# SCENECAST: LEARNING INDOOR SCENE SYNTHESIS FROM CURATED SYNTHETIC VIDEOS WITH LLM REWARD

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

3D indoor scene synthesis requires generating objects to form a complete and coherent scene. Existing approaches rely heavily on manually annotated datasets, leading to high costs and limited generalization. We present SceneCast, an annotation-free framework that trains on large-scale, unlabeled videos, addressing two key challenges: (1) Videos provide only partial views of a scene, requiring the model to extrapolate unseen 3D regions, which is inherently error-prone; (2) Raw videos available online are uncurated, suffering from uncontrolled conditions and variable data quality. First, we design a Transformer-based model that predicts objects sequentially and is fine-tuned with reinforcement learning (RL), where an LLM acts as a spatial common-sense evaluator. The LLM provides reward signals through chain-of-thought reasoning and code-executable geometric checks to refine object configurations. Second, we introduce ViTrim, a data curation approach that generates training videos with controllable video generation models. Crucially, it retains only high-quality prefix clips via multi-criteria quality assessment with an entropy-based weighting strategy. Evaluated on curated synthetic video sets, our annotation-free approach outperforms state-of-the-art supervised methods in scene quality.

## 1 INTRODUCTION

3D indoor scene synthesis is a core form of world model, with broad applications in 3D computer vision, computer graphics, and embodied AI (Driess et al., 2023; Huang et al., 2023; Yang et al., 2024b;a). Given an initial (possibly empty) set of objects placed in a scene, the task requires generating additional objects to produce a complete, coherent scene. The backbone network can be further trained with language conditions to support text-to-scene generation.

Recent approaches, including rule-based parametric modeling (Purkait et al., 2020), deep generative frameworks (Wu et al., 2024; Bokhovkin et al., 2025), and 3D layout prediction pipelines (Fang et al., 2025; Feng et al., 2025), have addressed various aspects of the task. However, these methods critically depend on large-scale, high-quality annotated indoor datasets such as 3D-FRONT (Fu et al., 2021) and ScanNet (Dai et al., 2017), whose acquisition and annotation are costly and difficult to scale. This reliance has become a central bottleneck for further progress.

An alternative approach is to use Internet data, such as web images and videos for training. Recent reconstruction methods (e.g., VGGT (Wang et al., 2025), Depth Anything (Lin et al., 2025a), Lyra (Bahmani et al., 2026), SceneGen (Meng et al., 2026)) and recognition models can extract rich semantic 3D information from such data, *eliminating the need for task-specific human annotation*. Internet videos are particularly appealing: they are abundant and provide broader viewpoint coverage than images. However, training a 3D scene generation model with videos online introduces two major challenges.

First, videos typically provide only partial observations of an indoor environment, requiring the generation model to extrapolate unseen regions in 3D space. Without effective constraints, such extrapolation easily causes error accumulation, geometric inconsistency, and violation of physical plausibility. Second, Internet videos are inherently uncurated along multiple dimensions: camera trajectories are uncontrolled and often degenerate (e.g., near-planar motion, abrupt cuts), scene con-



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

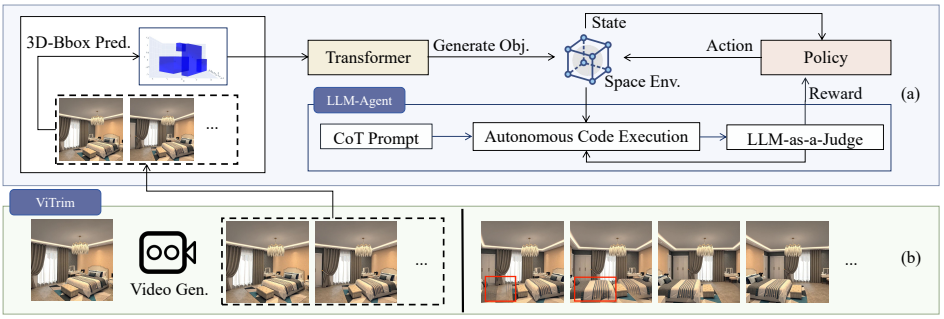



Figure 1: SceneCast overview. (a) Our generation model predicts 3D objects sequentially, starting from an initial set of objects. The model is fine-tuned with a reinforcement learning phase with LLM feedback. (b) Our ViTrim strategy removes low-quality tail segments from generated video using multiple quality assessment criteria, ensuring only valid clips are used for training.

tent is non-static, and photometric conditions vary widely. These factors collectively degrade data quality, and addressing them demands broad data collection and heavy post-processing, with little guarantee of data quality.

To address the first challenge, we propose a Transformer model trained to predict 3D objects sequentially, then fine-tuned with reinforcement learning (RL). Each object is represented as a labeled 3D bounding box and instantiated with a 3D asset at render time. As shown in Figure 1(a), given an initial set of objects, the model completes the scene by predicting the remaining objects one by one. To align the scene with spatial commonsense, RL iteratively refines the layout by adjusting object positions, sizes, and orientations, guided by reward signals from an LLM acting as a judge. The RL agent treats the current layout as its state and per-object attribute adjustments as actions. The LLM scores the layout against multiple criteria and returns a scalar reward to drive policy optimization. To improve scoring reliability, we prompt the LLM with Chain-of-Thought (CoT) reasoning backed by automatic code execution, enabling accurate geometric reasoning.

To address the quality control issue of raw Internet video, we propose a data curation approach called *3D-Quality Aware Video Trimming* (ViTrim). Rather than sourcing from the web, our approach generates training videos using recent video generation models. Pretrained on large-scale video data, these models encode strong priors over 3D object statistics and produce high-quality frames for short durations (e.g., a few seconds). Crucially, this approach offers control over scene content and camera trajectories, enabling targeted coverage of the indoor cases relevant to our task. A well-known problem with generated videos, however, is the error accumulation over long sequences. To address this, we propose a video trimming strategy, as shown in Figure 1 (b). The method evaluates segments of each generated video, and trims the video to retain the initial high-quality clip. It evaluates multiple metrics, including human perceptual quality, spatial consistency, and physical plausibility, using a set of reward models (Xu et al., 2023; 2024; Bansal et al., 2024) whose scores are aggregated via *entropy-based weighting*. By doing so, our approach ensures that only usable data enters the 3D reconstruction pipeline.

Our main contributions are summarized as follows: (1) An annotation-free 3D scene completion model that predicts 3D objects sequentially, and is fine-tuned with reinforcement learning using LLM feedback. (2) A data curation method that uses a video generation model to generate synthetic indoor scenes, followed by a quality-aware trimming strategy to retain only high-quality clips for training. (3) A comprehensive performance evaluation on the 3D-FRONT benchmark that shows our model outperforms state-of-the-art methods.

## 2 Related Work

**Indoor Scene Generation.** Early methods for indoor scene generation use parametric methods (Yu et al., 2015; Fu et al., 2017; Choi et al., 2015) to explicitly model the spatial layout of indoor scenes. In contrast, data-driven deep learning approaches attempt to learn scene distributions from large-scale labeled data (Wang et al., 2021; Li et al., 2018; Zhong et al., 2025). Recent research advances

include autoregressive generation methods (Ye et al., 2024; Paschalidou et al., 2021; Chattopadhyay et al., 2023; Choi et al., 2026; Sun et al., 2024), latent space generation methods (Purkait et al., 2020; Lin et al., 2025b), and diffusion generation methods (Zhai et al., 2024; Ju et al., 2024; Huang et al., 2025b; Tang et al., 2024; Meng et al., 2025; Bokhovkin et al., 2025; Chen et al., 2024). To reduce reliance on data, VIPScene (Huang et al., 2025a) leverages video models to synthesize scenes with consistent views and broad viewpoint coverage, and concurrent works distill video diffusion models into explicit 3D representations (Bahmani et al., 2026; Zhang et al., 2026) or generate 3D scenes directly from a single image (Meng et al., 2026; Koo et al., 2026). However, existing video and image driven methods fail to integrate indoor spatial common sense and global layout constraints into the generation process.

**LLM-Assisted Scene Generation.** Prior works have explored using LLMs in indoor scene generation either as a semantic interface (Yang et al., 2024b; Jain et al., 2023; Liu et al., 2023), or as a prior encoder that provides design rules and spatial constraints (Aguina-Kang et al., 2024; Zhang et al., 2024; Yang et al., 2025b; Jiang et al., 2026). Recently, LLMs have also been employed as reward evaluators, producing scalar reward signals or preference rankings to guide policy optimization (Sun et al., 2025; Yuan & Cheng, 2026; Ran et al., 2025). Concurrent to our work, several agent-based systems (Yang et al., 2026; Pfaff et al., 2026; Xia et al., 2026; He et al., 2026) employ LLM/VLM agents as explicit planners that iteratively construct scenes through tool calls. However, these designs rely on the LLM's native inference, which lacks geometric reasoning accuracy. In contrast, our work uses the LLM as an agent that generates and invokes code for precise geometric computation, and optimizes a dedicated layout backbone rather than the language model itself. Our backbone is trained on video-derived layout statistics, providing a statistically grounded layout that agentic planners can refine with constraints.

**Generated Video Quality Evaluation.** Existing video quality assessment methods operate along complementary dimensions: human-preference scores learned from annotated comparisons (He et al., 2024; Jiang et al., 2023; Shan et al., 2025; Xu et al., 2024) capture perceptual fidelity, while VideoPhy (Bansal et al., 2024) evaluates compliance with physical commonsense. Benchmark suites such as VBench (Huang et al., 2024), VBench2.0 (Zheng et al., 2025), and Video-Bench (Han et al., 2025) aggregate multiple such metrics for standardized model comparison. However, none of these methods assess 3D spatial validity, which is the failure mode most critical to downstream 3D scene reconstruction. Moreover, multi-metric suites report scores independently without task-adaptive fusion. Our ViTrim builds on VisionReward and VideoPhy as two of its evaluation dimensions, but introduces a spatial IoU signal targeting 3D structural quality, and fuses all dimensions via entropy-based weighting that adapts to each video's quality profile.

# 3 METHOD

In SceneCast, a 3D indoor scene is defined as an unordered set of 3D furniture objects $\mathcal{F} = \{f_v\}_{v=1}^{N}$. Each object $f_v$ has four attributes: $f_v = (c_v, p_v, u_v, d_v)$, where $p_v \in \mathbb{R}^3$ is the center position, $u_v \in \mathbb{R}$ is the rotation angle around the vertical (up) axis, $d_v \in \mathbb{R}^3$ is the object size, and $c_v \in \{1, 2, \ldots, C\}$ denotes the object category. The category vocabulary $\mathcal{C} = \{1, 2, \ldots, C\}$ is defined by a predefined label set used during object recognition (via Grounded-SAM (Ren et al., 2024)) in the data preparation pipeline, and is therefore extensible: increasing the number of predefined labels directly increases $C$ without architectural changes. In 3D-FRONT experiments, we set $C = 34$ to match the category of the baseline methods for fair comparison. Given an initial (possibly empty) 3D objects set, we use a transformer to generate the remaining objects step by step. Commonsense knowledge is further integrated into the learning process through reinforcement learning that interfaces with an LLM.

## 3.1 PRETRAINING OF GENERATION BACKBONE

For training data preparation, we first collect a set of indoor images from the Internet and use them as prompts for a video generation model to produce short video segments. We then apply our ViTrim method to retain only the valid prefix segments from these videos. Following this, we extract 3D objects using a reconstruction and recognition pipeline (e.g., VIPScene (Huang et al., 2025a)). The extracted objects are finally recorded to form our training dataset.

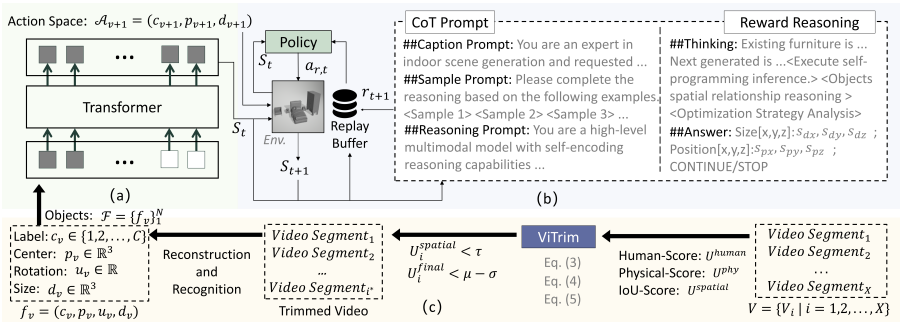



Figure 2: SceneCast pipelines. (a) Transformer-based autoregressive 3D indoor scene completion backbone network. (b) LLM-driven reinforcement learning fine-tuning loop incorporating CoT spatial reasoning, automatic geometric code execution, and multi-dimensional reward fusion. (c) ViTrim pipeline for high-quality video segment selection.

Figure 2 illustrates the workflow of our approach, which adopts a Transformer backbone. Different from previous work like ATISS (Paschalidou et al., 2021), our training data only contain incomplete scene layouts. Therefore, we make one modification: we remove the learnable end-of-sequence (EOS) token and fix the total generation steps to a predefined constant $N_{\text{max}}$, as shown in Figure 2(a). After pre-training, we fine-tune it via reinforcement learning (RL), where the reward signal is derived from an LLM-based reward model.

### 3.2 Fine-Tuning via Reinforcement Learning

**Preliminary.** We model the indoor 3D scene fine-tuning task as a Markov Decision Process (MDP) $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma)$, where $s_t \in \mathcal{S}$ is the state space represented by current generated 3D scene parameters, including furniture type, size, and pose; $\mathcal{A}$ is the action space comprising adjustments of object attributes; $\mathcal{T}$ is the state transition function that maps from current state $s_t$ to next state $s_{t+1}$ upon action $a_t$; $\mathcal{R}$ is the reward function generated by our LLM Agent; $\gamma \in [0, 1]$ is the discount factor. At each time step, the agent receives the current state $s_t$ and generates an action $a_t \in \mathcal{A}$ according to the policy $\pi$. Upon executing the action, the environment returns a scalar reward $r_{t+1} \sim \mathcal{R}(S_t, a_t)$, the next state $S_{t+1}$, and the done flag.

We adopt the Soft Actor-Critic (SAC) algorithm (Haarnoja et al., 2018) to alternately optimize the actor network $\pi_\theta$ and critic network $Q_\phi$; the actor and critic loss formulations are provided in Appendix C.1. Notably, object categories are not part of the action space: they are fixed by the backbone's autoregressive predictions, and the SAC policy refines only the continuous bounding-box attributes $(p_v, u_v, d_v)$ of the current layout. Generation termination is decided by the LLM's control signal $R_t^c$ rather than by the policy.

**LLM Reward Design.** The LLM-Agent-based reward mechanism involves three components:

(1) CoT-guided Spatial and Categorical Reasoning. Figure 2(b) shows the CoT prompt reasoning pipeline. Given a scene layout, we use role prompting (John, 2023) to guide the LLM to evaluate the category suitability, 3D IoU, and scale matching of the next object $f_{v+1}$ to be generated, assuming existing objects $\mathcal{F} = \{f_v\}_{v=1}^N$. We integrate a prompt-based self-coding mechanism to automatically generate and execute code during reasoning to obtain precise measurements. We further provide manually crafted layout examples and adopt the Chain-of-Thought (CoT) paradigm, which decomposes the complex indoor scene evaluation into explicit, manageable steps covering category verification, size rationality and relative position analysis.

(2) Automatic Code Execution. Following the calculation specifications derived from CoT reasoning, the LLM generates Python code for 3D geometric evaluation. The code executes in an isolated sandbox, taking the full scene state including object categories, 3D coordinates, sizes and rotations as input. It computes quantitative metrics such as 3D bounding box IoU, volume overlap, inter-object distance, floor adherence and category-aware scale matching. All metrics are normalized to $[0, 1]$ and directly mapped to position quality score $r_{\text{p}}$ and scale quality score $r_{\text{d}}$. If code execution fails, LLM iteratively revises the program according to error feedback to maintain stable reward.

(3) Reward Fusion and Early Stopping. We first construct a multi-dimensional reward set $R_t = \{R_t^{\text{p}}, R_t^{\text{d}}, R_t^{\text{c}}\}$ tailored for the CoT reasoning and automatic code execution pipeline. All reward components are normalized to $[0, 1]$, with a higher score indicating better scene quality. $R_t^{\text{p}} = \{s_{\text{px}}, s_{\text{py}}, s_{\text{pz}}\}$ and $R_t^{\text{d}} = \{s_{\text{dx}}, s_{\text{dy}}, s_{\text{dz}}\}$ denote the position quality and the scale quality scores along the $x$, $y$, and $z$ axes, assessing the distribution of object center positions in terms of inter-object spacing, consistency between each object's actual size and category-specific statistical priors, as well as functional area compatibility. Finally, $R_t^{\text{c}} = \{0, 1\}$ is a discrete control signal derived from the LLM CoT inference output, where "0" triggers early generation termination.

As shown in Figure 2(b), the optimization process follows a closed loop: (1) The actor outputs an initial scene state (with object categories, sizes, poses); (2) LLM performs CoT reasoning and code execution to generate reward $R_t$; (3) SAC uses $R_t$ to update actor/critic networks, adjusting object parameters to obtain a new state; (4) If $R_t$ reaches the predefined threshold, LLM triggers early stopping; otherwise, repeat the loop. When the score gap exceeds a threshold, the LLM revises its reasoning and code to improve evaluation accuracy.

### 3.3 3D-Quality Aware Video Trimming (ViTrim)

**Evaluation Framework.** We evaluate video quality using a sliding window. Given a video of total duration $T$, we set window length to 1 s and sliding step to 0.1 s, yielding sampled video segments set $V = \{V_i \mid i = 1, 2, \ldots, X\}$. Each segment is evaluated from three complementary dimensions, with all scores normalized to $[0, 1]$: (1) VisionReward score (Xu et al., 2024) $U^{\text{human}}$, which quantifies visual naturalness and human preference of indoor scenes; (2) VideoPhy score (Bansal et al., 2024) $U^{\text{phy}}$, which measures whether object arrangements comply with physical constraints; (3) Average object IoU $U^{\text{spatial}}$, which is proposed in this work to reflect spatial validity by penalizing redundant overlap between object pairs. The final evaluation score is an entropy-based weighted (Shannon, 1948; Kiker et al., 2005) sum of these scores, penalized by cumulative spatial error. The evaluation procedure is illustrated in Figure 2(c).

**Entropy-based Score Fusion.** Given a metric space $U = \{U^j \mid j = 1, 2, \ldots, Y\}$, where $U^j$ represents the $j$-th metric. For the $j$-th metric, its information entropy $E^j$ is:

$$E^j = -\frac{1}{\ln X} \sum_{i=1}^{X} p_i^j \ln\left(p_i^j + \epsilon\right), \tag{1}$$

where $p_i^j = U_i^j / \sum_{i=1}^{X} {U_i}^j$ is the normalized proportion of the $i$-th video segment, and $\epsilon = 10^{-8}$ avoids logarithmic singularity. When the denominator $\sum_{i=1}^{X} U_i^j = 0$, we uniformly set $p_i^j = 1/X$ for all video segments to ensure numerical stability. A lower value of $E^j$ implies a stronger discriminative degree $D^j = 1 - E^j$. We assign weights to the metric $w^j = D^j / \sum_{j=1}^{Y} D^j$, where $\sum_{j=1}^{Y} w^j = 1$. With the calculated objective weights, we define the fusion score for each video segment as:

$$U_i^{\text{fused}} = w^{\text{human}} \cdot U_i^{\text{human}} + w^{\text{phy}} \cdot U_i^{\text{phy}} + w^{\text{spatial}} \cdot U_i^{\text{spatial}}. \tag{2}$$

**Cumulative Spatial Error as Penalty.** Per-segment IoU only measures short-term video quality and does not reflect the progressive deterioration of scene quality commonly found in generated videos. To capture such an effect, we introduce cumulative spatial error. Given a set of spatial IoU scores $U^{\text{spatial}}$, we define the cumulative spatial error, for the video segments from 1 to $i$, as $E_i^{\text{cum}} = \sum_{k=2}^{i} (U_1^{\text{spatial}} - U_k^{\text{spatial}})$. We take the first video segment as a baseline, which is often of high quality but can exhibit certain spatial overlap due to the bounding box approximations, and add up the spatial error of each segment relative to the first one. We normalize this error term to $[0, 1]$, and incorporate it as a penalty into the fused score, obtaining the final validity score:

$$U_i^{\text{final}} = \text{segment}\left(U_i^{\text{fused}} - \gamma \cdot E_i^{cum},\ 0,\ 1\right), \tag{3}$$

where $\gamma > 0$ controls penalty strength ($\gamma = 0.5$ in our experiments), and segment$(\cdot)$ constrains scores within $[0, 1]$.

**Hard-Soft Constraints Guided Video Trimming.** ViTrim applies both hard and soft constraints to select valid video segments. A segment satisfying $U_i^{\text{spatial}} < \tau$ is immediately rejected, where $\tau$

| Method | Bedroom | | | | | LivingDiningRoom | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | FID ↓ | KID ↓ | SCA % | CKL ↓ | PIoU ↓ | FID ↓ | KID ↓ | SCA % | CKL ↓ | PIoU ↓ |
| ATISS | 123.06 | 0.70 | 57.20 | 1.85 | 0.48 | 135.27 | 7.98 | 67.72 | 2.78 | 1.52 |
| DiffuScene | 124.19 | 0.62 | 60.57 | 1.59 | 0.46 | 141.45 | 7.75 | 70.50 | 1.56 | 0.69 |
| VIPScene | 137.81 | 0.67 | 68.01 | 2.19 | 0.36 | 147.36 | 9.05 | 74.32 | 2.59 | 0.34 |
| Ours | **111.06** | **0.56** | **55.65** | **1.13** | **0.25** | **128.57** | **5.64** | **62.58** | **1.38** | **0.27** |

Table 1: Quantitative results. ATISS (Paschalidou et al., 2021), DiffuScene (Tang et al., 2024), and VIPScene (Huang et al., 2025a) are compared. SceneCast performs best across all metrics. PIoU (↓) is newly introduced to measure physical plausibility; baseline PIoU values are computed on their released scenes.

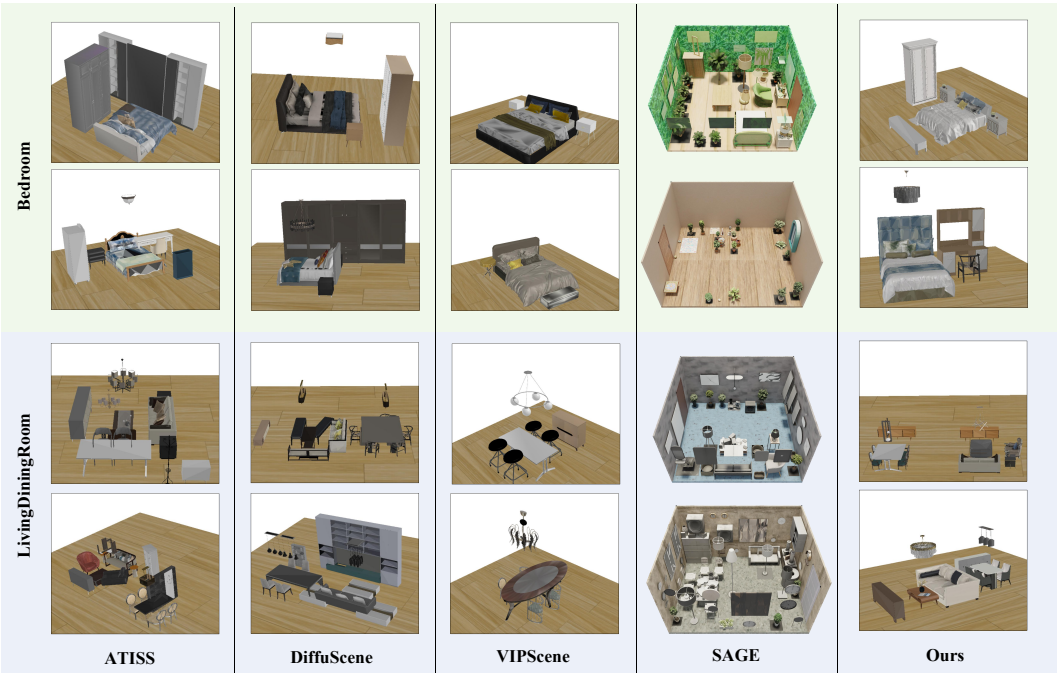



Figure 3: Generation samples from an empty scene. SceneCast performs well in producing reasonable and complete object layouts, with correct collision avoidance. ATISS and DiffuScene generate misaligned objects and unnatural layouts. Perhaps restricted by the video generation model, VIPScene generates a small number of objects with narrow coverage of the scene, missing objects that are commonly found in realistic scenes (e.g., wardrobe for bedroom, sofa for livingdiningroom). SAGE (Xia et al., 2026) lacks the understanding of the actual physical space, resulting in the generation of duplicate or poorly arranged objects (e.g., bedroom without bed, impassable passageway).

is a hard spatial threshold ($\tau = 0.5$ in our experiments), as severe spatial overlap clearly violates physical plausibility. Also, we compute the global mean $\mu$ and standard deviation $\sigma$ of all $U_i^{\text{final}}$ within a video. We set the soft quality threshold as $T_\sigma = \mu + \beta \cdot \sigma$, where $\beta$ is the weight of $\sigma$, and define the *trimming point* $i^\star$ as the first segment that violates either constraint: $i^\star = \min\{i \mid U_i^{\text{spatial}} < \tau \text{ or } U_i^{\text{final}} < T_\sigma\}$, where $U_i^{\text{spatial}} = 1 - IoU_i$. All segments preceding $i^\star$ are retained as the valid clip: $\mathcal{V}_{\text{valid}} = \{V_1, V_2, \ldots, V_{i^\star - 1}\}$, which is passed to the 3D reconstruction pipeline.

## 4 Experiments

**Datasets and Experimental Details.** We collected 20,000 bedroom images and 20,000 living room images from Google and the image collection of 3D-FRONT (see Appendix A.2 for details). Each image serves as the initial frame for LTX-Video (HaCohen et al., 2024), which synthesizes a 5-

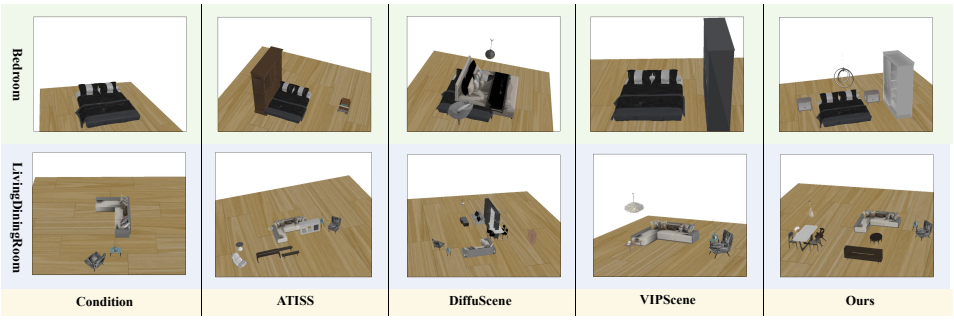



Figure 4: Scene completion results from partial scenes. Starting from a few objects, SceneCast completes the scenes more thoroughly and plausibly.

second video that preserves the original scene layout with temporally consistent frames. Following VIPScene (Huang et al., 2025a), we apply Grounded-SAM (Ren et al., 2024) for object segmentation, UniDepth (Piccinelli et al., 2024) for depth estimation, MASt3R (Leroy et al., 2024) for cross-frame correspondence, and Fast3R (Yang et al., 2025a) for 3D scene reconstruction. We evaluate all methods on living room, dining room, and bedroom scenes. Our model is trained on no 3D layout annotation data and rendered using 3D assets from 3D-FRONT (Fu et al., 2021) (see Appendix B.5 for a discussion of generalization). Specifically, we populate each predicted box $d_v$ by selecting the 3D-FUTURE asset whose normalized dimension vector $(w/h, d/h)$ minimizes the $\ell_1$ distance to that of $d_v$, so that the rendered object fits the predicted envelope without rescaling or deformation. Further experimental details and hyperparameters are provided in Appendix A.1.

**Metrics.** Following prior work (Paschalidou et al., 2021; Tang et al., 2024), we report FID, KID ($\times 0.001$), SCA, CKL ($\times 0.01$), and PIoU ($\times 0.01$). Lower FID, KID, CKL, and PIoU indicate better fidelity, diversity, category alignment, and spatial plausibility, respectively; an SCA near 50% indicates generated scenes are indistinguishable from real ones. Full metric definitions are provided in Appendix A.4.

**Baselines.** We compare against ATISS (Paschalidou et al., 2021), DiffuScene (Tang et al., 2024), and VIPScene (Huang et al., 2025a). ATISS and DiffuScene are both trained on annotated 3D-FRONT (Fu et al., 2021) data: ATISS generates scenes autoregressively over a discretized object space, while DiffuScene uses a diffusion model. VIPScene reconstructs 3D scenes from video by drawing on physical commonsense encoded in a video generation model, and requires no annotated data. Since VIPScene provides no per-scene ground truth, we follow standard protocols and evaluate all methods on global distribution metrics, ensuring a fair comparison. We also compare the generation quality against the recent agent-based method SAGE (Xia et al., 2026).

### 4.1 Experimental Results

Table 1 reports quantitative results for all methods. Although SAGE tackles a similar task, it does not learn from real scene data, so it cannot be fairly compared on 3D-FRONT metrics and scene completion. For SAGE, we report qualitative results only. Our approach outperforms all supervised and video-based baselines across metrics. Specifically, compared to DiffuScene, SceneCast improves FID by 10% and KID by 18%, demonstrating stronger alignment with the real scene distribution and greater diversity. Our SCA score is closest to 50%, confirming that the generated scenes are nearly indistinguishable from real ones. Critically, SceneCast achieves the lowest PIoU, indicating substantially fewer inter-object collisions and thus more physically plausible layouts—a dimension on which the distributional metrics (FID/KID) are silent. ATISS and DiffuScene require costly labeled training data, whereas SceneCast is annotation-free. VIPScene also requires no labels, yet underperforms across the metrics, because it reconstructs only the 3D scene visible in the partial generated video and never observes the full scene distribution.

Figure 3 visualizes scene generation results from an empty scene. The agent-based SAGE could produce more complete scenes; however, due to a lack of real-world scene distributions, the generated objects often exhibit redundancy and unreasonable spatial arrangements. Although ATISS and DiffuScene produce generally reasonable layouts, they frequently exhibit object misalignment and

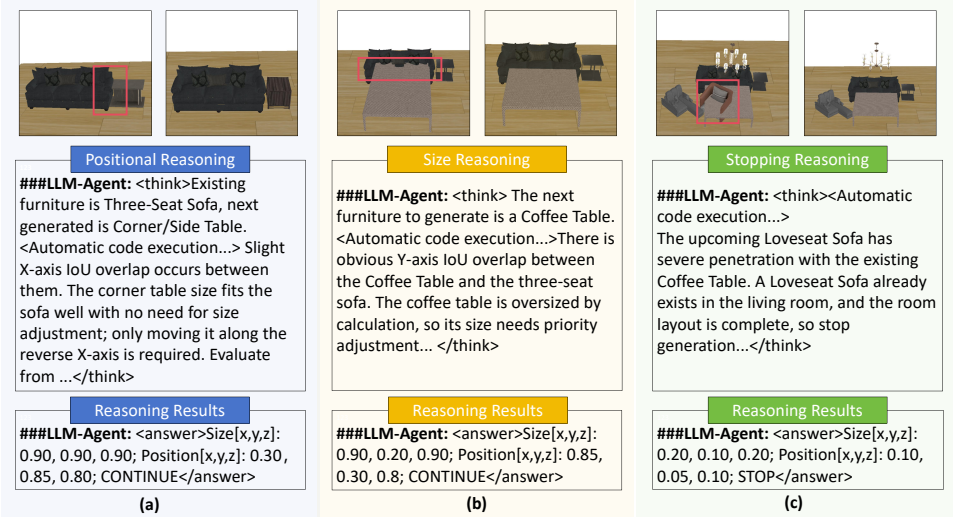



Figure 5: LLM reasoning process. (a) Position refinement for minor spatial overlap; (b) Size correction for severe object interpenetration; (c) Layout saturation detection and generation stopping. The LLM Agent applies CoT reasoning and self-programmed geometric computation. In each case, Size[x,y,z] denotes the corrected 3D dimension of an object, Position[x,y,z] its refined 3D centroid. CONTINUE indicates proceeding with object generation, while STOP terminates the process when layout is complete or spatially invalid.

interpenetration, yielding physically implausible arrangements. Both methods rely solely on statistical patterns in the training data and receive no spatial commonsense guidance. When the learned distribution fails to cover certain cases, the models produce layouts that violate physical plausibility. Figure 4 shows the results of scene completion starting from a few objects, a set of experiments revealing more details of how generated objects interact with existing ones. Due to the lack of explicit constraints, ATISS and DiffuScene can produce object categories that conflict with existing parts of the scene during scene completion (e.g., producing an extra sofa when a sofa already exists, or producing an extra bed when one already exists.). While VIPScene can usually reconstruct a reasonable layout from a video, the number and types of objects it generates are limited, resulting in an incomplete layout. In contrast, SceneCast generates complete, physically plausible layouts by drawing on the spatial commonsense encoded in the LLM reward.

### 4.2 Ablation Study

**LLM Reward Mechanism.** Figure 5 presents explicit CoT reasoning traces and numerical outputs from the LLM reward mechanism. In Figure 5(a), the LLM reasons over the spatial relationship between an existing three-seat sofa and a newly generated corner table. Through automatic code execution, it computes 3D IoU and detects a slight overlap along the X-axis. Since the object sizes are already within acceptable bounds, the LLM refines the position parameters to resolve the overlap and returns a CONTINUE signal to proceed with generation. In Figure 5(b), faced with severe interpenetration, the LLM prioritizes size calibration to eliminate the conflict. In Figure 5(c), foreseeing simultaneous penetration and redundant object placement, the LLM determines that the scene is complete and issues an early stopping signal. These examples show that the LLM reward captures spatial constraints (object conflicts, scale errors, redundant generation) that fixed rules cannot easily encode. We verify this quantitatively below.

**Reward Design Ablation.** We isolate the contribution of each component in the reward design by progressively augmenting the reward signal, keeping the SAC pipeline identical across all variants. Starting from the pre-RL backbone (**No RL**), we first add a purely algorithmic geometric reward (**Rule-Geo**) that penalizes collisions, enforces category-specific size ranges, and checks floor contact. We then augment it with hand-crafted categorical rules (**Rule+Cat**), such as "at most one bed per bedroom" and "nightstands adjacent to the bed." Next, we replace the rules with the LLM reward but disable code execution (**LLM w/o code**), so that the LLM reasons about geometry in natural lan-

| Reward config. | FID ↓ | PIoU ↓ |
|---|---|---|
| No RL | 156.6 | 1.34 |
| Rule-Geo | 147.3 | 0.78 |
| Rule+Cat | 143.5 | 0.72 |
| LLM w/o code | 141.3 | 0.52 |
| LLM w/ code | **111.1** | **0.25** |

(a) Reward ablation (bedroom)

| Curation strategy | FID ↓ | KID ↓ |
|---|---|---|
| No filtering | 133.9 | 0.69 |
| Fixed 1.5 s prefix | 136.2 | 0.70 |
| VR+VP only | 132.2 | 0.61 |
| ViTrim (equal $w$) | 127.1 | 0.64 |
| ViTrim (entropy) | **111.1** | **0.56** |

(b) ViTrim ablation (bedroom)

| Mask ratio | HCR ↑ | Dup ↓ |
|---|---|---|
| 25% | 0.67 | 0.18 |
| 50% | 0.58 | 0.17 |
| 75% | 0.34 | 0.11 |

(c) Conditional completion

Table 2: Ablations on the bedroom set. (a) Unified reward ablation. Each row augments the previous one with a single component, isolating its effect. The LLM reward outperforms hand-crafted rules, and code execution provides a further gain over language-only reasoning. (b) ViTrim ablation. All models share identical architecture and training hyperparameters, while only the data curation strategy differs. Rows 1–2 isolate the effect of trimming, row 3 isolates the spatial IoU signal and rows 4–5 compare entropy-based vs uniform weighting.(c) SceneCast completion quality across condition coverage on held-out 3D-FRONT scenes.

guage only. Finally, we enable automatic code execution (**LLM w/ code**), our full configuration. Table 2 (a) reports the results, where each row differs from the one above by exactly one component.

First, geometric reinforcement learning already improves over the backbone, but hand-crafted rules (Rule+Cat) yield only marginal gains, as a fixed rule set cannot enumerate the diversity of functional arrangements. Second, replacing rules with LLM reasoning (LLM w/o code) brings a substantial improvement, showing that open-ended spatial reasoning matters more than enumerated rules. Third, enabling code execution further reduces PIoU by 0.27, showing that precise geometric computation, not language reasoning alone, is essential for reliable reward. Together, these results justify both the use of an LLM and the code-execution mechanism.

**Effect of ViTrim.** ViTrim fuses two quality metrics (VisionReward, VideoPhy) with a new spatial IoU signal via entropy-based weighting. To isolate each factor, we train five otherwise identical models on the bedroom set, varying only the data curation strategy. Table 2 (b) reports the results. Comparing row 1 with row 5, discarding corrupt video suffixes reduces FID by 22.8, confirming that untrimmed videos introduce noisy training samples. The fixed 1.5 s prefix (row 2) performs no better than no filtering (133.9), suggesting that a fixed cut-off discards useful content in some videos while retaining corruption in others. Comparing row 3 with row 5, adding the spatial IoU signal further reduces FID by 21.1, showing that perceptual and physical quality scores alone are insufficient to detect geometric interpenetration. Comparing row 4 with row 5, entropy-based weighting adapts the cut-off to each video's quality profile and outperforms a uniform $\frac{1}{3}$ allocation. Qualitative visualization is provided in Appendix B.2.

**Conditional completion analysis.** Following the partial-scene completion protocol of DiffuScene (Tang et al., 2024), we mask 25%, 50%, and 75% of the objects in scenes and evaluate the result with category-based metrics. We adapt the category recall convention: hidden-category recall (HCR) is the fraction of masked categories regenerated, and duplicate rate (Dup) is the fraction of generated objects duplicating an input category beyond its input count. Instance-level matching is ill-posed for generative models, hence both metrics are category-based. Table 2 (c) evaluates SceneCast's conditional completion across condition coverage. HCR decreases monotonically as more objects are masked, confirming that completion is condition-driven; Dup remains low across all ratios, indicating that duplicating an existing category is rare for SceneCast.

## 5 Conclusion

Indoor scene generation has long been constrained by the cost of annotated training data. This work trains a scene completion model on unlabeled video data through two core designs: a Transformer backbone fine-tuned with LLM-guided reinforcement learning to suppress extrapolation errors and enforce spatial commonsense, and a 3D quality-aware video trimming strategy that selects synthetic training clips through multi-criteria scoring and filtering. On the 3D-FRONT benchmark, our method outperforms fully supervised baselines. Extending the model with language conditioning to enable text-to-scene generation is a natural next step, which we leave for future work.

### AI USE STATEMENT

Video generation models were used to construct the dataset, and generative AI tools were used during the preparation of this work to assist with language editing and improving the clarity and organization of the manuscript, as well as for literature search and summarization. All research ideas, methodological decisions, experimental results, and conclusions were independently examined and validated by the authors. Relevant literature suggested by AI tools was manually verified against the original sources, and all AI-assisted text and technical content were reviewed and revised by the authors. The authors take full responsibility for the final content of this work, including all text, claims, results, and artifacts produced with the aid of generative AI.

## REFERENCES

Rio Aguina-Kang, Maxim Gumin, Do Heon Han, Stewart Morris, Seung Jean Yoo, Aditya Ganeshan, R Kenny Jones, Qiuhong Anna Wei, Kailiang Fu, and Daniel Ritchie. Open-universe indoor scene generation using llm program synthesis and uncurated object databases. *arXiv preprint arXiv:2403.09675*, 2024.

Sherwin Bahmani, Tianchang Shen, Jiawei Ren, Jiahui Huang, Yifeng Jiang, Haithem Turki, Andrea Tagliasacchi, David Lindell, Zan Gojcic, Sanja Fidler, et al. Lyra: Generative 3d scene reconstruction via video diffusion model self-distillation. In *International Conference on Learning Representations*, volume 2026, pp. 82850–82880, 2026.

Hritik Bansal, Zongyu Lin, Tianyi Xie, Zeshun Zong, Michal Yarom, Yonatan Bitton, Chenfanfu Jiang, Yizhou Sun, Kai-Wei Chang, and Aditya Grover. Videophy: Evaluating physical commonsense for video generation. *arXiv preprint arXiv:2406.03520*, 2024.

Aleksey Bokhovkin, Quan Meng, Shubham Tulsiani, and Angela Dai. Scenefactor: Factored latent 3d diffusion for controllable 3d scene generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 628–639, 2025.

Aditya Chattopadhyay, Xi Zhang, David Paul Wipf, Himanshu Arora, and René Vidal. Learning graph variational autoencoders with constraints and structured priors for conditional indoor 3d scene generation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 785–794, 2023.

Jiacheng Chen, Yuhang Zhang, Xiang Li, Sifei Liu, and Xiaolong Wang. Instructscene: Instruction-driven 3d indoor scene generation with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20762–20772, 2024.

Jeongjun Choi, Yeonsoo Park, and H Jin Kim. Scenenat: Masked generative modeling for language-guided indoor scene synthesis. *arXiv preprint arXiv:2601.07218*, 2026.

Wongun Choi, Yu-Wei Chao, Caroline Pantofaru, and Silvio Savarese. Indoor scene understanding with geometric and semantic contexts. *International Journal of Computer Vision*, 112(2):204–220, 2015.

Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5828–5839, 2017.

Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

Chuan Fang, Yuan Dong, Kunming Luo, Xiaotao Hu, Rakesh Shrestha, and Ping Tan. Ctrl-room: Controllable text-to-3d room meshes generation with layout constraints. In *2025 International Conference on 3D Vision (3DV)*, pp. 692–701. IEEE, 2025.

Weitao Feng, Hang Zhou, Jing Liao, Li Cheng, and Wenbo Zhou. Casagpt: cuboid arrangement and scene assembly for interior design. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 29173–29182, 2025.

Huan Fu, Bowen Cai, Lin Gao, Ling-Xiao Zhang, Jiaming Wang, Cao Li, Qixun Zeng, Chengyue Sun, Rongfei Jia, Binqiang Zhao, 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.

Qiang Fu, Xiaowu Chen, Xiaotian Wang, Sijia Wen, Bin Zhou, and Hongbo Fu. Adaptive synthesis of indoor scenes via activity-associated object relation graphs. *ACM Transactions on Graphics*, 36(6):1–13, 2017.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. Pmlr, 2018.

Yoav HaCohen, Nisan Chiprut, Benny Brazowski, Daniel Shalem, Dudu Moshe, Eitan Richardson, Eran Levin, Guy Shiran, Nir Zabari, Ori Gordon, et al. Ltx-video: Realtime video latent diffusion. *arXiv preprint arXiv:2501.00103*, 2024.

Hui Han, Siyuan Li, Jiaqi Chen, Yiwen Yuan, Yuling Wu, Yufan Deng, Chak Tou Leong, Hanwen Du, Junchen Fu, Youhua Li, et al. Video-bench: Human-aligned video generation benchmark. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 18858–18868, 2025.

Xuan He, Dongfu Jiang, Ge Zhang, Max Ku, Achint Soni, Sherman Siu, Haonan Chen, Abhranil Chandra, Ziyan Jiang, Aaran Arulraj, et al. Videoscore: Building automatic metrics to simulate fine-grained human feedback for video generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 2105–2123, 2024.

Yun He, Kelin Yu, and Matthias Zwicker. Sceneorchestra: Efficient agentic 3d scene synthesis via full tool-call trajectory generation. *arXiv preprint arXiv:2604.19907*, 2026.

Jiangyong Huang, Silong Yong, Xiaojian Ma, Xiongkun Linghu, Puhao Li, Yan Wang, Qing Li, Song-Chun Zhu, Baoxiong Jia, and Siyuan Huang. An embodied generalist agent in 3d world. *arXiv preprint arXiv:2311.12871*, 2023.

Rui Huang, Guangyao Zhai, Zuria Bauer, Marc Pollefeys, Federico Tombari, Leonidas Guibas, Gao Huang, and Francis Engelmann. Video perception models for 3d scene synthesis. *arXiv preprint arXiv:2506.20601*, 2025a.

Zehuan Huang, Yuan-Chen Guo, Xingqiao An, Yunhan Yang, Yangguang Li, Zi-Xin Zou, Ding Liang, Xihui Liu, Yan-Pei Cao, and Lu Sheng. 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, 2025b.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21807–21818, 2024.

Ayush Jain, Siddharth Goyal, Benjamin Liu, and Jitendra Malik. Scenegpt: Towards generalizable indoor scene generation with large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023.

Haiyan Jiang, Deyu Zhang, Dongdong Weng, Weitao Song, and Henry Been-Lirn Duh. Hog-layout: Hierarchical 3d scene generation, optimization and editing via vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 31586–31596, June 2026.

Yuming Jiang, Shuai Yang, Tong Liang Koh, Wayne Wu, Chen Change Loy, and Ziwei Liu. Text2performer: Text-driven human video generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22747–22757, 2023.

Ibrahim John. The art of asking chatgpt for high-quality answers. *Nzunda Technologies Ltd*, 2023.

Xiaoliang Ju, Zhaoyang Huang, Yijin Li, Guofeng Zhang, Yu Qiao, and Hongsheng Li. 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.

Gregory A Kiker, Todd S Bridges, Arun Varghese, Thomas P Seager, and Igor Linkov. Application of multicriteria decision analysis in environmental decision making. *Integrated environmental assessment and management*, 1(2):95–108, 2005.

Gwanhyeong Koo, Hyunsu Kim, Youngji Kim, Taejae Lee, Siwoo Lim, Sunjae Yoon, Suyong Yeon, and Chang D Yoo. Inspace: Structure-aware 3d indoor scene generation from a single 360 image. In *European Conference on Computer Vision*, pp. 60–79. Springer, 2026.

Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European conference on computer vision*, pp. 71–91. Springer, 2024.

Wenbin Li, Sajad Saeedi, John McCormac, Ronald Clark, Dimos Tzoumanikas, Qing Ye, Yuzhong Huang, Rui Tang, and Stefan Leutenegger. Interiornet: Mega-scale multi-sensor photo-realistic indoor scenes dataset. *arXiv preprint arXiv:1809.00716*, 2018.

Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025a.

Yangkai Lin, Jiabao Lei, and Kui Jia. Scenelcm: End-to-end layout-guided interactive indoor scene generation with latent consistency model. *arXiv preprint arXiv:2506.07091*, 2025b.

Zhaoyang Liu, Xinyu Chen, Yikang Li, Xiaogang Wang, and Hongsheng Li. Llm3dscene: Large language model for 3d indoor scene generation. *arXiv preprint arXiv:2312.09149*, 2023.

Quan Meng, Lei Li, Matthias Nießner, and Angela Dai. Lt3sd: Latent trees for 3d scene diffusion. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 650–660, 2025.

Yanxu Meng, Haoning Wu, Ya Zhang, and Weidi Xie. Scenegen: Single-image 3d scene generation in one feedforward pass. In *2026 International Conference on 3D Vision (3DV)*, pp. 543–553. IEEE, 2026.

Despoina Paschalidou, Amlan Kar, Maria Shugrina, Karsten Kreis, Andreas Geiger, and Sanja Fidler. Atiss: Autoregressive transformers for indoor scene synthesis. *Advances in neural information processing systems*, 34:12013–12026, 2021.

Nicholas Pfaff, Thomas Cohn, Sergey Zakharov, Rick Cory, and Russ Tedrake. Scenesmith: Agentic generation of simulation-ready indoor scenes. In *Forty-third International Conference on Machine Learning*, 2026.

Luigi Piccinelli, Yung-Hsu Yang, Christos Sakaridis, Mattia Segu, Siyuan Li, Luc Van Gool, and Fisher Yu. Unidepth: Universal monocular metric depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10106–10116, 2024.

Pulak Purkait, Christopher Zach, and Ian Reid. Sg-vae: Scene grammar variational autoencoder to generate new indoor scenes. In *European Conference on Computer Vision*, pp. 155–171. Springer, 2020.

Xingjian Ran, Yixuan Li, Linning Xu, Mulin Yu, and Bo Dai. Direct numerical layout generation for 3d indoor scene synthesis via spatial reasoning. *arXiv preprint arXiv:2506.05341*, 2025.

Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024.

Mengyi Shan, Zecheng He, Haoyu Ma, Felix Juefei-Xu, Peizhao Zhang, Tingbo Hou, and Ching-Yao Chuang. Populate-a-scene: Affordance-aware human video generation. *arXiv preprint arXiv:2507.00334*, 2025.

Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

Qi Sun, Hang Zhou, Wengang Zhou, Li Li, and Houqiang Li. Forest2seq: Revitalizing order prior for sequential indoor scene synthesis. In *European Conference on Computer Vision*, pp. 251–268. Springer, 2024.

Wenzhuo Sun, Mingjian Liang, Wenxuan Song, Xuelian Cheng, and Zongyuan Ge. Roomplanner: Explicit layout planner for easier llm-driven 3d room generation. *arXiv preprint arXiv:2511.17048*, 2025.

Jiapeng Tang, Yinyu Nie, Lev Markhasin, Angela Dai, Justus Thies, and Matthias Nießner. 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.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 5294–5306, 2025.

Xinpeng Wang, Chandan Yeshwanth, and Matthias Nießner. Sceneformer: Indoor scene generation with transformers. In *2021 International conference on 3D vision (3DV)*, pp. 106–115. IEEE, 2021.

Zijie Wu, Mingtao Feng, Yaonan Wang, He Xie, Weisheng Dong, Bo Miao, and Ajmal Mian. External knowledge enhanced 3d scene generation from sketch. In *European Conference on Computer Vision*, pp. 286–304. Springer, 2024.

Hongchi Xia, Xuan Li, Zhaoshuo Li, Qianli Ma, Jiashu Xu, Ming-Yu Liu, Yin Cui, Tsung-Yi Lin, Wei-Chiu Ma, Shenlong Wang, et al. Sage: Scalable agentic 3d scene generation for embodied ai. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22358–22368, 2026.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:15903–15935, 2023.

Jiazheng Xu, Yu Huang, Jiale Cheng, Yuanming Yang, Jiajun Xu, Yuan Wang, Wenbo Duan, Shen Yang, Qunlin Jin, Shurun Li, et al. Visionreward: Fine-grained multi-dimensional human preference learning for image and video generation. *arXiv preprint arXiv:2412.21059*, 2024.

Jianing Yang, Alexander Sax, Kevin J Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 21924–21935, 2025a.

Yandan Yang, Baoxiong Jia, Peiyuan Zhi, and Siyuan Huang. 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, 2024a.

Yandan Yang, Baoxiong Jia, Shujie Zhang, and Siyuan Huang. Sceneweaver: All-in-one 3d scene synthesis with an extensible and self-reflective agent. *Advances in neural information processing systems*, 38:140319–140351, 2026.

Yixuan Yang, Zhen Luo, Tongsheng Ding, Junru Lu, Mingqi Gao, Jinyu Yang, Victor Sanchez, and Feng Zheng. Optiscene: Llm-driven indoor scene layout generation via scaled human-aligned data synthesis and multi-stage preference optimization. *arXiv preprint arXiv:2506.07570*, 2025b.

Yue Yang, Fan-Yun Sun, Luca Weihs, Eli VanderBilt, Alvaro Herrasti, Winson Han, Jiajun Wu, Nick Haber, Ranjay Krishna, Lingjie Liu, et al. Holodeck: Language guided generation of 3d embodied ai environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16227–16237, 2024b.

Zhaoda Ye, Yang Liu, and Yuxin Peng. Maan: Memory-augmented auto-regressive network for text-driven 3d indoor scene generation. *IEEE Transactions on Multimedia*, 26:11057–11069, 2024.

Lap-Fai Yu, Sai-Kit Yeung, and Demetri Terzopoulos. The clutterpalette: An interactive tool for detailing indoor scenes. *IEEE transactions on visualization and computer graphics*, 22(2):1138–1148, 2015.

Chenyang Yuan and Xiaoyuan Cheng. Planning with language and generative models: Toward general reward-guided wireless network design. *arXiv preprint arXiv:2602.00357*, 2026.

Guangyao Zhai, Evin Pınar Örnek, Dave Zhenyu Chen, Ruotong Liao, Yan Di, Nassir Navab, Federico Tombari, and Benjamin Busam. Echoscene: Indoor scene generation via information echo over scene graph diffusion. In *European Conference on Computer Vision*, pp. 167–184. Springer, 2024.

Haoran Zhang, Cheng Lin, Yang Li, Hongbo Fu, and Hao Zhang. Llmscene: Large language model guided 3d indoor scene synthesis. *IEEE Transactions on Visualization and Computer Graphics*, 2024.

Yisu Zhang, Chenjie Cao, Tengfei Wang, Xuhui Zuo, Junta Wu, Jianke Zhu, and Chunchao Guo. Worldstereo: Bridging camera-guided video generation and scene reconstruction via 3d geometric memories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 40327–40339, June 2026.

Dian Zheng, Ziqi Huang, Hongbo Liu, Kai Zou, Yinan He, Fan Zhang, Lulu Gu, Yuanhan Zhang, Jingwen He, Wei-Shi Zheng, et al. Vbench-2.0: Advancing video generation benchmark suite for intrinsic faithfulness. *arXiv preprint arXiv:2503.21755*, 2025.

Weipeng Zhong, Peizhou Cao, Yichen Jin, Li Luo, Wenzhe Cai, Jingli Lin, Hanqing Wang, Zhaoyang Lyu, Tai Wang, Bo Dai, et al. Internscenes: A large-scale simulatable indoor scene dataset with realistic layouts. *arXiv preprint arXiv:2509.10813*, 2025.

# A Experimental Details

## A.1 Implementation Details and Computational Cost

Training is performed on a single NVIDIA RTX 8000 GPU. Table 3 lists all hyperparameters. Table 4 reports the wall-clock training cost for each stage. Calling the LLM at every SAC step is prohibitively slow. We therefore adopt a hybrid strategy: for most steps the reward $r_t$ is computed by the cheap geometric rule (3D IoU, floor contact, category- specific size ranges), and every $K = 50$ steps the LLM agent performs the full CoT-and-code evaluation, whose score replaces the rule-based reward for that step and calibrates the rule thresholds for the subsequent interval. This keeps the total LLM calls within a few dozen per training run while retaining spatial commonsense in the reward. Inference requires an LLM in the loop for early stopping, incurring per-scene latency orders of magnitude above pure forward methods (ATISS, DiffuScene). A learned stopping criterion distilled from the LLM is left for future work.

## A.2 Data Collection and Curation

The seed images are drawn from two sources: (i) Internet images retrieved with room-type queries (e.g., bedroom interior, living room interior), and (ii) 2D indoor images from the image collection of 3D-FRONT (Fu et al., 2021), which is curated separately from the dataset's 3D scene layouts and carries no layout annotations. The two collections are label-free on the image side. No 3D layout annotation (bounding boxes, category labels, or room meta-data) is ever used. The 3D-FRONT images enter the pipeline as plain 2D images only. We collected 20,000 bedroom and 20,000 living-room images from Google. Since the 40,000 seed images are unlabeled and serve only as initial frames for video synthesis, no train/validation split is applied to them. Concretely, candidate images were retrieved with room-type queries and filtered in two passes. We first removed near-duplicate and non-indoor images, then discarded samples that are unsuitable as an initial video frame: resolution below [512×512], strong blur or motion artifacts, prominent watermarks or text, and scenes dominated by people or non-static content. We further balanced the set across layout styles to avoid over-representing any single template. This yields 40,000 images used as initial frames for LTX-Video. Each image is used as the initial frame for LTX-Video (HaCohen et al., 2024) to synthesize a 5-second video at 24 fps. We then apply the following reconstruction pipeline to extract 3D bounding boxes from each video: Grounded-SAM (Ren et al., 2024) for object segmentation, UniDepth (Piccinelli et al., 2024) for monocular depth estimation, MASt3R (Leroy et al., 2024) for cross-frame correspondence, and Fast3R (Yang et al., 2025a) for 3D reconstruction. Before entering the reconstruction pipeline, each video is trimmed by ViTrim to remove temporally degraded suffixes.

## A.3 Category Space and Asset Matching

The object category set $\mathcal{C}$ is defined by the predefined label vocabulary of Grounded-SAM (Ren et al., 2024), which covers common indoor furniture and fixture types. The cardinality $|\mathcal{C}|$ is not hard-coded in the model architecture; the category head outputs a logits vector of length $|\mathcal{C}|$, so the vocabulary can be extended to new room types or furniture categories without architectural changes by simply adjusting the output dimension and providing corresponding text prompts for segmentation.

In our experiments we set $|\mathcal{C}| = 34$ to align with the 3D-FRONT (Fu et al., 2021) furniture vocabulary, ensuring a fair comparison with supervised baselines that are trained and evaluated on the same category set.

At render time, each predicted bounding box is instantiated with a 3D asset retrieved from the 3D-FRONT library. We select the asset whose aspect ratio $(w{:}d{:}h)$ most closely matches that of the predicted box. This retrieval-based strategy keeps the generation pipeline fully annotation-free while producing visually plausible renderings.

| Parameter | Description | Value |
|---|---|---|
| **General Setup** | | |
| Random Seed | Global random seed | 42 |
| Optimizer | Base optimizer | AdamW |
| $\epsilon$ | Entropy numerical stability | $10^{-8}$ |
| **Transformer Backbone** | | |
| $N_{\max}$ | Maximum generation steps | 20 |
| Batch Size / LR / Epochs | Pretraining settings | 64 / 1e-3 / 10 000 |
| **SAC & LLM Reward** | | |
| $\gamma$ | MDP discount factor | 0.99 |
| Actor LR / RL Batch Size | SAC settings | 3e-4 / 64 |
| $K$ | LLM evaluation interval (hybrid reward) | 50 steps |
| LLM Model | Reward model | Qwen-3.5 |
| LLM Code Iter | Max code revision iterations | 3 |
| **ViTrim Video Curation** | | |
| Window / Step / Frame Int. | Sliding window (dur., step, frame int.) | 1 s / 0.1 s / 0.1 s |
| $\gamma_{\text{penalty}}$ / $\tau$ / $\beta_\sigma$ | Penalty / hard IoU / soft-std weights | 0.5 / 0.5 / $-0.5$ |

Table 3: Hyperparameters of SceneCast.

| **Stage** | **Time** | **GPU Mem.** |
|---|---|---|
| Backbone pretraining | $\sim$36 h | 3–5 GB |
| SAC fine-tuning (incl. LLM) | $\sim$15-20 min | 1–2 GB |
| LLM reward (per step) | $\sim$10-30 s | – |

Table 4: Wall-clock training cost on a single RTX 8000. SAC fine-tuning time includes LLM inference and code execution overhead.

### A.4 Evaluation Metrics

All metrics are computed over 1,000 synthesized scenes per room type. Following the protocol of (Paschalidou et al., 2021; Tang et al., 2024), both generated and ground-truth scenes are rendered as $256 \times 256$ top-down semantic maps via orthographic projection with a unified camera setup, where each object is colored by its semantic category.

**FID** (Fréchet Inception Distance) measures the distance between the feature distributions of generated and real scene renderings; lower values indicate better fidelity and diversity.

**KID** ($\times 0.001$, Kernel Inception Distance) is an unbiased variant of FID that uses a polynomial kernel; lower is better.

**SCA** (Scene Classification Accuracy) is the accuracy of a binary classifier trained to discriminate real from synthetic scene renderings. A score closer to 50% indicates that generated scenes are indistinguishable from real ones.

**CKL** ($\times 0.01$, Category KL Divergence) measures the KL divergence between the semantic class distributions of synthesized and ground-truth scenes; lower values indicate better alignment with the real category distribution.

**PIoU** ($\times 0.01$, Pairwise Intersection-over-Union) computes the average 3D bounding-box IoU across all object pairs within each generated scene, then averages over the evaluation set. Unlike FID/KID/SCA, which operate on 2D renderings, PIoU is computed directly from the 3D bounding boxes and therefore provides a direct measure of physical plausibility: lower values indicate fewer inter-object collisions.

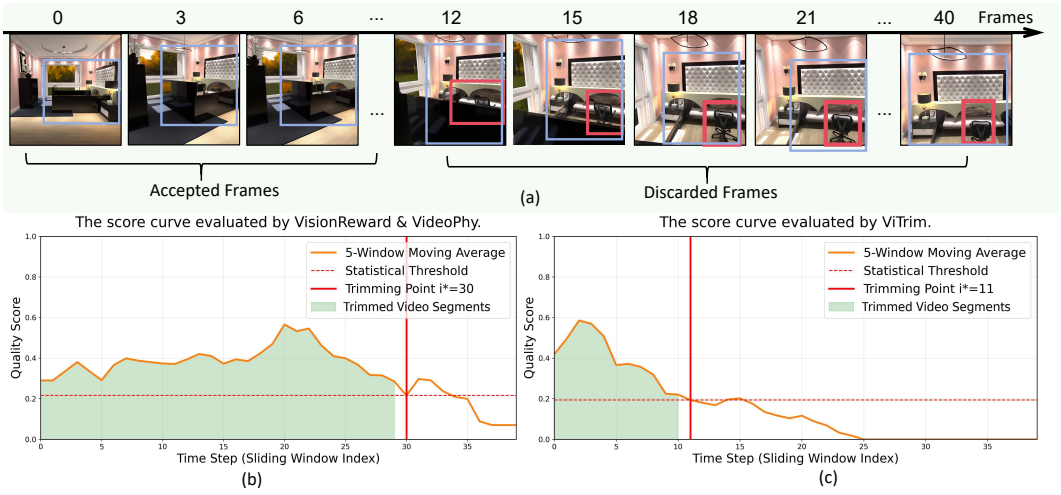



Figure 6: Qualitative evidence for the spatial IoU ablation (Table 2 (b), row 3 vs. row 5). (a) A generated bedroom video; a chair interpenetrates the bed at Frame 11 (red box). (b) VisionReward + VideoPhy score curves (row 3 setting): the interpenetration is not detected, and low-quality segments pass the threshold. (c) ViTrim fused score with spatial IoU (row 5): the quality drop is identified at Frame 11, and the corrupt suffix is discarded.

# B More Experiments

## B.1 More Qualitative Comparisons

We provide additional qualitative comparison results in Figure 7 and Figure 8.

## B.2 ViTrim Score Curve Visualization

Table 2 (b) in the main text shows that removing the spatial IoU signal (row 3) degrades downstream generation quality. Figure 6 provides qualitative evidence for this gap. We take a generated bedroom video (40 frames sampled at 0.1 s intervals) in which a chair interpenetrates the bed beginning at Frame 11 (Figure 6a, red box) — a typical failure mode of video generation models. Figure 6(b) shows the quality scores produced by VisionReward and VideoPhy alone (the setting of row 3 in Table 2 (b)). The score curve remains entirely flat across Frame 11; the interpenetration does not trigger a quality drop, and substantial portion of corrupt segments passes the statistical threshold. This explains why the VR+VP-only variant retains noisy training samples and yields higher FID. Figure 6(c) shows the fused score of ViTrim (row 5), which incorporates the spatial IoU signal $U_i^{\text{spatial}} = 1 - IoU_i$ alongside VisionReward and VideoPhy. The quality curve turns precisely at Frame 11, coinciding with the onset of interpenetration. By integrating the spatial IoU signal, ViTrim accurately locates the quality-degradation boundary and discards the corrupt suffix, retaining a valid prefix of approximately 1.5 s (15 frames) for this video.

## B.3 Effect of Different LLMs

To verify that our prompting strategy is not tightly coupled to a specific LLM, we replace the reward model with three alternatives spanning open-source and commercial APIs: Qwen-2.7-image, Qwen-3-vl-plus, and Qwen-3.5. As shown in Table 5 and Figure 9, all three LLMs produce structurally sound layouts, with FID improving as model capability increases. This confirms that the code-execution reward mechanism is portable across LLM backends.

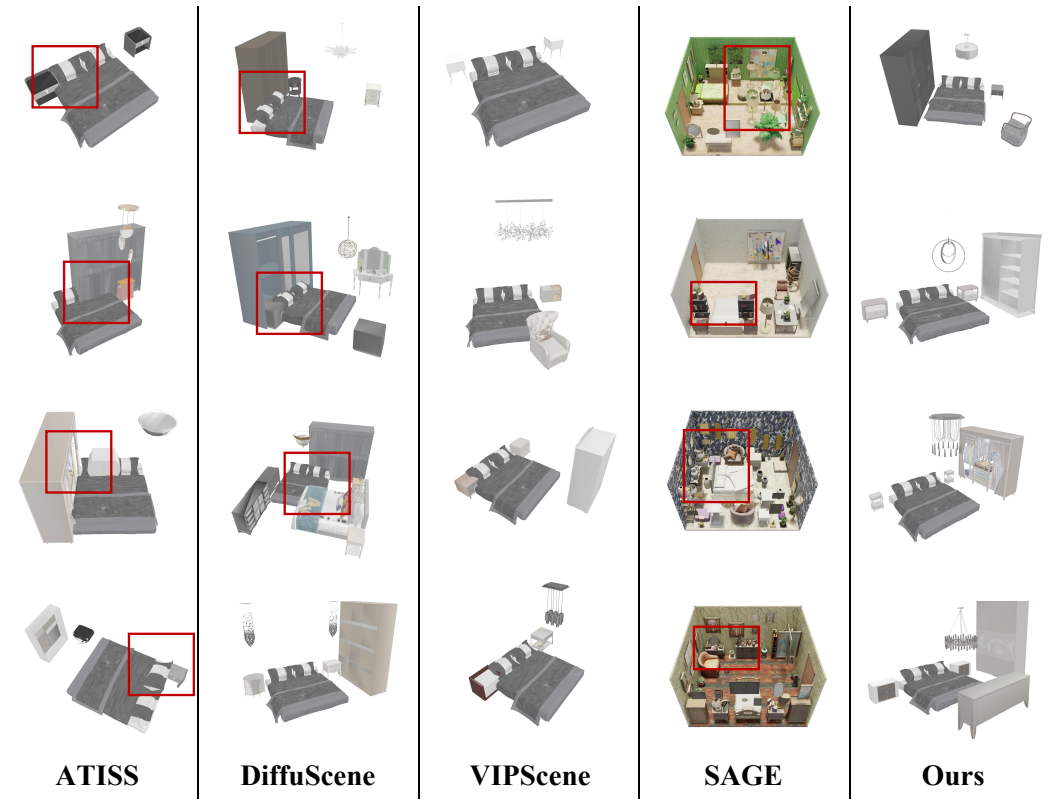



Figure 7: Qualitative Comparison in Bedroom.

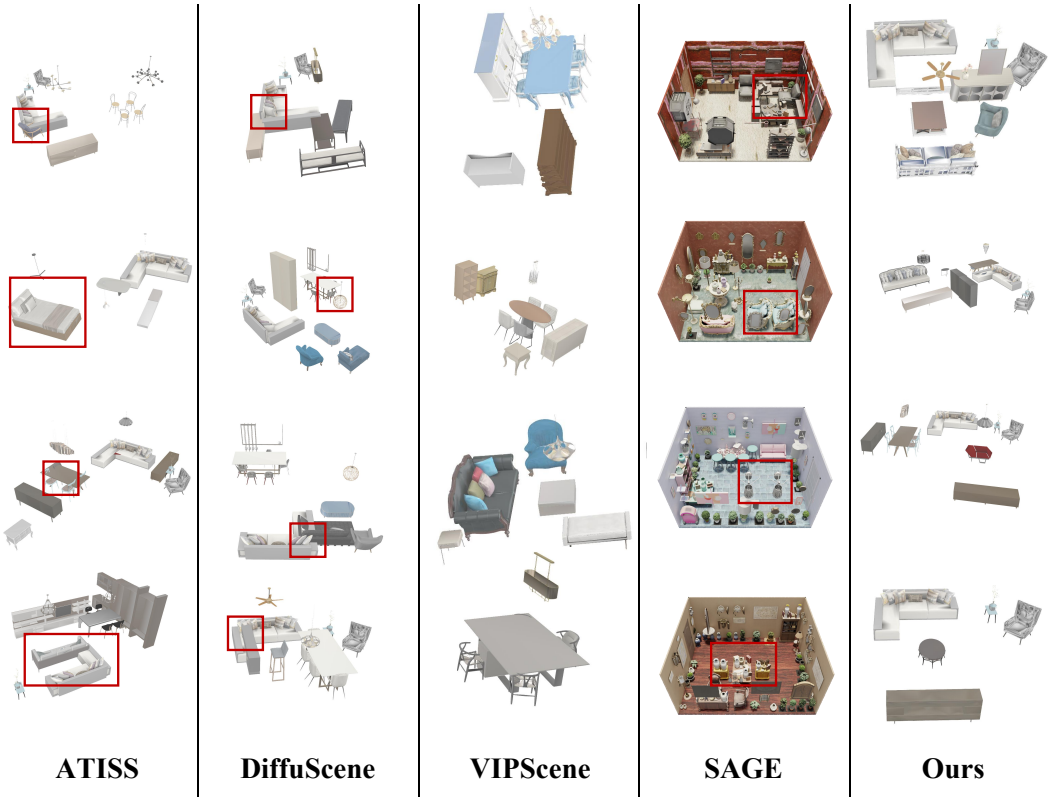
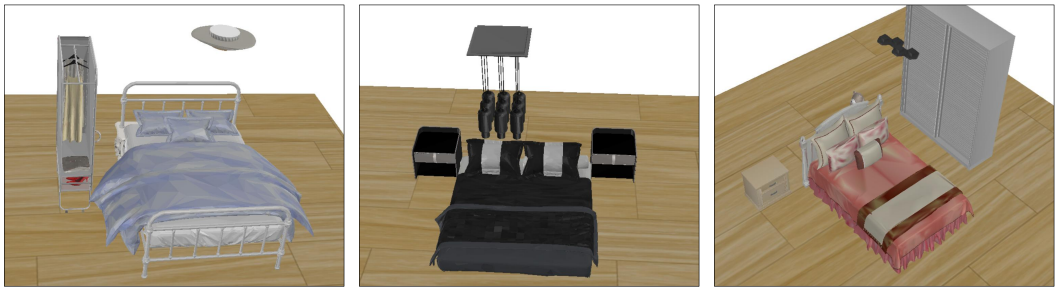



Figure 8: Qualitative Comparison in LivingDiningroom.

| LLM | FID ↓ | SCA % |
|---|---|---|
| Qwen-2.7-image | 142.3 | 76.1 |
| Qwen-3-vl-plus | 128.5 | 65.6 |
| Qwen-3.5 | 111.1 | 55.7 |

Table 5: Scene generation quality under different LLM reward backends (bedroom).



**Qwen-2.7-image** **Qwen-3-vl-plus** **Qwen-3.5**

Figure 9: Generated scenes using different LLMs as reward backends. All variants preserve structural integrity.

### B.4 Effect of Prompting Strategy

The main-text unified reward ablation (Table 2 (a)) isolates the contribution of each reward component, where "LLM w/o code" corresponds to P3 below and "LLM w/ code (full)" corresponds to P4. Here we additionally report two simpler prompting baselines (P1, P2) that precede the CoT design.

- **P1 (Standard Prompting).** A direct task definition that asks the LLM to score the layout without multi-step reasoning or task decomposition.
- **P2 (Zero-shot CoT).** Appends "Let's think step by step" to decompose the scoring task into sub-problems, without providing demonstrations.
- **P3 (Few-shot CoT).** Embeds human-crafted chain-of-thought demonstrations in the prompt, guiding the LLM to imitate a standard reasoning paradigm. This is the "LLM w/o code" row in Table 2 (a).
- **P4 (Code-Execution CoT).** Further allows the LLM to generate and execute Python code during reasoning for precise geometric computation. This is the full model ("LLM w/ code") in Table 2 (a).

Table 6 shows a consistent improvement from P1 to P4. The largest single jump occurs between P3 and P4 (FID: $141.3 \rightarrow 111.1$), confirming that the ability to execute geometric code —- rather than CoT prompting alone —- is the primary driver of reward accuracy. Figure 10 visualizes the corresponding scene outputs.

### B.5 Annotation-free adaptation to new room types

To test whether the learned layout prior generalizes beyond the bedroom and living/dining room types used in the main evaluation, we apply SceneCast to two room types not included in the quantitative benchmarking: *kitchens* and *studies*. For each room type, we collect a smaller set of seed images from the Internet (approximately 5,000 per type), generate and trim videos with the same pipeline as the main experiments, and define a room-specific category vocabulary for Grounded-SAM segmentation. No 3D-FRONT annotations are used at any stage; the only change relative to the main experiments is the seed image query and the category label set.

Figure 11 visualizes the generated layouts as colored bounding boxes (each color denotes a distinct category), decoupling layout quality from asset rendering. The generated layouts respect room-specific conventions, confirming that the learned spatial prior is not tied to bedroom/living-room

| | Description | FID ↓ | SCA % |
|---|---|---|---|
| P1 | Standard Prompting | 159.5 | 79.5 |
| P2 | Zero-shot CoT | 156.5 | 76.5 |
| P3 | Few-shot CoT | 141.3 | 64.3 |
| P4 | Code-Execution CoT (ours) | 111.1 | 55.7 |

Table 6: Progressive prompting ablation (bedroom). P3 and P4 correspond to the last two rows of Table 2 (a) in the main text.

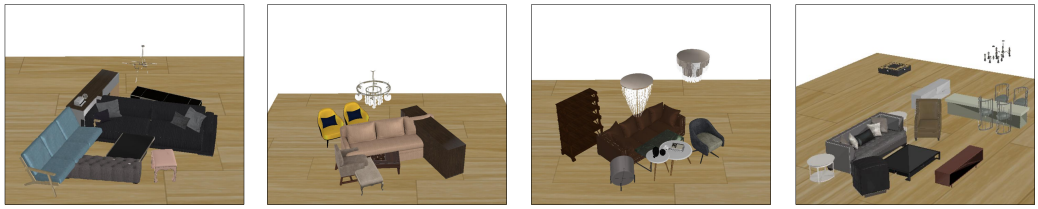



Figure 10: Generated scenes under different prompting strategies (P1–P4).

patterns. These results confirm that the spatial commonsense acquired through LLM-guided RL is not specific to the two evaluation room types, and that the pipeline generalizes to spatially distinct environments without architectural modifications.

We note that this experiment also implicitly demonstrates category extensibility: the kitchen and study vocabularies include categories absent from the 34-class bedroom/living-room set, requiring only an expanded category head and corresponding segmentation prompts. Quantitative evaluation on these room types is deferred to future work, as 3D-FRONT provides insufficient ground-truth scenes for kitchen and study to compute reliable distributional metrics.

# C Algorithm and Model Details

## C.1 SAC Formulation

We adopt the Soft Actor-Critic (SAC) algorithm (Haarnoja et al., 2018) to optimize the policy. The actor network $\pi_\theta$ maximizes expected reward with an entropy regularization term weighted by $\alpha$:

$$\mathcal{L}_\pi = -\mathbb{E}_{a_t \sim \pi_\theta} \left[ Q_\phi(s_t, a_t) - \alpha \log \pi_\theta(a_t | s_t) \right]. \tag{4}$$

The critic network $Q_\phi$ is updated via the Bellman backup:

$$\mathcal{L}_Q = \mathbb{E}_{(s_t, a_t) \sim \mathcal{D}} \left[ \left( Q_\phi(s_t, a_t) - (r_t + \gamma V(s_{t+1})) \right)^2 \right], \tag{5}$$

where $\mathcal{D}$ is the experience replay buffer, $V(\cdot)$ is the soft state value function, and $r_t$ is the scalar reward returned by the LLM Agent. In our setting, the actor $\pi_\theta$ outputs an action $a_t \in \mathcal{A}$ that adjusts the bounding-box attributes $(p_v, u_v, d_v)$ of the current layout, so each SAC step produces a refined scene layout that is immediately evaluable by the LLM reward. Note that the actor and critic networks are trained as usual, and the LLM agent never modifies the scene: the corrected Size/Position values shown in Figure 5 are byproducts of its chain-of-thought reasoning, which serve only to justify the scalar reward, not as executed actions.

## C.2 Transformer Backbone: Differences from ATISS

The Transformer backbone follows the autoregressive architecture of ATISS (Paschalidou et al., 2021) with three modifications.

**(1) Visual encoder replaces the floor-plan encoder.** ATISS conditions generation on a rasterized floor-plan image. Since our input is an unlabeled Internet photograph $\mathcal{I}$ with no floor-plan annotation, we replace the floor-plan encoder with a visual encoder: a pre-trained DINOv2 ViT-L/14

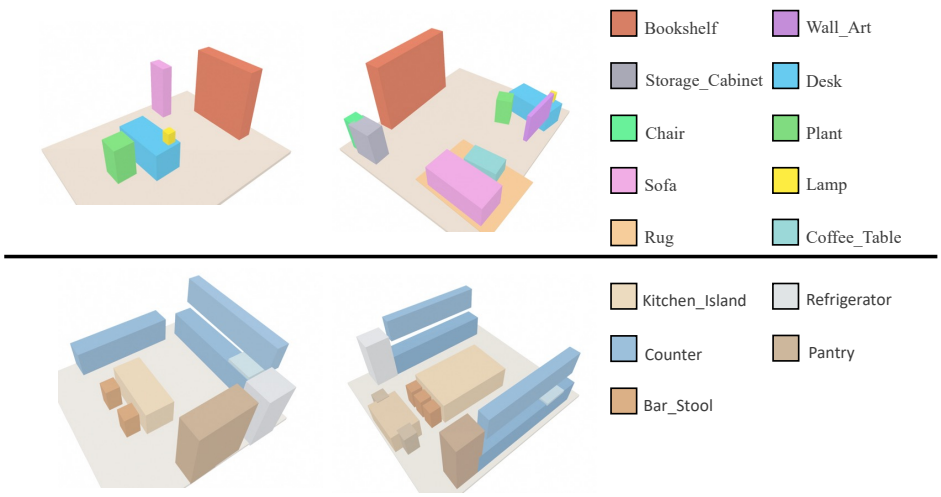



Figure 11: Generated layouts for unseen room types (kitchen, study), visualized as category-colored bounding boxes. The model produces spatially plausible arrangements for room types not included in the main quantitative evaluation.

extracts a 768-dimensional global feature from $\mathcal{I}$, which is then projected to a 64-dimensional scene context vector $\mathbf{F}_{\mathcal{I}} \in \mathbb{R}^{64}$ via a two-layer MLP. This vector serves the same role as the floor-plan feature in ATISS, conditioning all subsequent object predictions.

**(2) No end-of-sequence token.** ATISS uses a learnable EOS token to terminate generation. Because our training data extracted from video reconstructions contain only partial (incomplete) scene layouts, an EOS token would receive noisy supervision. We therefore remove it and fix the total number of generation steps to $N_{\text{max}}$. The effective object count $N \leq N_{\text{max}}$ is determined at inference time by the LLM early-stopping signal $R_t^{\text{c}}$.

**(3) Training on incomplete layouts.** ATISS is trained on complete, annotated scenes from 3D-FRONT. Our training set consists of partial layouts reconstructed from trimmed video clips, so the model learns to *complete* a scene rather than reproduce a full annotation. The pre-training objective is the same permutation-invariant autoregressive negative log-likelihood as in ATISS, applied to the available objects in each partial layout.

All remaining components—the structure encoder $h_\theta : (c_v, p_v, u_v, d_v) \mapsto \mathbf{C}_v \in \mathbb{R}^{64}$, the 6-layer multi-head self-attention Transformer encoder with a learnable object query $q \in \mathbb{R}^{64}$, and the autoregressive attribute extractor (category $\rightarrow$ position $\rightarrow$ rotation $\rightarrow$ size, each via a 2-layer MLP with mixture-of-logistics output)—follow ATISS without modification.

### C.3 ViTrim: Normalization and Implementation Details

Section 3.3 presents the core ViTrim formulation. Here we provide the normalization and implementation details omitted from the main text for brevity.

**Score normalization.** The raw VisionReward score $S_i$ and VideoPhy score $P_i$ are both monotonically increasing (higher is better). We align them with $U_i^{\text{spatial}} = 1 - \text{IoU}_i$ (higher means better spatial quality) via min-max normalization:

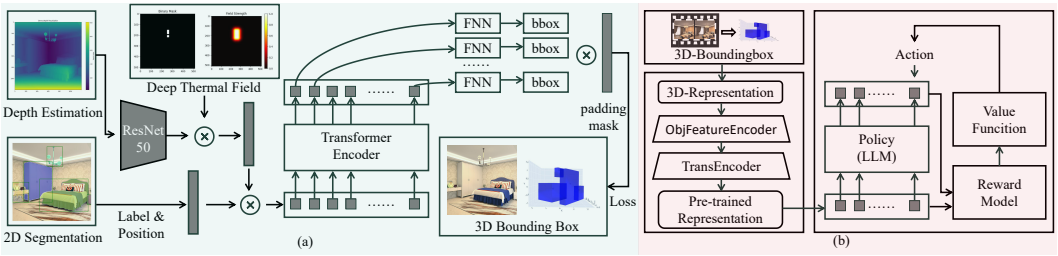



Figure 12: 2D-to-3D bounding box extraction pipeline. A monocular RGB image is processed by parallel segmentation and depth branches, followed by a Transformer encoder that aggregates object-centric features for 3D box regression.

$$U_i^{\text{human}} = \frac{S_i - \min(S)}{\max(S) - \min(S)}, \tag{6}$$

$$U_i^{\text{phy}} = \frac{P_i - \min(P)}{\max(P) - \min(P)}, \tag{7}$$

where $S = [S_1, \ldots, S_X]^\top$ and $P = [P_1, \ldots, P_X]^\top$ are the score sequences over all $X$ segments. After normalization, all three scores $U_i^{\text{human}}, U_i^{\text{phy}}, U_i^{\text{spatial}} \in [0, 1]$, and a **larger value indicates better quality** along that dimension.

**Spatial IoU computation.** For each segment $V_i$, we extract 3D bounding boxes via the reconstruction pipeline and compute the average pairwise IoU:

$$IoU_i = \frac{2}{M(M-1)} \sum_{v<v'} \text{IoU}_{3\text{D}}(f_v, f_{v'}), \tag{8}$$

where $M$ is the number of extracted objects in $V_i$. The spatial quality score is then $U_i^{\text{spatial}} = 1 - \text{IoU}_i$, so that severe overlap **lowers** the score (worse quality), consistent with the other two dimensions after normalization.

**Numerical stability.** When $\max(\mathbf{S}) = \min(\mathbf{S})$ (all segments receive the same raw score), we set $U_i^{\text{human}} = 0.5$ for all $i$; likewise for $U^{\text{phy}}$. In the entropy computation (Eq. 1 in the main text), when $\sum_{i=1}^{X} U_i^j = 0$, we set $p_i^j = 1/X$ uniformly. The smoothing term $\epsilon = 10^{-8}$ prevents $\ln(0)$.

**Cumulative error normalization.** The raw cumulative spatial error Ecum $E_i^{\text{cum}} = \sum_{k=2}^{i}(U_1^{\text{spatial}} - U_k^{\text{spatial}})$ defined in Section 3.3 (used in Eq. 3)

### C.4 2D-TO-3D BOUNDING BOX EXTRACTION

This module is an optional standalone alternative for single-image input and is not used in the main experiments (Sec. 4.1–4.2), which rely solely on the video-based reconstruction pipeline of Appendix A.2. Its 3D-FRONT-based supervision therefore does not affect the annotation-free property of the main pipeline. In addition to the video-based reconstruction pipeline used in the main experiments, we provide an alternative monocular 3D bounding box prediction model for extracting 3D objects from single 2D images. Figure 12 illustrates the architecture.

Given a single RGB image, a ResNet-50 backbone extracts multi-scale visual features. Two parallel branches perform 2D semantic segmentation (producing object-level masks) and monocular depth estimation. The masked object regions and depth-aware features are fed into an object-centric Transformer encoder that aggregates discriminative representations per object. A multi-layer feed-forward network then regresses the 3D bounding box parameters $(p_v, u_v, d_v)$ for each detected object, conditioned on its category $c_v$ and 2D box prior. The model is trained on 3D-FRONT (Fu et al., 2021) with a joint multi-task loss supervising 2D segmentation, depth estimation, and 3D box regression. This pipeline serves as a lightweight alternative when video input is unavailable.

# D LIMITATIONS AND FUTURE WORK

**Category and asset scope.** Our quantitative evaluation uses $|\mathcal{C}|=34$ aligned with 3D-FRONT to ensure a fair comparison with supervised baselines. Although the pipeline generalizes to unseen room types and category vocabularies (Appendix B.5), the current asset retrieval relies on the 3D-FRONT model library, which limits visual diversity for categories not covered by this library. Incorporating larger asset repositories or generative 3D asset models would broaden applicability.

**Reconstruction fidelity.** Training data quality is bounded by the video reconstruction pipeline: when the generated video exhibits severe occlusion or degenerate camera motion, the extracted 3D bounding boxes carry noise that propagates into the learned distribution. A related concern is that partially observed objects may yield bounding boxes whose centers and extents are systematically underestimated, potentially biasing the model toward generating objects smaller than those in the real distribution. ViTrim alleviates this by discarding clips in which objects are severely incomplete or interpenetrating, and our aspect-ratio-based asset matching (Appendix A.3) instantiates each predicted box with a proportionally consistent 3D model, so that mild scale bias does not translate into visual distortion; nevertheless, neither fully removes the bias. More robust reconstruction methods, explicit uncertainty estimation during extraction, and incorporating category-specific scale priors into the reward would further improve data quality.