings, ground-truth (GT) images are usually unavailable. Existing methods therefore use diffusion priors to generate pseudo-GT images for distilling the 3D representation.

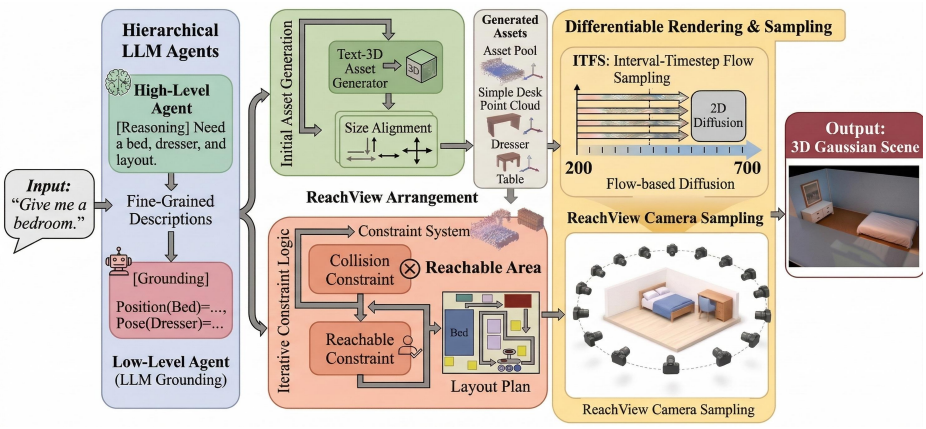


Fig. 2: RoomPlanner follows a 'Reasoning-Grounding, Arrangement, and Optimization' pipeline, decomposing the complex 3D scene generation task into scalable subtasks.

Diffusion Models Diffusion models [13, 32] generate data by iteratively denoising samples from pure noise. During training, the noisy sample $\mathbf{x}_t$ from data $\mathbf{x} \sim p_{\text{data}}$ is defined as $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\,\mathbf{x} + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is standard Gaussian noise and $\bar{\alpha}_t$ controls the noise level. Discrete diffusion timestep t is sampled from a uniform distribution $p_t \sim \mathcal{U}(0, t_{max})$. The denoising network θ predicts injected noise $\boldsymbol{\epsilon}_\theta$ and is optimized with the score-matching objective: $\mathcal{L}_t = \mathbb{E}_{\mathbf{x}\sim p_{\text{data}},\, t\sim p_t,\, \boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\left[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\|_2^2\right]$.

DreamFusion [30] distills 3D representations from a 2D text-to-image diffusion model through Score Distillation Sampling (SDS). Let θ denote parameters of a differentiable 3D representation, and let g denote a rendering function. For camera pose c, the rendered image is $x_0 = g(\theta, c)$, where x_0 is a clean (noise-free) rendering. A noisy view at timestep t is: $x_t = \alpha_t\, x_0 + \sigma_t\, \boldsymbol{\epsilon}$. SDS then distills θ through a frozen 2D diffusion model ϕ:

$$\nabla_\theta \mathcal{L}_{\text{SDS}}(\theta) = \mathbb{E}_{t,\boldsymbol{\epsilon},c}\left[w(t)\,(\boldsymbol{\epsilon}_\phi(\mathbf{x}_t; y, t) - \boldsymbol{\epsilon}) \times \frac{\partial g(\theta, c)}{\partial \theta}\right], \tag{1}$$

where $w(t)$ is a timestep-dependent weight and $\boldsymbol{\epsilon}_\phi$ is the noise estimator of the frozen diffusion model ϕ conditioned on the text prompt y.

4 Methodology

RoomPlanner is a fully automated pipeline that generates complete, complex, and controllable 3D indoor scenes from concise user prompts. As shown in Fig. 2, the framework consists of three stages. First, a hierarchical LLM-driven planner infers the target room type and produces detailed textual descriptions $\{y_i\}_{i=1}^N$ for

Algorithm 1 RoomPlanner Framework

Input: short-prompt description $\mathbf{I}$
Output: 3D room scene $\mathbf{Scene}^{\star}$
// High-Level Reasoning
$(\mathcal{O}, \mathcal{S}) \leftarrow \text{LLMPARSE}(\mathbf{I})$
// Low-Level Grounding
$\mathcal{P} \leftarrow \text{INITPOINTCLOUD}(\mathcal{O}, \mathcal{S})$
$(\hat{\mathcal{O}}, \hat{\mathcal{S}}) \leftarrow \text{ALIGNSIZES}(\mathcal{O}, \mathcal{S}, \mathcal{P})$
// Layout Arrangement I: Collision Constraint
$\varphi_{\text{layout}} \leftarrow \text{ARRANGE}(\hat{\mathcal{O}}, \hat{\mathcal{S}}, \mathcal{P})$
for $t_1 = 1$ to maxTimeIter **do**
    if not $\text{COLLISIONFREE}(\varphi_{\text{layout}})$ **then**
        $\varphi_{\text{layout}} \leftarrow \text{FIXLAYOUT}(\varphi_{\text{layout}}, \textit{collision})$
    else
        break ▷ collision-free layout $\hat{\varphi}_{\text{layout}}$ found
$\hat{\varphi}_{\text{layout}} \leftarrow \varphi_{\text{layout}}$
// Layout Arrangement II: Reachability Constraint
for $t_2 = 1$ to maxTimeIter **do**
    if not $\text{REACHABLE}(\hat{\varphi}_{\text{layout}})$ **then**
        $\hat{\varphi}_{\text{layout}} \leftarrow \text{FIXLAYOUT}(\hat{\varphi}_{\text{layout}}, \textit{reachable})$
    else
        break ▷ reachable layout $\varphi^{\star}_{\text{layout}}$ found
// Differentiable Scene Rendering
$\mathcal{T} \leftarrow \text{REACHVIEW}(\varphi^{\star}_{\text{layout}})$
$\mathbf{Scene}^{\star} \leftarrow \text{ITFS}(\varphi^{\star}_{\text{layout}}, \mathcal{T})$

individual assets together with a scene-level layout specification y_{scene}. Second, an initial 3D point cloud is generated for each asset via a text-to-3D object generator [16,36] conditioned on y_i, while a layout planner determines collision-free and traversable poses for the aligned assets, yielding a scene map that includes object coordinates, orientations, physical properties, and style descriptions. The assets are then placed into the room coordinate system according to the predicted poses. Finally, single-stage differentiable scene optimization is performed using Interval Timestep Flow Sampling together with ReachView camera trajectories constructed on the traversable map induced by the layout, producing realistic renderings with high 3D consistency.

4.1 Hierarchical LLM Agents Planning

Given a concise user prompt T, such as *"A bedroom"*, the high-level planner leverages the reasoning capability of LLM agents to semantically expand T and identify object categories that are appropriate for scene population. The planner selects object types that match the scene style implied by the prompt and produces an indexed set (O_i, S_i) for the room. Subsequently, a low-level LLM agent planner augments these assets with stylistic, textural, and material attributes.

For spatial grounding, text-to-3D object generator [16,36] is used to generate a point cloud representation $\mathbf{p}_i = (x_i, y_i, z_i, c_i)$ for each object O_i. Candidate center coordinates (x_i^c, y_i^c, z_i^c) are then estimated to initialize object positions in the scene. Each object O_i is associated with a semantic label. To ensure semantic and physical plausibility, a physical agent aligns the real-world scale of each object and rescales it when inconsistencies are detected. The updated object parameters and spatial context $(\hat{O}_i, \hat{S}_i)$ are serialized for downstream modules. Further details are provided in the supplementary material.

4.2 Layout Arrangement Planning

Taking the detailed layout criterion $\varphi_{\text{layout}} = (\hat{O}_i, \hat{S}_i, \mathbf{p}_i)$ as input, indoor scene generation is formulated as a layout arrangement problem in an enclosed space with four walls, following the strategy of a prior 3D layout arrangement method [42]. The process starts by constructing structural elements such as floors, walls, doors, and windows, followed by the placement of floor-mounted furniture and wall-mounted objects. The layout is then refined via interactive optimization by enforcing two constraints, as specified below.

Collision Constraint Given the spatial constraints φ_{layout} inferred by LLM agents, a depth-first search algorithm, inspired by Holodeck [42], is used to identify feasible placements for object candidates. Objects are placed sequentially according to inferred symbolic rules, such as positioning objects close to walls, aligning objects with other objects (*e.g., nightstands* beside a *bed*), or orienting objects in specified directions. To improve layout quality, wall-mounted object types are decoupled from ground-level object types while ensuring that both categories follow the same symbolic reasoning process. Each candidate placement is accepted only if it passes a conservative overlap test on object footprints, where boundary touching is permitted, and a room-boundary containment check. In addition, a small size buffer is applied during candidate generation to reduce near-collisions. Specifically, a collision-aware layout reward $\mathcal{R}_{\text{coll}}$ is defined to ensure physical feasibility:

$$\mathcal{R}_{\text{coll}} = -\Big(\sum_{1 \le i < j \le N} \text{IoU}_{3D}(b_i, b_j) + \sum_{i=1}^{N} \sum_{k=1}^{W} \text{IoU}_{3D}(b_i, b_k^{wall}) \Big), \tag{2}$$

where b_i and b_j represent the bounding boxes of objects, and b_k^{wall} denotes the bounding boxes of wall structures. IoU_{3D} measures the overlap between two candidate 3D bounding boxes. A layout is valid only when $\mathcal{R}_{\text{coll}} = 0$. In practice, colliding candidates are discarded during search, which is executed under a fixed time budget; objects that cannot be placed without collisions within this budget are omitted from the final layout.

Reachability Constraint Beyond collision avoidance, navigability is enforced by requiring all objects to be accessible to a virtual agent represented by the bounding box b^{agent}. **This reachability design directly benefits later camera**

trajectory planning, since the agent and the camera can share the same traversability map and shortest-path solver. Firstly, the 3D Scene Map is rasterized into a 2D traversable map, where occupied regions are conservatively expanded to account for the clearance of the agent and the camera. The A* search algorithm is then applied from a fixed starting point (*e.g.,* a doorway or an anchor asset) to a set of near-object contact locations. An object is considered reachable only if at least one valid path exists. The reachability reward $\mathcal{R}_{\text{reach}}$ is defined as:

$$\mathcal{R}_{\text{reach}} = -\sum_{i=1}^{N} \mathbf{1}\big[\pi_i = \emptyset\big], \tag{3}$$

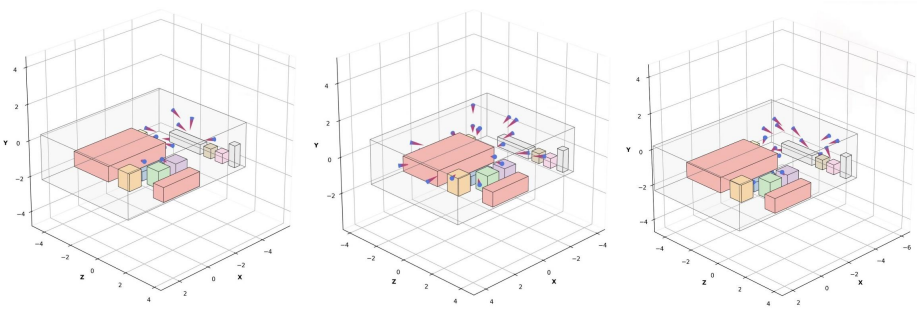
where π_i denotes the A* path to object i on the traversable map (and $\pi_i = \emptyset$ if A* fails). If an object is deemed unreachable, local resampling is applied within a restricted grid neighborhood during the search process; if no viable solution is identified within the time budget, the object is removed from the layout. This constraint also provides a feasible traversable map for subsequent ReachView trajectory generation. This integrated reasoning process ensures both physical plausibility and functional accessibility of the final layout $\varphi^{\star}_{\text{layout}}$.

4.3 Differentiable Scene Optimization

ReachView Camera Trajectory Sampling Unlike prior scene generators that rely on fixed or weakly constrained camera policies [8, 18, 19], ReachView samples cameras on the same traversable map used by the reachability constraint. We rasterize room bounds and object footprints into a 2D occupancy map, dilate occupied cells for camera clearance, and reject poses that violate room bounds, collide with assets, or lose line of sight to the target object. This coupling makes the rendering cameras consistent with the physically feasible layout instead of treating view selection as a post-processing step.

ReachView then decomposes supervision into two modes (Fig. 3). The scene-centric global walk starts from an anchor location such as a doorway, visits objects in nearest-neighbor order, and samples poses along gain-aware A* paths to preserve room-level context. The object-centric orbit samples upper-hemisphere close views around the current target to strengthen local geometry and appearance. The final hybrid trajectory interleaves these modes, producing the intended *"zoom-in and zoom-out"* behavior: optimization repeatedly alternates between holistic scene context and close object inspection rather than overusing either distant global views or isolated local views.

Interval Timestep Flow Sampling In the rendering stage of Fig. 2, an Interval-Timestep Flow Sampling (ITFS) strategy is introduced to improve the quality of raw 3D scene representations by using a range of timesteps to generate 2D diffusion views. The approach integrates ITFS with reflection flow matching and multi-timestep sampling to optimize scene quality from coarse to fine while enhancing semantic details (visualized in Fig. 5).



(a) Global Walk (b) Object Orbit (c) Hybrid

Fig. 3: Camera trajectory modes in ReachView. Red arrows denote camera view directions. The global walk covers room-level context, the object orbit provides close target views, and the hybrid trajectory interleaves both modes for balanced scene and object supervision.

Conditional Flow Matching (CFM) [22] provides an effective framework for training continuous normalizing flows due to computational efficiency and strong empirical performance. To obtain higher-fidelity priors, ϕ is instantiated with Stable Diffusion 3.5 [4], which adopts a rectified-flow formulation to model forward and reverse processes through a time-dependent vector field. The forward interpolation is defined as $x(t) = (1-t)\,x_0 + t\,\boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and the reverse field $v(x,t) = \frac{\partial x}{\partial t}$ is approximated by a neural network v_ϕ. The training process follows the CFM objective:

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t,\boldsymbol{\epsilon}}\left[w(t)\,\|v_\phi(x_t;t) - (\boldsymbol{\epsilon} - x_0)\|_2^2\right].$$

Building on the CFM objective and SDS, a **Flow Distillation Sampling (FDS)** strategy is introduced, with gradient $\nabla_\theta \mathcal{L}_{\text{FDS}}$ defined as:

$$\mathbb{E}_{t,\boldsymbol{\epsilon}}\Big[w(t)\big(v_\phi(x_t;y,t) - (\boldsymbol{\epsilon} - x_0)\big) \times \frac{\partial g(\theta,c)}{\partial \theta}\Big]. \tag{4}$$

Empirically, smaller timesteps (*e.g.,* $t \in [200, 300]$) emphasize geometric structures, whereas larger timesteps (*e.g.,* $t > 500$) favor semantic alignment, which is consistent with observations in DreamScene [19]. Since FDS samples only a single timestep, it does not exploit both regimes simultaneously. To address this limitation, ITFS aggregates supervision from multiple timestep intervals $\{t_i\}_{i=1}^m$ and optimizes:

$$\mathbb{E}_{\{t_i\},\boldsymbol{\epsilon}}\left[\sum_{i=1}^{m} w(t_i)\,\big(v_\phi(x_{t_i};y,t_i) - (\boldsymbol{\epsilon} - x_0)\big) \times \frac{\partial g(\theta,c)}{\partial \theta}\right], \tag{5}$$

where m denotes the number of sampled timesteps.

Fig. 4: Qualitative comparison on interactive indoor scene generation with Set-the-Scene [3], GALA3D [23], and DreamScene [19].

Mode-aware interval design. To align diffusion supervision with ReachView observations, mode-dependent interval selection is adopted. When the camera operates in the *Zoom-in* object-centric close-range mode, timesteps are sampled from $t \in [200, 600]$ (e.g., combining $[200, 400]$ and $[400, 600]$) to strengthen geometric fidelity and local physical details. When the camera follows the scene-centric global trajectory and the view covers multiple assets, timesteps are sampled from $t \in [400, 800]$ (e.g., combining $[400, 600]$ and $[600, 800]$) to reinforce semantic consistency and overall scene harmony. This multi-interval strategy enables early optimization steps to refine geometric features and later steps to consolidate semantic coherence, thereby producing 3D scenes that are both structurally accurate and visually consistent. Due to space constraints, detailed experimental results are provided in the supplementary material.

5 Experiments

Implementation Details We use GPT-4o [29] as the LLM agent for both reasoning and grounding. Shap-E [16] provides initial object point clouds, and Stable Diffusion 3.5 Medium [5] guides 3DGS optimization. Unless otherwise stated, all methods run on one NVIDIA RTX 4090 (24GB) with 1500 rendering iterations at 512×512 resolution.

5.1 Qualitative Results

The Fig. 4 presents side-by-side comparisons with state-of-the-art baselines. GALA3D [23] often shows limited asset diversity, while Set-the-Scene [3] exhibits

Type	Method	Layout Planning	End-to-End Generation	Room Scene	Edition	Realistic (object)	Realistic (scene)	Surface Reconstruction
Visual-Guided	Text2Room [14]	-	-	-	-	✓	✓	✓
	SceneWiz3D [44]	-	-	-	✓	-	-	✓
	Pano2Room [31]	-	-	-	✓	-	-	✓
	Director3d [20]	-	-	-	✓	-	-	✓
	Scene4U [15]	-	-	✓	-	-	-	✓
Rule-based	Set-the-Scene [3]	✓	-	-	-	✓	✓	✓
	GALA3D [23]	✓	✓	-	✓	✓	-	-
	AnyHome [8]	✓	-	✓	✓	✓	-	✓
	GraphDreamer [9]	✓	✓	-	✓	✓	-	-
	DreamScene [19]	✓	✓	-	✓	✓	✓	-
	Ours	✓	✓	✓	✓	✓	✓	✓

Table 1: Comparison of 3D indoor scene synthesis methods, categorized into rule-based and visual-guided approaches. ✓ means capability supported; '-' means without that capability. Our method demonstrates comprehensive capabilities across all metrics.

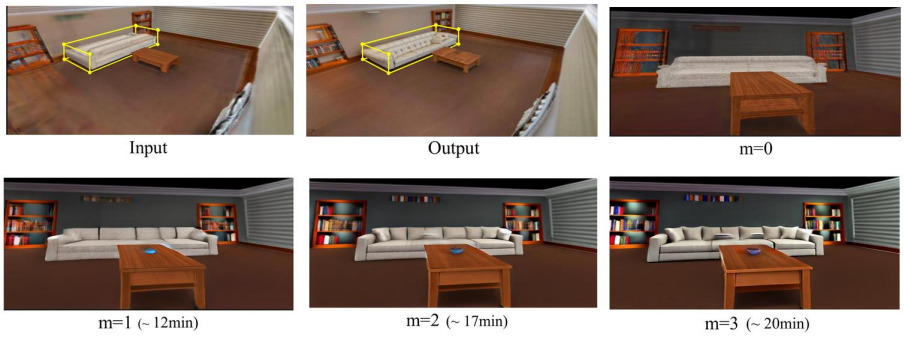


Fig. 5: Effectiveness of ITFS. The coach labels in yellow boxes guide the orientation towards a more rational perspective. As the timesteps progress from $m = 0$ to $m = 3$, the scene quality is optimized from a coarse representation to a fine depiction, enriched with more semantic details.

geometric discontinuities. DreamScene [19] tends to produce opacity artifacts and coarse surface geometry. In contrast, our method produces physically plausible scenes with cleaner object separation and stronger editability. Controlled runtime and policy comparisons are provided in the supplementary material.

Effectiveness of ITFS As illustrated in Fig. 5, our method leverages implicit layout knowledge obtained from a 2D generation model to further optimize the 3D scene. The proposed ITFS strategy significantly enhances scene generation by refining object orientation and detail. The coach labels in the two yellow boxes guide the orientation to achieve a more rational and coherent perspective. As the timesteps progress from $m = 0$ to $m = 3$, the scene quality is optimized from a coarse representation to a fine, detailed depiction. This progression not only improves geometric accuracy but also enriches the semantic details, resulting in a more realistic and contextually appropriate rendering.

Method	User Study (1-5)		Automatic Metrics		
	↑ Rationality	↑ Quality	↑ AS	↑ IR	↑ CS %
Set-the-Scene	1.13	2.80	4.83	41.10	26.05
GALA3D	2.61	3.22	5.07	42.67	25.40
DreamScene	3.10	3.15	5.13	43.85	26.77
Ours	**4.10**	**3.86**	**5.31**	**48.83**	**28.44**

Table 2: Quantitative evaluation on indoor scene generation. Rationality and Quality are user-study scores, while AS, IR, and CS are automatic image-text metrics.

5.2 Quantitative Results

User Study for Semantic Alignment We conduct a user study to evaluate scene rationality and visual quality. We select five common indoor scene types (*i.e.,* living room, kitchen, dining room, bedroom, and office). Thirty participants (professional interior designers, Embodied-AI engineers, and graduate students) score rendered scenes in randomized order on a 1–5 scale. We report the mean across participants. The evaluation includes:

- **Rationality:** physical plausibility and functional layout quality, including geometric consistency, collision-free placement, and human reachability.
- **Quality:** perceptual visual quality and text-scene consistency, including lighting, material realism, and semantic alignment.

As shown in Tab. 2, our method achieves the highest mean scores on both Rationality and Quality.

3D Scene Generation Quality As shown in Tab. 2, we also report three automatic metrics: Aesthetic Score (AS) [25], ImageReward (IR) [37], and CLIP Score (CS) [12]. Because these are 2D proxy signals on rendered views, we interpret them jointly with user-study and layout-centric metrics. Set-the-Scene [3], GALA3D [23], and DreamScene [19] obtain lower scores due to limited scene diversity and reduced rendering realism. Our method achieves the strongest overall performance under this protocol.

Layout Quality To evaluate scene layouts, we report two physical metrics in Table 3: collisions (Col) and out-of-bounds (OOB). We compare our method against three contemporaneous approaches. All methods use the same assets provided by a Text-3D generator controlled by the low-level agent across five scenes (living room, kitchen, dining room, bedroom, and workplace), and we report averages over these scenes.

Table 3: Layout quality summary. Col: colliding object pairs; OOB: out-of-bounds objects; Reach: % objects reachable via A* from the start anchor.

Method	**Col↓**	**OOB↓**	**Reach↑**
HOLODECK	0.00	0.87	97.3
LayoutVLM	11.09	7.53	86.2
Our	**0.00**	**0.20**	**100.0**

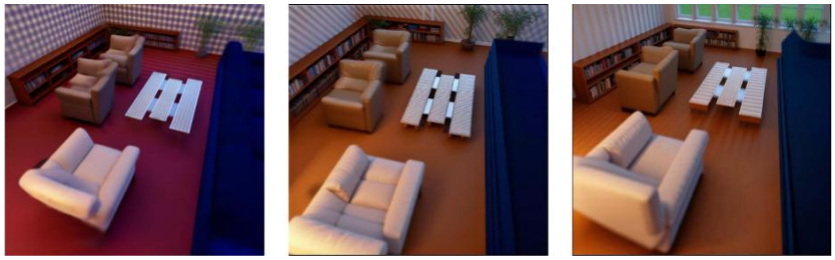


Fig. 6: Rendering results for bedroom and living room in the ITFS and ReachView ablation study.

Our method achieves physical metrics comparable to Holodeck and outperforms LayoutGPT [7] and LayoutVLM [1], indicating that the Reachability Constraint designed for ReachView effectively reduces collisions and out-of-bounds errors.

Ablation Study To evaluate the effectiveness of our core design, we conducted ablation experiments to isolate the contribution of each component. We considered three variants: (A) remove ReachView and ITFS and optimize the scene using conventional camera trajectories and SDS; (B) remove ReachView and render using ITFS only; and (C) render using both ReachView and ITFS. As shown in Figure 6, using only SDS leads to oversaturation/overexposure. When ReachView is omitted, scenes and objects lose detail and exhibit geometric instability. For example, doors and windows become less distinct and armchairs geometry becomes unstable.

6 Conclusion

We present RoomPlanner, a text-to-3D room generation framework that couples LLM-based scene specification, reachability-aware layout arrangement, and ReachView/ITFS-based rendering. Experiments show that this coupling improves scene rationality, visual quality, and editability while reducing runtime compared with prior pipelines.

Limitation Our current implementation is tuned for a single NVIDIA RTX 4090 (24GB). This memory budget limits Gaussian growth in later optimization stages, which can reduce fine light-transport and material detail. Future work will study more memory-efficient representations and broader analyses of view-budget efficiency and distillation stability.

7 Acknowledgments

This project is funded by Basic Research Program of Jiangsu (BK20250441).

References

1. Layoutvlm: Differentiable optimization of 3d layout via vision-language models. In: Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR). pp. 29469–29478 (June 2025)
2. Chen, Z., Walsman, A., Memmel, M., Mo, K., Fang, A., Vemuri, K., Wu, A., Fox, D., Gupta, A.: Urdformer: A pipeline for constructing articulated simulation environments from real-world images. arXiv preprint arXiv:2405.11656 (2024)
3. Cohen-Bar, D., Richardson, E., Metzer, G., Giryes, R., Cohen-Or, D.: Set-the-scene: Global-local training for generating controllable nerf scenes (2023)
4. Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al.: Scaling rectified flow transformers for high-resolution image synthesis. In: Forty-first international conference on machine learning (2024)
5. Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., et al.: Scaling rectified flow transformers for high-resolution image synthesis. In: Forty-First International Conference on Machine Learning (ICML) (2024)
6. Feng, W., Zhu, W., jui Fu, T., Jampani, V., Akula, A., He, X., Basu, S., Wang, X.E., Wang, W.Y.: Layoutgpt: Compositional visual planning and generation with large language models. arXiv preprint arXiv:2305.15393 (2023)
7. 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)
8. 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)
9. 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)
10. Gao, J., Zhou, D., Liang, M., Liu, L., Fu, C.W., Hu, X., Heng, P.A.: Disco-layout: Disentangling and coordinating semantic and physical refinement in a multi-agent framework for 3d indoor layout synthesis (2025), `https://arxiv.org/abs/2510.02178`
11. Gao, J., Zhou, D., Liang, M., Liu, L., Fu, C.W., Hu, X., Heng, P.A.: Disco-layout: Disentangling and coordinating semantic and physical refinement in a multi-agent framework for 3d indoor layout synthesis (2025), `https://arxiv.org/abs/2510.02178`
12. Hessel, J., Holtzman, A., Forbes, M., Bras, R.L., Choi, Y.: CLIPScore: a reference-free evaluation metric for image captioning. In: EMNLP (2021)
13. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: Advances in Neural Information Processing Systems (NeurIPS). pp. 6840–6851 (2020)
14. Höllein, L., Cao, A., Owens, A., Johnson, J., Nießner, M.: Text2room: Extracting textured 3d meshes from 2d text-to-image models pp. 7909–7920 (2023)
15. Huang, Z., He, J., Ye, J., Jiang, L., Li, W., Chen, Y., Han, T.: Scene4u: Hierarchical layered 3d scene reconstruction from single panoramic image for your immerse exploration. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 26723–26733 (2025)
16. Jun, H., Nichol, A.: Shap-e:generating conditional 3d implicit functions (2023), `https://arxiv.org/abs/2305.02463`

17. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics **42**(4), 1–14 (2023)
18. Kumaran, V., Rowe, J., Mott, B., Lester, J.: Scenecraft: automating interactive narrative scene generation in digital games with large language models. In: Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment. vol. 19, pp. 86–96 (2023)
19. Li, H., Shi, H., Zhang, W., Wu, W., Liao, Y., Wang, L., Lee, L.h., Zhou, P.Y.: Dreamscene: 3d gaussian-based text-to-3d scene generation via formation pattern sampling. In: European Conference on Computer Vision. pp. 214–230. Springer (2024)
20. Li, X., Lai, Z., Xu, L., Qu, Y., Cao, L., Zhang, S., Dai, B., Ji, R.: Director3d: Real-world camera trajectory and 3d scene generation from text. arXiv:2406.17601 (2024)
21. Liang, Y., Yang, X., Lin, J., Li, H., Xu, X., Chen, Y.: Luciddreamer: Towards high-fidelity text-to-3d generation via interval score matching. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6517–6526 (2024)
22. Lipman, Y., Chen, R.T.Q., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. In: The Eleventh International Conference on Learning Representations (2023), https://openreview.net/forum?id=PqvMRDCJT9t
23. Liu, R., Wu, R., Hoorick, B.V., Tokmakov, P., Zakharov, S., Vondrick, C.: Gala3d: Towards text-to-3d complex scene generation via layout-guided diffusion. arXiv preprint **arXiv:2303.16969** (2023)
24. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021)
25. Murray, N., Marchesotti, L., Perronnin, F.: Ava: A large-scale database for aesthetic visual analysis. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. pp. 2408–2415 (2012). https://doi.org/10.1109/CVPR.2012.6247954
26. Nichol, A., Jun, H., Dhariwal, P., Mishkin, P., Chen, M.: Point-e: A system for generating 3d point clouds from complex prompts. arXiv preprint arXiv:2212.08751 (2022)
27. Nießner, M., Zollhöfer, M., Izadi, S., Stamminger, M.: Real-time 3d reconstruction at scale using voxel hashing. ACM Transactions on Graphics **32**(6), 1–11 (2013)
28. 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)
29. OpenAI, other: Gpt-4 technical report (2024), https://arxiv.org/abs/2303.08774
30. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: Dreamfusion: Text-to-3d using 2d diffusion. arXiv (2022)
31. Pu, G., Zhao, Y., Lian, Z.: Pano2room: Novel view synthesis from a single indoor panorama. In: SIGGRAPH Asia 2024 Conference Papers. pp. 1–11 (2024)
32. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. In: arXiv preprint. vol. arXiv:2010.02502 (2020), https://arxiv.org/abs/2010.02502, published in ICLR 2021
33. 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)

34. Wang, T., Mao, X., Zhu, C., Xu, R., Lyu, R., Li, P., Chen, X., Zhang, W., Chen, K., Xue, T., et al.: Embodiedscan: A holistic multi-modal 3d perception suite towards embodied ai. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19757–19767 (2024)
35. Wang, Z., Lu, C., Wang, Y., Bao, F., Li, C., Su, H., Zhu, J.: Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. arXiv (2023)
36. Xiang, J., Lv, Z., Xu, S., Deng, Y., Wang, R., Zhang, B., Chen, D., Tong, X., Yang, J.: Structured 3d latents for scalable and versatile 3d generation. arXiv preprint arXiv:2412.01506 (2024)
37. Xu, J., Liu, X., Wu, Y., Tong, Y., Li, Q., Ding, M., Tang, J., Dong, Y.: Imagereward: learning and evaluating human preferences for text-to-image generation. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. pp. 15903–15935 (2023)
38. Yang, X., Man, Y., Chen, J.K., Wang, Y.X.: Scenecraft: Layout-guided 3d scene generation. In: Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., Zhang, C. (eds.) Advances in Neural Information Processing Systems. vol. 37, pp. 82060–82084. Curran Associates, Inc. (2024). https://doi.org/10.52202/079017-2608, https://proceedings.neurips.cc/paper_files/paper/2024/file/953d276d037e701fcd97dbb34ebb2394-Paper-Conference.pdf
39. Yang, Y., Jia, B., Zhang, S., Huang, S.: Sceneweaver: All-in-one 3d scene synthesis with an extensible and self-reflective agent. In: Advances in Neural Information Processing Systems (NeurIPS) (2025)
40. 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)
41. 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)
42. Yang, Y., Sun, F.Y., Weihs, L., VanderBilt, E., Herrasti, A., Han, W., Wu, J., et al.: Holodeck: Language guided generation of 3d embodied ai environments. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 16227–16237 (2024)
43. Yu, H.X., Duan, H., Herrmann, C., Freeman, W.T., Wu, J.: Wonderworld: Interactive 3d scene generation from a single image. In: Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR). pp. 5916–5926 (June 2025)
44. Zhang, Q., Wang, C., Siarohin, A., Zhuang, P., Xu, Y., Yang, C., Lin, D., Zhou, B., Tulyakov, S., Lee, H.Y.: Towards text-guided 3d scene composition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6829–6838 (June 2024)
45. Zhou, D., Huang, J., Bai, J., Wang, J., Chen, H., Chen, G., Hu, X., Heng, P.A.: Magictailor: Component-controllable personalization in text-to-image diffusion models. arXiv preprint arXiv:2410.13370 (2024)