SCENEBLOOM: AGENTIC INDOOR ENVIRONMENT GENERATION WITH REALISTIC AND DIVERSE LAYOUTS FOR EMBODIED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Training generalizable embodied agents requires realistic, varied, and simulation-ready indoor environments. Language-guided methods support semantic planning and object-level reasoning, but precise spatial grounding remains difficult. Visual generative models offer rich priors over plausible arrangements, yet their outputs require geometric reasoning to become physically valid 3D scenes. We introduce SceneBloom, an agentic framework that combines visual layout sampling with contract-guided 3D compilation. A hierarchical sampler uses diffusion priors to generate furniture arrangements within functional zones. A compiler then uses relations extracted from each proposal to guide geometric fitting and local pose correction. Across five room types, SceneBloom attains the highest visual quality and physical stability among the compared methods and is the only method with no detected mesh collisions or out-of-bounds objects. Its layouts are also closer to the distribution of real rooms than those of most baselines. Ablations show that visual proposals outperform textual proposals in both physical plausibility and visual quality, and that contract guidance further improves layout coherence and realism. Fine-tuning an object-navigation policy on these scenes improves its success on unseen goals in both generated and real environments, and raising scene difficulty through SceneBloom further improves success in real environments.

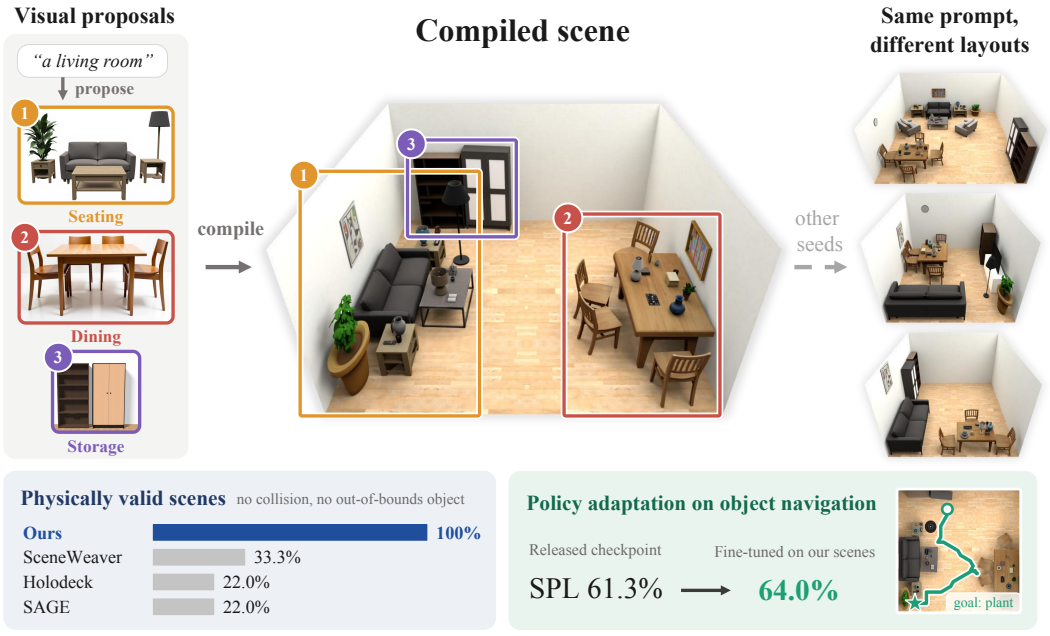

Figure 1: **Overview of SceneBloom.** SceneBloom compiles visual proposals of functional zones into a simulation-ready 3D scene, and different samples of the same prompt yield different layouts. Its scenes are physically valid far more often than those of prior generators, and fine-tuning an object-navigation policy on them improves its performance.

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.

1 INTRODUCTION

Learning-based embodied agents benefit from training across varied environments. Prior work shows that procedurally generated environments support generalization across embodied tasks, and that language-guided scene generation can improve navigation in new scene types (Deitke et al., 2022; Yang et al., 2024b). These findings motivate generators that provide diverse indoor layouts while retaining the spatial organization of real environments.

Existing generators obtain layout structure from hand-crafted placement rules or learned distributions of 3D configurations (Deitke et al., 2022; Tang et al., 2024; Yang et al., 2024a), while language-guided and agentic systems combine flexible scene specification with spatial reasoning and iterative refinement (Yang et al., 2024b; 2026; Xia et al., 2026; Pfaff et al., 2026). Image diffusion models offer a complementary prior over how objects are arranged together. Recent methods exploit generated images and visual guidance to construct realistic 3D scenes (Wang et al., 2024a; Gu et al., 2025; Ling et al., 2026). This suggests a division of responsibilities in which semantic planning establishes functional organization and visual sampling proposes detailed arrangements. The challenge is to carry those arrangements into geometrically valid 3D scenes.

Visual plausibility alone is insufficient for this realization. Depth, scale, and pose errors from 3D lifting produce overlapping or misaligned furniture when the reconstructed objects are assembled, so geometric correction is necessary. Moving or rotating objects to resolve these conflicts, however, can break the relationships that made the sampled arrangement plausible, for example by turning a chair away from its desk. Our guiding insight is to separate the qualitative relationships to preserve from the numerical poses that must be repaired.

We introduce SceneBloom, an agentic framework that implements this separation through two stages, *hierarchical visual sampling* and *contract-guided 3D compilation*. In the first stage, a vision–language planner divides the room into functional zones, using retrieved exemplar rooms as references, and specifies each zone's furniture inventory and approximate floor region. A diffusion model then samples a visual proposal of each zone's furniture arrangement, so that zoning supplies room-level structure while each proposal models one furniture group jointly. The diffusion model uses asset-specific adapters, each trained on renders of one catalog asset, so that the proposals depict the objects used for 3D construction. In the second stage, the proposals are lifted into 3D. Before any object is moved, a vision–language model extracts *relation contracts* from the proposal images, recording relationships such as facing, alignment, and proximity among the lifted objects. A hierarchical compiler then fits furniture within zones, assembles the zones into a room, and refines object poses to restore the contract relations. The contracts thus fix the relationships to preserve, while the compiler repairs only the numerical poses.

Across five room types, SceneBloom scores highest on every visual-quality axis and on physical stability. It is also the only method with no detected mesh collisions or out-of-bounds objects. Its layouts more closely match real indoor layout distributions than SAGE and Holodeck on both Matterport3D and Scan2CAD. Ablations show that visual proposals outperform textual proposals on every physical and visual-quality metric, and that contract guidance further improves layout coherence and realism. Our scenes also keep the walkable floor connected far more often than the same objects placed at random. An object-navigation policy fine-tuned on these scenes achieves the highest success on unseen goals in both held-out generated and real environments, above continued training on the real scans used for pretraining. Adding interior occluders to better match the difficulty of real homes further improves success in real environments.

Our key contributions are:

- We introduce an agentic framework for generating diverse, realistic, simulation-ready indoor environments for embodied learning. It combines diffusion-based layout sampling with contract-guided 3D compilation to preserve functional organization while resolving geometric conflicts.
- We show that our scenes are both visually realistic and physically valid, and that their layouts align more closely with real rooms than those of most prior generators.
- We demonstrate that policies trained on our scenes transfer to real environments better than additional real-scan training, and that controllable scene difficulty strengthens this transfer.

2 Related Work

3D Indoor Scene Synthesis Procedural methods assemble indoor scenes under hand-crafted spatial rules (Deitke et al., 2022; Raistrick et al., 2024), while autoregressive and diffusion models learn object configurations from annotated 3D scenes (Paschalidou et al., 2021; Tang et al., 2024; Lin & Mu, 2024). Physically guided variants further incorporate geometric constraints into layout sampling (Yang et al., 2024a; Li et al., 2026). The arrangements of all these approaches remain bounded by the rules they encode or the 3D scenes they learn from. To draw on broader visual knowledge, image-guided methods generate scene proposals through whole-scene synthesis or progressive inpainting, then recover and refine 3D object placements (Gu et al., 2025; Wang et al., 2024a; Ling et al., 2026). Although these methods bring rich visual priors into scene generation, image-level plausibility does not ensure compatibility with actual asset geometry, and correcting the resulting conflicts can disrupt the relationships that make the proposed layout functional. Our approach addresses both problems, since its proposals depict the actual assets to be placed and its compiler repairs their geometry under relation contracts extracted from the proposals.

Language-Guided and Agentic Scene Generation Language-guided methods prompt an LLM to write an object inventory and spatial constraints, which a solver satisfies (Yang et al., 2024b). Agentic systems turn generation into a feedback loop, in which a planner selects generation and editing tools step by step (Yang et al., 2026; Xia et al., 2026) or specialized agents divide scene design, critique, and workflow control (Pfaff et al., 2026; Ran et al., 2026). Rendered views, geometric checks, and physics simulation guide each revision. Other work structures layout decisions through functional-zone graphs, semantic and physical decomposition, or compilable layout programs (Zhang et al., 2026a; Luo et al., 2026; Tang et al., 2026). Iterative refinement, however, repeats tool calls, rendering, and review, which slows generation (He et al., 2026), and errors in object orientation or in coordination across stages can leave arrangements functionally inconsistent (Yang et al., 2026; Pfaff et al., 2026). Our framework uses agentic planning for functional zoning and diffusion sampling for furniture arrangement. Each zone's arrangement is thus sampled in one pass rather than revised object by object, and relation contracts keep it functional during 3D compilation.

Synthetic Environments for Embodied Learning Synthetic environments provide controllable variation in the scenes and interactions for policy learning. Procedural and language-guided generators expand training coverage through alternative layouts, scene types, and object categories (Deitke et al., 2022; Yang et al., 2024b; Wang et al., 2024b). Studies of synthetic navigation datasets further show that scene realism matters alongside dataset size, motivating attention to object occurrence and spatial arrangement patterns (Khanna et al., 2024). For manipulation, simulation frameworks pair varied environments and tasks with demonstrations (Wang et al., 2023; Nasiriany et al., 2024; Xia et al., 2026; Nasiriany et al., 2026). Real-to-sim approaches construct training scenes from observations, including families of alternative assets and configurations that retain relevant geometric and semantic affordances (Dai et al., 2024; Ranawaka et al., 2026). Together, these directions motivate training environments that combine variation with realistic spatial organization. SceneBloom provides both, and its generation settings can also adjust scene difficulty for policy learning.

3 Method

Given a room description, SceneBloom generates a simulation-ready 3D scene in two stages (Figure 2). Hierarchical visual sampling (Section 3.1) divides the room into functional zones and samples a visual proposal of each zone's furniture arrangement. Contract-guided 3D compilation (Section 3.2) lifts these proposals into 3D, fits them into the room, and restores their functional relations before adding wall and surface objects. This division lets the visual prior supply plausible arrangements while compilation makes them physically valid.

3.1 Hierarchical Visual Sampling

Hierarchical visual sampling divides the work between room-level planning and zone-level arrangement. A vision–language model (VLM) plans the functional zones of the room, and a diffusion model then proposes the furniture arrangement of each zone as an image. Each proposal thus depicts one coherent furniture group, while the zone plan provides room-level structure.

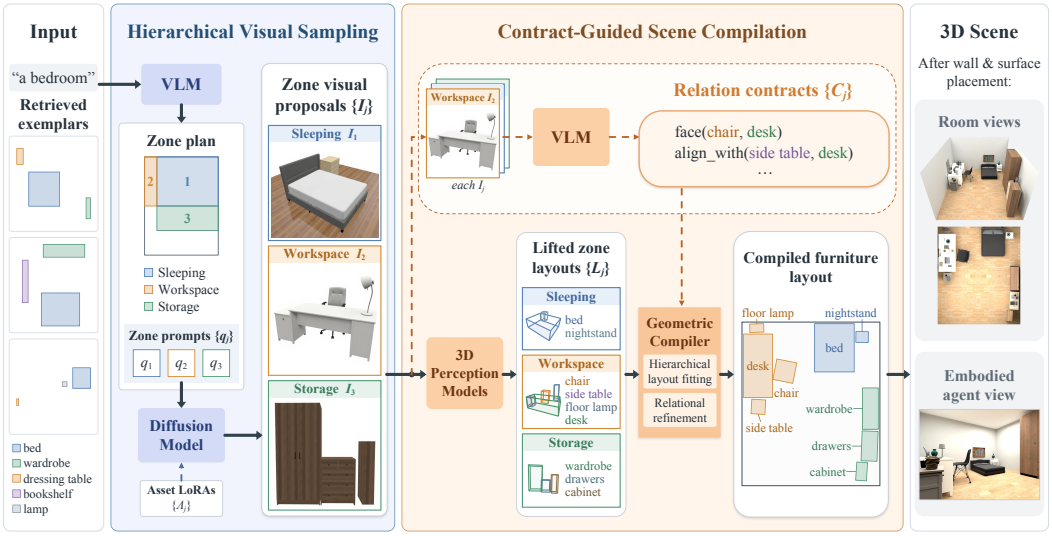

Figure 2: **Framework overview.** SceneBloom decouples visual layout sampling from 3D scene compilation. Given a room description and retrieved exemplars, a vision–language model (VLM) produces a zone plan with a prompt q_j for each zone, and a diffusion model conditioned on q_j and asset LoRAs Λ_j samples zone visual proposals I_j. 3D perception models lift the proposals into zone layouts L_j, and a VLM extracts relation contracts $\mathcal{C}_j$ from the proposal images. Guided by these relation contracts, the geometric compiler applies hierarchical layout fitting and relational refinement to produce the compiled furniture layout. Wall and surface objects are then added, yielding the simulation-ready scene shown in room and agent views.

Functional zoning. Deciding which furniture a room needs and how the room divides into functional zones requires commonsense knowledge that a VLM has. The VLM therefore determines the room dimensions and divides the room into functional zones $\{z_j\}_{j=1}^{J}$, using retrieved exemplar rooms as references (Appendix A.2). Each zone z_j specifies a plan region, a furniture inventory F_j with categories and counts, an anchor category for its central piece, the wall it adjoins, and a zone prompt q_j. Placement within each zone is left to visual proposals and geometric compilation.

Visual layout proposals. Image diffusion models learn from large image collections how furniture is typically arranged. We use this prior to propose the arrangement of each zone. Because the model generates an entire furniture group jointly, a proposal determines all relative placements at once rather than through sequential per-object decisions. For zone z_j, a text-to-image diffusion model G is conditioned on the zone prompt q_j and on asset LoRA adapters Λ_j selected according to F_j:

$$I_j = G(q_j, \Lambda_j; \epsilon_j), \tag{1}$$

where I_j is the proposal image and ϵ_j is the diffusion seed. Each adapter is trained on renders of one catalog asset (Ruiz et al., 2023), so that the proposal depicts the assets used for 3D construction. Adapter fusion (Liu et al., 2025) allows several of these assets to appear in one image. Varying the seed ϵ_j then yields diverse arrangements of the same assets, each grounded in the objects to be built. Appendix A.2 describes adapter training and fusion.

3.2 Contract-Guided 3D Compilation

Lifting a proposal into 3D introduces two kinds of error: geometric conflicts between objects and disrupted functional relations. Hierarchical layout fitting resolves the former, and relation contracts guide the recovery of the latter.

Lifting and relation extraction. Monocular 3D perception (Yao et al., 2026), with depth and camera intrinsics from a monocular estimator (Bochkovskiy et al., 2025), estimates a box for each object in I_j and yields a zone layout L_j with each object's category, dimensions, position, and orientation. To recover relations that geometric separation cannot restore, such as a nightstand

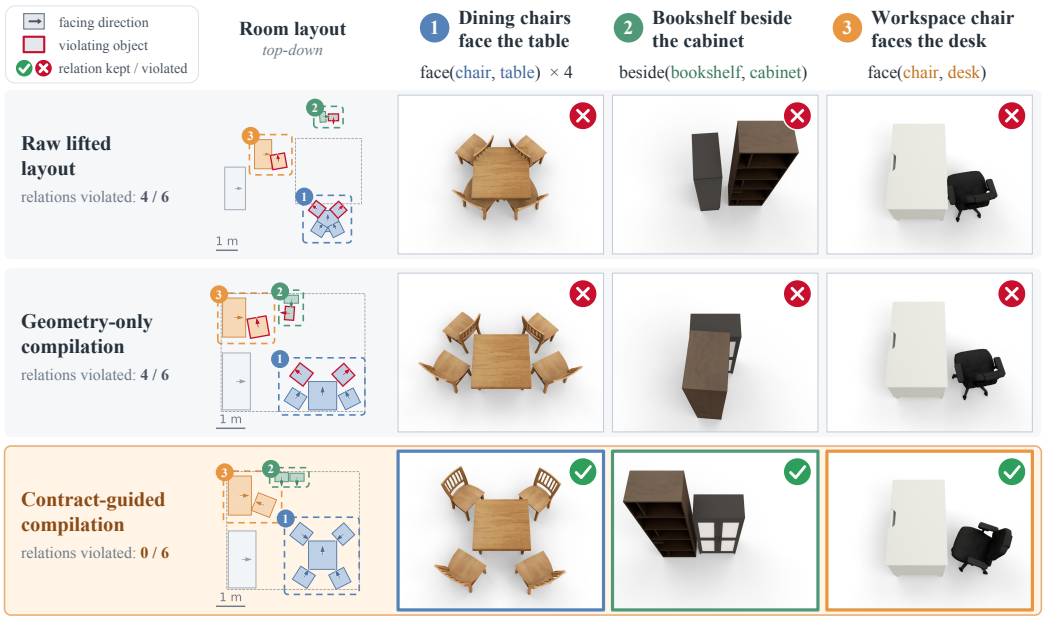

Figure 3: **Effect of contract-guided compilation.** All rows start from the same lifted scene, and each adds one step; the enlarged views show relation groups from the contract. *Raw lifted layout*: objects keep their lifted poses, and lifting errors break relations such as chairs facing their table. *Geometry-only compilation*: fitting assembles a collision-free room but leaves these relations broken. *Contract-guided compilation*: fitting under the contract, followed by relational refinement, restores every relation while the layout stays collision-free.

beside its bed, the VLM extracts a relation contract $\mathcal{C}_j = \{(u, r, v, \gamma)\}$ from the proposal image and grounds it to the objects of L_j. In each entry, object u holds relation r to a target object or wall v with confidence γ. The contract is fixed before compilation. Unlike language-authored constraints that prescribe a layout (Yang et al., 2024b), it records relations present in the sampled arrangement. Its high-confidence relations, which we call binding, guide the rest of compilation (Appendix A.3).

Hierarchical layout fitting. We resolve geometry within zones and then assemble the zones according to the zone plan, reconciling the planned zone regions with the footprint that the proposed furniture actually needs. *Within each zone*, the anchor of z_j is aligned with the zone's wall and faces into the room. The remaining objects of L_j are then separated from the fixed anchor and from one another until their footprints no longer overlap, with $\mathcal{C}_j$ guiding how related objects are separated. The fitted objects define the zone envelope E_j, which is then frozen so that room-level assembly moves the zone only as a rigid whole. *Among zones*, the zone plan fixes for each pair of zones (z_i, z_j) a separating axis and their order along it. Along that axis, an envelope bound to a wall stays flush against it, any other envelope keeps its planned position where possible, and otherwise assembly shifts the later envelope along the axis until it starts at least a clearance gap after the earlier one ends. The room is enlarged only when the envelopes cannot fit within it. Fitting thus resolves geometric conflicts while preserving the arrangement of each proposal within its zone.

Relational refinement. After assembly, geometric predicates check each binding relation of $\mathcal{C}_j$. Relation-specific operators propose local rotations or translations to restore violated relations, subject to room-boundary and footprint feasibility checks. Because these adjustments are local, they restore relations without disturbing the arrangement sampled in the proposal. Relations such as chairs facing their table decide whether a zone is functional and coherent, so restoring them directly supports visual quality. Figure 3 contrasts geometry-only compilation, which omits this relational step, with contract-guided compilation.

Scene completion. Finally, the VLM adds wall-mounted objects and small objects on furniture surfaces. For each wall or furniture top, it chooses object categories and suggests positions in that

surface's coordinates, and geometric placement maps these positions into the room while checking available space, boundaries, and intersections (Appendix A.4).

4 EXPERIMENTS

4.1 SCENE GENERATION

Implementation Details. We use Qwen3-VL-8B-Instruct (Bai et al., 2025) for functional zoning and scene compilation, and FLUX.1-dev (Labs, 2024) with FreeFuse (Liu et al., 2025) and asset-specific LoRA adapters for visual layout proposals. OVMono3D-LIFT (Yao et al., 2026) lifts 2D boxes from Grounding DINO (Liu et al., 2023) into 3D, with depth and camera intrinsics from Depth Pro (Bochkovskiy et al., 2025). Reference exemplars are drawn from M3DLayout (Zhang et al., 2026b). Appendix A details generation configurations, asset sources, and reference-data separation.

Baselines. We compare against three representative methods using their official codebases. Holodeck (Yang et al., 2024b) uses an LLM to specify spatial constraints for a layout solver. SceneWeaver (Yang et al., 2026) uses a self-reflective agent to coordinate generation and editing tools. SAGE (Xia et al., 2026) uses visual and physics critics to refine scenes and validate placements in simulation. We retain method-specific asset sources, use our Qwen3-VL-8B-Instruct backbone for Holodeck and SceneWeaver for a controlled comparison, and keep the model pair specified in SAGE's paper. Appendix A details these adaptations.

Metrics. We evaluate physical plausibility, visual quality, and real-layout alignment. Following SceneEval (Tam et al., 2026), we report mesh-level collision rates at the object (Coll. obj) and scene (Coll. scene) levels and mean out-of-bounds objects per scene (OOB). We also report the stable-object rate (Stab.) under REST3D (Ma et al., 2026). Following SceneWeaver and SAGE (Yang et al., 2026; Xia et al., 2026), GPT-4o (Hurst et al., 2024) scores realism (Real.), functionality (Func.), layout coherence (Coh.), and completeness (Compl.). For real-layout alignment, *Layout Distribution Discrepancy* (LDD) applies unbiased RBF maximum mean discrepancy (MMD^2) (Gretton et al., 2012) to features encoding object composition and spatial relations, against two references of real rooms: Matterport3D (Chang et al., 2017) and Scan2CAD (Avetisyan et al., 2019). Appendices B and C provide metric definitions, scoring protocols, hardware, and resource accounting.

4.1.1 MAIN RESULTS

We generate and evaluate 10 scenes for each of five room types: bedrooms, living rooms, dining rooms, bathrooms, and offices, using the template `a <room type>`.

Physical Plausibility. Our scenes are the only ones free of mesh collisions and out-of-bounds objects, and they are the most stable under REST3D (Table 1). The baselines add objects one at a time without resolving the scene as a whole, so conflicts accumulate as a scene grows: SAGE's agent inserts objects in 407 of its 504 tool calls across its 50 scenes but moves a placed object in only 44, and Holodeck spawns repeated copies of an asset at a single point. The instability also reaches the furniture itself: 31% of Holodeck's, 26% of SAGE's, and 13% of SceneWeaver's large objects, those at least 0.6 m across, move beyond the stability threshold, against 3% of ours. Our compiler instead resolves all furniture of a room jointly before completion and verifies each surface object against its support, so no conflict is left to accumulate. Additional stability evaluations appear in Appendix C.

Visual Quality. Our scenes score highest on all four visual-quality axes. SAGE is a strong baseline and places the most objects, but its placement loop keeps adding them until floors and surfaces are crowded (Figure 4), while our scenes score higher on functionality and completeness with fewer objects. Holodeck and SceneWeaver place furniture through textual constraints and edits, so functional relations are often lost, for example dining chairs that stand apart from the table and face away from it. Our proposals depict each functional group as a whole and the contract preserves its relations through compilation; the ablation in Figure 5, which fixes the object inventory, shows that arrangement alone raises all four visual-quality scores.

Table 1: **Comparison with scene generation methods.** Bold/underline mark best/second-best.

Method	#Obj	Physical Plausibility				Visual Quality ↑				LDD ↓		Time ↓
		Coll. ↓		OOB ↓	Stab. ↑	Real.	Func.	Coh.	Compl.	MP3D	Scan2CAD	(min)
		obj	scene									
Holodeck	18.0	32.7	78.0	.160	35.1	4.66	5.07	4.23	4.30	.2509	.2342	**1.6**
SceneWeaver	15.4	33.6	<u>60.0</u>	.133	<u>67.5</u>	5.30	5.50	4.51	4.71	**.0972**	<u>.1816</u>	142.6
SAGE	34.2	<u>12.7</u>	78.0	<u>.020</u>	45.1	<u>5.91</u>	<u>5.62</u>	<u>4.54</u>	<u>6.46</u>	.2191	.2251	73.6
SceneBloom (ours)	23.6	**0.0**	**0.0**	**.000**	**75.3**	**6.18**	**6.29**	**4.76**	**6.74**	<u>.1759</u>	**.1340**	<u>2.7</u>

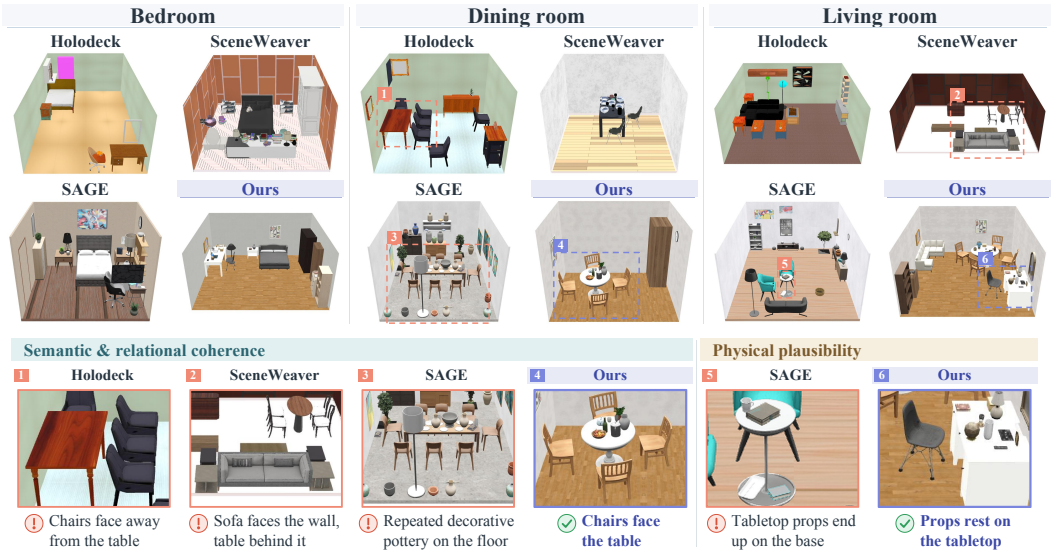

Figure 4: **Qualitative comparison.** Selected scenes and enlarged regions illustrate functional organization and physical placement across methods, each retaining its native asset source. Aggregate results are in Table 1.

Real-Layout Alignment. Our layouts are the closest to the Scan2CAD reference and the second closest to Matterport3D. Because LDD compares which objects a room contains and how they are arranged around its anchor furniture, a low distance on both references indicates that our rooms are organized like real rooms rather than fitted to a single dataset. SceneWeaver is closest to Matterport3D using its released PhyScene initializations, but this advantage does not carry over to Scan2CAD. Held-out real rooms score near zero against their own reference, far below every method, so these distances reflect genuine differences in layout organization (Appendix C.2).

Efficiency. Generation takes 2.7 minutes per layout, approximately 27× faster than SAGE on the same hardware, because each zone's arrangement is sampled once and compiled without an iterative placement loop. Holodeck is faster per layout, but 78% of its scenes contain collisions (Appendix C).

4.1.2 Ablation Study

To isolate visual sampling and contract guidance, we replace the visual proposal with a textual proposal, in which a VLM directly specifies the position and orientation of each object. Each proposal is compiled with and without the contract extracted from it (Table 11).

Visual proposals provide a stronger foundation for compilation, and contract guidance further improves them while preserving physical plausibility. Under both contract settings, visual proposals outperform textual proposals on every physical and visual-quality metric. Holding the visual proposal fixed, contract guidance additionally raises layout coherence from 4.31 to 4.65 and realism from 5.66 to 5.83, consistent with its role in restoring object poses and relations. These gains coincide with a reduction in contract violations from 37.5% to 4.2% (Appendix D), and the compiled

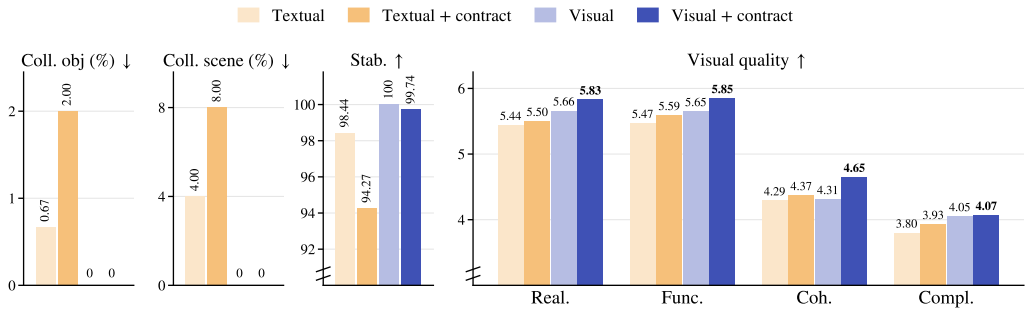

Figure 5: **Ablation of proposal source and contract guidance.** Results on 50 furniture-only scenes with shared inventories, assets, and sizes. Appendix D lists the exact values.

layouts remain collision-free. Visual sampling thus establishes layout quality, and contract guidance preserves the sampled relations through 3D compilation.

For textual proposals, contract guidance improves visual quality but increases collisions and reduces stability. These proposals are less plausible to begin with, as their objects are more crowded than in visual proposals and some already collide. Restoring relations within such arrangements can introduce further conflicts. Contract guidance therefore refines a plausible proposal but cannot substitute for one, which is why SceneBloom pairs it with visual sampling.

4.2 Object-Goal Navigation Learning

We then use our scenes to fine-tune an object-navigation policy, first comparing them with other training sources (Section 4.2.1) and then raising their difficulty with SceneBloom (Section 4.2.2).

Task and Evaluation. We study open-vocabulary object-goal navigation using OVSegDT (Zemskova et al., 2025), conditioned on RGB observations, language goals, target masks, and history. Every policy is fine-tuned from the released OVSegDT checkpoint (the pretrained policy) with PPO (Schulman et al., 2017) for 128,000 environment steps under an identical recipe. We evaluate on held-out scenes from our generator and an 864-episode screening subset of HM3D-OVON Val Unseen (Yokoyama et al., 2024; Ramakrishnan et al., 2021). We report success rate (SR) and success weighted by path length (SPL) (Anderson et al., 2018) at a 1.0 m threshold.

Training Episodes. We import scenes into Habitat (Savva et al., 2019) and construct episodes from visible goal viewpoints and reachable starts. Each generated training source provides 120 scenes, 24 for each of the five room types in Section 4.1; Appendix E gives further training details.

4.2.1 Training on Our Scenes Improves Generalization

We compare SceneBloom (ours), Randomized (the same scenes with floor-object poses resampled), and Agent-planned (text-agent layouts with zoning), alongside the pretrained policy without fine-tuning and a real-scan continuation that applies the same recipe to 142 HM3D training scans.

Connected Space for Navigation. Our layouts keep the walkable floor connected far more often than the randomized control, which holds the same objects. Only 0.9% of our layouts split the floor into separate regions, against 36.8% of the randomized layouts, excluding furniture tops that the agent cannot climb. A split floor confines every episode to one region, leaving less space to navigate and shorter paths to learn from; arranging furniture coherently keeps the whole room available for training. Appendix E compares the agent-planned layouts and the external generators.

Policy Fine-Tuning and Generalization. Fine-tuning on our scenes achieves the highest success rates on unseen goals in both held-out generated and real environments (Table 2). For goals used during fine-tuning, it raises SR from 73.67% to 74.60% and SPL from 61.31% to 64.00%, and Randomized and Agent-planned layouts yield similar gains. Further, on goal categories not used during fine-tuning, our scenes yield the highest SR and SPL, whereas every other fine-tuned policy

Table 2: **Policy fine-tuning and generalization.** SR and SPL (%) on 100 seen-goal and 107 unseen-goal held-out generated scenes and on the 36 scanned scenes of the screening subset. Fine-tuned rows average three training seeds. Bold/underline mark best/second-best.

Policy	Generated environments				Real environments	
	Seen goals		Unseen goals		Unseen goals	
	SR	SPL	SR	SPL	SR	SPL
Pretrained (no fine-tuning)	73.67	61.31	<u>52.57</u>	38.29	<u>48.26</u>	**21.18**
Fine-tuned on 142 real scans						
Real-scan continuation	73.02	61.12	51.59	<u>38.42</u>	47.57	20.68
Fine-tuned on 120 generated scenes						
Randomized	**74.88**	63.26	51.51	38.09	48.15	20.74
Agent-planned	74.39	<u>63.39</u>	51.63	37.65	47.99	20.72
SceneBloom (ours)	<u>74.60</u>	**64.00**	**53.19**	**38.89**	**48.57**	<u>21.17</u>

falls below the pretrained policy in SR. In real environments, where both the scenes and the goal categories are unseen, our scenes also outperform real-scan continuation, which only revisits the 142 HM3D scans seen during pretraining and falls below the pretrained policy. Our generator instead supplies new and varied rooms on demand, so 120 generated scenes improve generalization where 142 real scans do not.

4.2.2 Controllable Difficulty Improves Real-Environment Transfer

Matching the difficulty of real environments further improves transfer (Table 3). Real homes, especially multi-room ones, often hide the goal behind walls and furniture, whereas our rooms keep more goals in view. Because SceneBloom organizes each room into functional zones, it can introduce this difficulty directly. Using the same prompts as the 120 scenes in Table 2, we regenerate each scene with an additional interior zone of tall wardrobes (SceneBloom with occluders). As a control, Randomized with occluders keeps these occluders fixed and resamples the other floor objects, and all policies are fine-tuned with the identical recipe.

Table 3: **Real-environment transfer with occluders.** SR and SPL (%). Each middle row differs from the last row in one factor.

Policy	SR	SPL
Pretrained	48.26	<u>21.18</u>
Real-scan continuation	47.57	20.68
SceneBloom (ours)	<u>48.57</u>	21.17
Randomized with occluders	48.34	20.72
SceneBloom with occluders	**49.50**	**21.35**

Three comparisons explain this gain. First, the added difficulty itself helps, as success in real environments rises from 48.57% to 49.50%, while removing the occluders from the same regenerated scenes returns it to 48.42%. Second, the difficulty is effective when paired with a coherent layout, as randomizing the other floor objects around the same occluders yields only 48.34%. Third, this generated variation is more useful than further real data, as SceneBloom with occluders also surpasses real-scan continuation (47.57%) under the same training budget. Together, these results show how controllable generation supplies useful training variation by combining task-relevant difficulty with coherent spatial organization.

5 Conclusion

We presented SceneBloom, an agentic framework that separates the relationships to preserve in a layout from the poses to repair. Visual proposals supply realistic arrangements, and contract-guided compilation repairs their geometry while keeping the proposed relations, so our scenes are physically valid, visually coherent, and close to real room layouts. Sampling each zone once and resolving each room jointly also makes generation far faster than iterative agentic methods. These scenes in turn benefit embodied learning, as fine-tuning on them yields the highest object-navigation success on unseen goals in both generated and real environments, and using SceneBloom to match their difficulty to real rooms raises real-environment success further.

AI USE STATEMENT

Generative models are components of our method and evaluation. A vision–language model plans functional zones, extracts relation contracts, and places wall and surface objects; a text-to-image diffusion model samples the layout proposals; and GPT-4o scores visual quality (Sections 3 and 4). We did not use generative AI tools to develop theoretical models, formulate or prove mathematical claims, or propose hypotheses, and translation and qualitative data analysis do not apply to this work. Additionally, we used generative AI tools to create figures, refine the paper, and search for related literature. We have reviewed all AI-assisted work and verified every cited work against its source. We take full responsibility for the final content of this work.

ETHICS STATEMENT

This work generates synthetic indoor scenes and trains navigation policies in simulation. It involves no human subjects or personal data, and visual quality is scored by a vision–language model rather than by human raters. All datasets are used under their licenses, and the asset sources are listed in Appendix A.

REPRODUCIBILITY STATEMENT

Section 3 and Appendix A describe the generation pipeline, and Appendix G gives the zoning, contract-extraction, and judge prompts. Appendix B defines the metrics and evaluation protocols, and Appendix E specifies the training sources, budget, and seeds, and Section 4.2 lists the evaluation sets. We provide the code for the generation pipeline.

REFERENCES

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025.

Peter Anderson, Angel X. Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, and Amir Zamir. On evaluation of embodied navigation agents. *ArXiv*, abs/1807.06757, 2018. URL https://api.semanticscholar.org/CorpusID:49865049.

Armen Avetisyan, Manuel Dahnert, Angela Dai, Manolis Savva, Angel X Chang, and Matthias Nießner. Scan2cad: Learning cad model alignment in rgb-d scans. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2609–2618. IEEE, 2019.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Mikołaj Bińkowski, Dougal J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=r1lUOzWCW.

Alexey Bochkovskiy, Amaël Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan Richter, and Vladlen Koltun. Depth pro: Sharp monocular metric depth in less than a second. In *International Conference on Learning Representations*, volume 2025, pp. 75602–75637, 2025.

Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017.

Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.

Tianyuan Dai, Josiah Wong, Yunfan Jiang, Chen Wang, Cem Gokmen, Ruohan Zhang, Jiajun Wu, and Li Fei-Fei. Automated creation of digital cousins for robust policy learning. In *Conference on Robot Learning (CoRL)*, 2024.

Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Kiana Ehsani, Jordi Salvador, Winson Han, Eric Kolve, Aniruddha Kembhavi, and Roozbeh Mottaghi. procthor: Large-scale embodied ai using procedural generation. *Advances in neural information processing systems*, 35:5982–5994, 2022.

Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13142–13153, 2023.

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 *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10913–10922. IEEE, 2021a.

Huan Fu, Rongfei Jia, Lin Gao, Mingming Gong, Binqiang Zhao, Steve Maybank, and Dacheng Tao. 3d-future: 3d furniture shape with texture. *International Journal of Computer Vision*, 129 (12):3313–3337, 2021b.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773, 2012. URL `http://jmlr.org/papers/v13/gretton12a.html`.

Zeqi Gu, Yin Cui, Zhaoshuo Li, Fangyin Wei, Yunhao Ge, Jinwei Gu, Ming-Yu Liu, Abe Davis, and Yifan Ding. Artiscene: Language-driven artistic 3d scene generation through image intermediary. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 2891–2901, 2025.

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.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Zhao Jin, Zhengping Che, Tao Li, Zhen Zhao, Kun Wu, Yuheng Zhang, Yinuo Zhao, Zehui Liu, Qiang Zhang, Xiaozhu Ju, et al. Artvip: Articulated digital assets of visual realism, modular interaction, and physical fidelity for robot learning. In *International Conference on Learning Representations*, volume 2026, pp. 75272–75301, 2026.

Mukul Khanna, Yongsen Mao, Hanxiao Jiang, Sanjay Haresh, Brennan Shacklett, Dhruv Batra, Alexander Clegg, Eric Undersander, Angel X Chang, and Manolis Savva. Habitat synthetic scenes dataset (hssd-200): An analysis of 3d scene scale and realism tradeoffs for objectgoal navigation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16384–16393. IEEE, 2024.

Black Forest Labs. Flux. `https://github.com/black-forest-labs/flux`, 2024.

Minzhang Li, Kuixiang Shao, Xuebing Li, Yuyang Jiao, Yinuo Bai, Hengan Zhou, Sixian Shen, Jiayuan Gu, and Jingyi Yu. Spread: Spatial-physical reasoning via geometry aware diffusion. *arXiv preprint arXiv:2603.27573*, 2026.

Chenguo Lin and Yadong Mu. Instructscene: Instruction-driven 3d indoor scene synthesis with semantic graph prior. In *International Conference on Learning Representations*, volume 2024, pp. 25687–25718, 2024.

Lu Ling, Chen-Hsuan Lin, Tsung-Yi Lin, Yifan Ding, Yu Zeng, Yichen Sheng, Yunhao Ge, Ming-Yu Liu, Aniket Bera, and Max Li. Scenethesis: A language and vision agentic framework for 3d scene generation. In *International Conference on Learning Representations*, volume 2026, pp. 136596–136629, 2026.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.

Yaoli Liu, Yao-Xiang Ding, and Kun Zhou. Freefuse: Multi-subject lora fusion via adaptive token-level routing at test time. *arXiv preprint arXiv:2510.23515*, 2025.

Zhen Luo, Yixuan Yang, Xudong XU, Jinkun Hao, Zhaoyang Lyu, Feng Zheng, Jiangmiao Pang, and Yanwei Fu. STABLE: Simulation-ready tabletop layout generation via a semantics–physics dual system. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=pGpEGZhL7Y.

Xiaoxuan Ma, Jiashun Wang, Nicolas Ugrinovic, Yehonathan Litman, and Kris Kitani. Rest3d: Reconstructing physically stable 3d scenes from a single image. *arXiv preprint arXiv:2605.30338*, 2026.

Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac gym: High performance GPU based physics simulation for robot learning. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL https://openreview.net/forum?id=fgFBtYgJQX_.

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots. In *Robotics: Science and Systems (RSS)*, 2024.

Soroush Nasiriany, Sepehr Nasiriany, Abhiram Maddukuri, and Yuke Zhu. Robocasa365: A large-scale simulation framework for training and benchmarking generalist robots. In *International Conference on Learning Representations (ICLR)*, 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.

Ryan Po, Guandao Yang, Kfir Aberman, and Gordon Wetzstein. Orthogonal adaptation for modular customization of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7964–7973, June 2024.

Alexander Raistrick, Lingjie Mei, Karhan Kayan, David Yan, Yiming Zuo, Beining Han, Hongyu Wen, Meenal Parakh, Stamatis Alexandropoulos, Lahav Lipson, et al. Infinigen indoors: Photorealistic indoor scenes using procedural generation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21783–21794. IEEE, 2024.

Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John M Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. Habitat-matterport 3d dataset (HM3d): 1000 large-scale 3d environments for embodied AI. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL https://openreview.net/forum?id=-v4OuqNs5P.

Xingjian Ran, Xiaoye Mo, Sihao Liu, Jianyu Zhang, Li Luo, and Bo Dai. Scenemosaic: Efficient and diverse simulation-ready scene generation via hybrid agentic layout evolution. *arXiv preprint arXiv:2609.05594*, 2026.

Nadun Ranawaka, Josiah Wong, Wei-Lin Pai, Wei-Teng Chu, Tianyuan Dai, Masoud Moghani, Hang Yin, Yunfan Jiang, Wesley Durbano, Brandon Huynh, et al. Simfoundry: Modular and automated scene generation for policy learning and evaluation. *arXiv preprint arXiv:2606.28276*, 2026.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22500–22510. IEEE, 2023.

Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9338–9346. IEEE, 2019.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Hou In Ivan Tam, Hou In Derek Pun, Austin T Wang, Angel X Chang, and Manolis Savva. Sceneeval: Evaluating semantic coherence in text-conditioned 3d indoor scene synthesis. In *2026 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 7355–7365. IEEE, 2026.

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*, 2024.

Song Tang, Kaiyong Zhao, Yuliang Li, Qingsong Yan, Penglei Sun, Junyi Zou, Qiang Wang, and Xiaowen Chu. Spatialgrammar: A domain-specific language for llm-based 3d indoor scene generation. *arXiv preprint arXiv:2604.27555*, 2026.

Ao Wang, Lihao Liu, Hui Chen, Zijia Lin, Jungong Han, and Guiguang Ding. Yoloe: Real-time seeing anything. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 24591–24602. IEEE, 2025.

Yian Wang, Xiaowen Qiu, Jiageng Liu, Zhehuan Chen, Jiting Cai, Yufei Wang, Tsun-Hsuan Wang, Zhou Xian, and Chuang Gan. Architect: Generating vivid and interactive 3d scenes with hierarchical 2d inpainting. *Advances in Neural Information Processing Systems*, 37:67575–67603, 2024a.

Yufei Wang, Zhou Xian, Feng Chen, Tsun-Hsuan Wang, Yian Wang, Katerina Fragkiadaki, Zackory Erickson, David Held, and Chuang Gan. Robogen: Towards unleashing infinite data for automated robot learning via generative simulation. *arXiv preprint arXiv:2311.01455*, 2023.

Zhaowei Wang, Hongming Zhang, Tianqing Fang, Ye Tian, Yue Yang, Kaixin Ma, Xiaoman Pan, Yangqiu Song, and Dong Yu. Divscene: Towards open-vocabulary object navigation with large vision language models in diverse scenes. *arXiv preprint arXiv:2410.02730*, 2024b.

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. *arXiv preprint arXiv:2602.10116*, 2026.

Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21469–21480. IEEE, 2025.

Yandan Yang, Baoxiong Jia, Peiyuan Zhi, and Siyuan Huang. Physcene: Physically interactable 3d scene synthesis for embodied ai. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16262–16272. IEEE, 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.

Yue Yang, Fan-Yun Sun, Luca Weihs, Eli VanderBilt, Alvaro Herrasti, Winson Han, Jiajun Wu, Nick Haber, Ranjay Krishna, Lingjie Liu, Chris Callison-Burch, Mark Yatskar, Aniruddha Kembhavi, and Christopher Clark. 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, June 2024b.

Jin Yao, Hao Gu, Xuweiyi Chen, Jiayun Wang, and Zezhou Cheng. Open vocabulary monocular 3d object detection. In *2026 International Conference on 3D Vision (3DV)*, pp. 1801–1811. IEEE, 2026.

Naoki Yokoyama, Ram Ramrakhya, Abhishek Das, Dhruv Batra, and Sehoon Ha. Hm3d-ovon: A dataset and benchmark for open-vocabulary object goal navigation. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5543–5550. IEEE, 2024.

Tatiana Zemskova, Aleksei Staroverov, Dmitry Yudin, and Aleksandr Panov. Ovsegdt: segmenting transformer for open-vocabulary object goal navigation. *arXiv preprint arXiv:2508.11479*, 2025.

Meisheng Zhang, Shizhao Sun, Yang Zhao, Ziyuan Liu, Zhijun Gao, and Jiang Bian. Orchestrating spatial semantics via a zone-graph paradigm for intricate indoor scene generation. In *Forty-third International Conference on Machine Learning*, 2026a. URL https://openreview.net/forum?id=Z3cAYsB72W.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, et al. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.

Yiheng Zhang, Zhuojiang Cai, Mingdao Wang, Meitong Guo, Tianxiao Li, Li Lin, and Yuwang Wang. M3dlayout: A multi-source dataset of 3d indoor layouts and structured descriptions for 3d generation. In *CVPR*, 2026b.

A Implementation Details

A.1 Baseline Configurations

We evaluate Holodeck (Yang et al., 2024b), SceneWeaver (Yang et al., 2026), and SAGE (Xia et al., 2026) from their released codebases, retaining their native asset sources and scene-generation procedures. SceneWeaver's release supplies scene initializations as a bank of precomputed samples covering bedrooms, living rooms, and dining rooms. We therefore run SceneWeaver on exactly the three room types its release supports, giving 30 scenes, and Holodeck and SAGE on all five, giving 50 scenes each. Prompts, solver settings, and asset sources are unchanged; Holodeck and SceneWeaver run on Qwen3-VL-8B-Instruct, the backbone of our pipeline, and SAGE keeps the model pair specified in its paper, GPT-OSS-120B (Agarwal et al., 2025) with Qwen3-VL-30B-A3B-Instruct (Table 9); the tool set follows the release's own instruction to enable the tools one can supply. Runtime is measured over each method's executable released pipeline, and Table 7 states the included stages and hardware.

Asset library. Our asset library holds 192 assets: 75 floor-furniture meshes across 22 categories, 109 surface objects, and 8 wall objects. We generate most of them using the TRELLIS (Xiang et al., 2025) text-to-3D model. The 27 articulated floor meshes, which cover eight storage and sanitary categories, come from the ArtVIP dataset (Jin et al., 2026) instead. Our library is also extensible: once a category-compatible mesh has a trained LoRA adapter, the visual sampler can draw it, and the VLM, the compiler, and the prompts stay unchanged. Each baseline keeps its own asset source. Holodeck retrieves from Objaverse (Deitke et al., 2023), SAGE generates its assets with TRELLIS text-to-3D, and SceneWeaver's released pipeline draws 3D-FUTURE (Fu et al., 2021b) furniture through its released initializations, with procedural small objects from Infinigen (Raistrick et al., 2024). Each of these systems treats its asset pipeline as part of its contribution, so substituting our meshes would evaluate a different system.

A.2 Zone-Wise Visual Proposals

For each room, retrieval restricts M3DLayout (Zhang et al., 2026b) exemplars to the requested room type and ranks them by cosine similarity between Qwen3-Embedding (Zhang et al., 2025) encodings of the room description and of each exemplar's structured text. The top-ranked exemplar is kept as an anchor, and two more are drawn from the remaining candidates by similarity-weighted sampling without replacement. This varies the structural references while keeping the nearest neighbor.

The retrieval corpus is built from two disjoint indices. The first holds 5,173 3D-FRONT (Fu et al., 2021a) rooms and covers bedrooms, living rooms, and dining rooms. The second holds 627 Matterport3D regions and covers bathrooms, offices, kitchens, lounges, recreation rooms, and meeting rooms; it contains no bedroom, living-room, or dining-room region. The real-layout references of Section B.1 are Matterport3D bedrooms, living rooms, and dining rooms, which neither index contains, and Scan2CAD rooms, which come from ScanNet and are absent from both. No room that enters either reference distribution can therefore be retrieved as an exemplar, and the conditioning set and the evaluation references are disjoint by construction.

Every floor-furniture mesh the sampler can draw has its own LoRA adapter, trained on multi-view renders of that mesh with DreamBooth and prior preservation (Ruiz et al., 2023) and orthogonal adaptation (Po et al., 2024); wall and surface objects are placed geometrically after compilation and need no adapter. The pipeline inserts each adapter's trigger token before the category word it stands for, so the word sofa in a zone prompt yields the particular sofa selected from our library. A zone uses at most five adapters, one per furniture category, and FreeFuse (Liu et al., 2025) runs them together in a single generation, binding each adapter to the tokens that name its object and to the image region those tokens attend to. Every furniture object compiled from the image is therefore the library mesh whose adapter drew it. This grounding motivates the adapters: it reduces the appearance mismatch between a proposal and the objects used for construction, so the compiler rebuilds the sampled arrangement from the objects the image shows rather than from meshes that merely share their categories.

A.3 CONTRACT-GUIDED SCENE COMPILATION

Before any geometry is fitted, the VLM reads each zone's original proposal image once. The VLM returns a fixed contract $\mathcal{C}_j = \{(u, r, v, \gamma)\}$ holding only the relations the proposal itself shows, where u is a reconstructed object and v is a reconstructed object or a wall, r is one of the relations in Table 4, and γ is the VLM's confidence. Relations at $\gamma \geq 0.75$ are binding during compilation. The prompt and output schema are given in Appendix G.

Table 4: Relations a zone contract can hold. The VLM selects relations; the compiler checks them geometrically and performs all pose changes.

Relation	Meaning	Permitted refinement
`face`	Subject faces the target.	Yaw, or moving in front of it.
`align_with`	Subject is axis-aligned with the target.	Yaw.
`beside`	Subject stands at the target's side.	Yaw, or moving to its side.
`move_near`	Subject stays within reach of the target.	Moving and turning toward it.
`back_to`	Subject's back is against its zone's wall.	Yaw, or moving to the wall.

The compiler then works outward. It first resolves each zone on its own, separating that zone's objects around its anchor under the guidance of the zone's contract, and records the zone's internal pairwise distances. It next places the zones within the room plan, moving every zone rigidly, and the recorded distances confirm that no zone is deformed. Finally, it checks each binding relation and applies the rotation or translation that restores each violated one.

A.4 WALL AND SURFACE OBJECT PLACEMENT

Wall objects are placed first and surface objects second, and both stages follow the same propose-then-verify pattern. The VLM chooses object categories, each realized by a seeded library mesh, and proposes positions in normalized coordinates of the target surface, along and up a wall or across a furniture top. Geometry converts these positions into room coordinates and rejects placements that collide with objects already placed. The two conversions follow.

For a wall with unit tangent $\mathbf{t}$, inward normal $\mathbf{n}$, and line point $\mathbf{p}_0$ at along-wall coordinate 0, let $[a_0, a_1]$ be the admissible along-wall coordinates of the object's center within its zone. A wall object of depth d with normalized proposal $(u, v) \in [-1, 1]^2$ is mapped to

$$a = a_0 + \frac{u+1}{2}(a_1 - a_0), \qquad \mathbf{p}_{xz} = \mathbf{p}_0 + a\mathbf{t} + \frac{d}{2}\mathbf{n}, \tag{2}$$

while v selects a valid vertical position between the floor and ceiling. The object's yaw faces the wall normal. If the proposal refers to supporting furniture, its horizontal coordinate is aligned to the furniture center and its base height is placed above the furniture top. Candidates are rejected when their wall rectangles overlap or their 3D boxes intersect furniture.

For a support surface centered at $\mathbf{c} = (c_x, c_z)$ with half extents (h_x, h_z), yaw θ, and top height y_{top}, the VLM proposes (u, v, ψ), where $(u, v) \in [-1, 1]^2$. The object's world pose is

$$\mathbf{p}_{xz} = \mathbf{c} + R(\theta) \begin{bmatrix} uh_x \\ vh_z \end{bmatrix}, \qquad y = y_{\text{top}} + \delta, \qquad \theta_{\text{obj}} = \theta + \psi, \tag{3}$$

where δ is a small clearance. We retain a placement only when its oriented footprint lies inside the inset support rectangle and does not overlap an accepted object.

B EVALUATION PROTOCOLS AND METRIC DEFINITIONS

B.1 LAYOUT DISTRIBUTION DISCREPANCY

Layout Distribution Discrepancy (LDD) measures how closely generated layouts follow the distribution of real layouts. We use two independently collected references of real scanned rooms. Matterport3D (Chang et al., 2017) is read through the M3DLayout annotations, which express each

room's furniture as oriented boxes, and Scan2CAD (Avetisyan et al., 2019) aligns CAD models to ScanNet (Dai et al., 2017) scans. For each room type, every annotated room of a reference is pooled into one empirical distribution with no model fitted to it (Section C.2 lists the counts).

We map generated and reference objects to a shared coarse taxonomy. For each room type, we define an anchor category a and associated categories $\mathcal{A}$: bed with storage and table in bedrooms, table with chairs in dining rooms, and table with sofas and chairs in living rooms. Each associated object is assigned to its nearest anchor in the ground plane.

For anchor instance j and category c, let $\Delta_{jck} = \mathbf{p}_{ck} - \mathbf{p}_{a_j}$, $r_{jck} = \|\Delta_{jck}\|_2$, and $\theta_{jck} = \operatorname{atan2}(\Delta^z_{jck}, \Delta^x_{jck})$. We summarize angular and radial structure by

$$m_{jcq} = \frac{1}{|I_{jc}|} \sum_{k \in I_{jc}} e^{\mathrm{i} q \theta_{jck}}, \quad q = 1, \dots, 4, \tag{4}$$

$$b_{jct} = \frac{1}{n_{jc}} \sum_{k} \exp\left[-\frac{1}{2} \left(\frac{\log(1 + r_{jck}) - \log(1 + t)}{0.30} \right)^2 \right], \quad t \in \mathcal{T}, \tag{5}$$

where n_{jc} is the number of assigned objects, $I_{jc} = \{k : r_{jck} > \epsilon\}$ with $\epsilon = 10^{-8}$ m, and $\mathcal{T} = \{0, 0.5, 1, 1.5, 2, 3, 4\}$ meters. The category descriptor is

$$\begin{aligned} h_{jc} = \big[& \mathbf{1}[n_{jc} > 0], \log(1 + n_{jc}), \tfrac{1}{n_{jc}} \textstyle\sum_k \mathbf{1}[r_{jck} \le \epsilon], \\ & \{|m_{jcq}|\}_{q=1}^4, \{b_{jct}\}_{t \in \mathcal{T}}, \log(1 + \operatorname{median}_k r_{jck}) \big]. \end{aligned} \tag{6}$$

Empty sets yield zeros. We also encode pairwise angular organization between associated categories $c < c'$ as

$$h_{jcc'} = \left\{ \frac{1}{|I_{jc}|\,|I_{jc'}|} \sum_{k \in I_{jc}} \sum_{l \in I_{jc'}} \cos(q(\theta_{jck} - \theta_{jc'l})) \right\}_{q=1}^{4}. \tag{7}$$

The room descriptor concatenates anchor presence, the log anchor count, global log counts of associated categories, and the anchor-averaged descriptors:

$$\phi(x) = \left[\mathbf{1}[N_a > 0], \log(1 + N_a), \{\log(1 + N_c)\}_{c \in \mathcal{A}}, \frac{1}{\max(N_a, 1)} \sum_j \big[\{h_{jc}\}_{c \in \mathcal{A}}, \{h_{jcc'}\}_{c < c'} \big] \right]. \tag{8}$$

The descriptor has four properties. First, every spatial term is measured relative to a functional anchor, so it records how a room organizes itself around the object that gives the room its purpose. Second, only the magnitudes $|m_{jcq}|$ enter, so a room and a rotated copy of it receive identical features. Third, counts enter through $\log(1 + n)$ and radii through soft bins, which compresses the range of counts and distances across room sizes. Fourth, an absent category is represented by its presence flag with the remaining entries set to zero, so a missing functional group still contributes to the distance. The descriptor uses object categories and positions only, so it is unaffected by object size and orientation and never sees asset identity, texture, or mesh, which keeps LDD independent of the asset-coverage differences discussed in Section A.

Each reference defines the normalization for every method. Its means and standard deviations, the latter floored at 0.1, standardize each feature dimension, and the kernel bandwidth is derived from the same reference, so all four methods are scored on one scale and none is measured against statistics taken from its own output.

For generated layouts $G = \{g_i\}_{i=1}^n$ and reference layouts $R = \{r_j\}_{j=1}^m$, with $x_i = \phi(g_i)$ and $y_j = \phi(r_j)$, LDD is the unbiased maximum mean discrepancy (Gretton et al., 2012):

$$\mathrm{LDD}(G, R) = \widehat{\mathrm{MMD}}^2_{k_{\mathrm{RBF}}}(X, Y), \tag{9}$$

$$k_{\mathrm{RBF}}(x, y) = \exp\left(-\frac{\|x - y\|_2^2}{2\sigma^2} \right), \tag{10}$$

$$\widehat{\mathrm{MMD}}^2(X, Y) = \frac{\sum_{i \neq i'} k(x_i, x_{i'})}{n(n-1)} + \frac{\sum_{j \neq j'} k(y_j, y_{j'})}{m(m-1)} - \frac{2 \sum_{i,j} k(x_i, y_j)}{nm}. \tag{11}$$

The RBF kernel is characteristic, so a zero population MMD implies equality of the represented distributions. LDD is therefore sensitive to differences in typical layouts and to differences in higher-order distribution shape. We set σ to the median pairwise distance among standardized reference features and average room-level scores equally across bedrooms, dining rooms, and living rooms. Lower is closer, and the unbiased finite-sample estimate may be slightly negative. Each method contributes ten scenes per room type.

Kernel Inception Distance (KID) is itself an unbiased MMD estimator with a cubic polynomial kernel applied to Inception image features (Bińkowski et al., 2018). Our diagnostic Layout-KID keeps the estimator and kernel,

$$k_{\mathrm{cubic}}(x, y) = \left(1 + \frac{x^\top y}{d}\right)^3, \tag{12}$$

where d is the feature dimension, but replaces Inception features with the same structured layout representation. It therefore offers a familiar moment-based sensitivity check. LDD stays the primary measure because the RBF kernel tests equality of the represented layout distributions more directly. Multi-scale LDD averages RBF kernels with bandwidths 0.5σ, σ, and 2σ to check that conclusions do not depend on one bandwidth.

B.2 Physical Plausibility

We follow SceneEval (Tam et al., 2026) for method-independent geometric evaluation. Object collision is the percentage of objects participating in at least one mesh intersection, computed per scene and averaged over scenes, and scene collision is the percentage of scenes containing any such object; contact tolerance is zero. OOB is the mean number of objects per scene whose footprint falls outside the room under SceneEval's ray-cast test with a 1% tolerance.

Dynamic stability follows REST3D (Ma et al., 2026). We replay every nonempty exported scene in Isaac Gym (Makoviychuk et al., 2021) at $1/60$ seconds, using six position iterations, one velocity iteration, and a maximum depenetration velocity of $5\,\mathrm{m/s}$. Scenes settle for 120 steps. An object is stable when its final translation is at most $0.1\,\mathrm{m}$ and its rotation is at most $0.1\,\mathrm{rad}$. Stab. pools all movable objects; position drift and peak linear and angular velocities provide severity measures. Every method uses the same simulator, timestep, thresholds, coordinate conversion, and evaluation code.

B.3 Visual Quality

We use GPT-4o-2024-08-06 to score realism, functionality, layout coherence, and completeness on a 0–10 scale, following the four axes used by SceneWeaver (Yang et al., 2026) and SAGE (Xia et al., 2026). The judge receives a top-down render and a text list of object positions, orientations, and sizes. We use temperature zero and average five judgments per scene, and all four methods are scored in one run. The evaluation interface and score bands are given in Appendix G.

C Additional Scene Generation Results

C.1 Physical Stability

Table 5 complements the stable-object rate (Stab.) in the main paper with continuous motion severity. Our scenes have the lowest mean position drift and the lowest peak linear and angular velocities under the shared dynamic replay.

C.2 Real-Layout Distribution Alignment

References. Matterport3D contributes, through its M3DLayout annotations, 236 bedroom, 62 dining-room, and 76 living-room regions. Scan2CAD (Avetisyan et al., 2019) aligns CAD models to 1,506 ScanNet (Dai et al., 2017) scans of 704 distinct spaces. Scan2CAD does not distribute room types, so we type its rooms with the anchor rules of Section B, which gives 275 bedrooms, 495 dining rooms, and 287 living rooms from 126, 275, and 152 spaces; applied to Matterport3D, the

Table 5: Physical stability under the REST3D protocol (Ma et al., 2026). Stab. is the percentage of movable objects within the translation and rotation thresholds; Movable obj./scene counts the movable objects replayed, and the last three columns measure motion severity. Bold indicates the best point estimate. All four methods are replayed with the same solver settings.

Method	Scenes	Movable obj./scene	Stab. ↑	Pos. Drift (m) ↓	Lin. Vel. (m/s) ↓	Ang. Vel. (rad/s) ↓
Holodeck	50	15.0	35.1%	0.7923	2.573	16.727
SceneWeaver	30	14.8	67.5%	0.2075	0.944	12.708
SAGE	50	30.6	45.1%	0.9757	2.470	11.015
SceneBloom (ours)	50	21.8	**75.3%**	**0.0996**	**0.811**	**4.724**

same rule leaves the ordering of methods unchanged. Scan2CAD's ShapeNet (Chang et al., 2015) categories are mapped to our taxonomy, and the mapping is checked against object footprints.

Calibration. A real-versus-real control sets the zero point of each reference: ten real rooms from held-out buildings or spaces, scored against the rest of their reference, give a median LDD of -0.001 on Matterport3D and 0.014 on Scan2CAD over 200 draws, far below every generated LDD.

Results. Table 6 adds two kernels to the Matterport3D LDD of Table 1. Under all three, our layouts are closer to Matterport3D than SAGE's and Holodeck's, and in a bootstrap that resamples reference buildings and generated scenes, our LDD is below SAGE's and Holodeck's in 96.5% and 99.0% of replicates, respectively. SceneWeaver, which starts from released PhyScene initializations, is closest to Matterport3D but not to Scan2CAD, where our layouts are the closest of all four methods.

Table 6: **Real-layout distribution alignment on two references.** Lower is closer to the reference. LDD is the metric of Table 1; multi-scale LDD and Layout-KID replace its kernel as defined in Section B.1. Bold/underline mark best/second-best.

Reference	Metric	Holodeck	SceneWeaver	SAGE	SceneBloom (ours)
Matterport3D	LDD	.2509	**.0972**	.2191	<u>.1759</u>
	Multi-scale LDD	.2082	**.0983**	.1948	<u>.1617</u>
	Layout-KID	5.0997	**1.1484**	5.3881	<u>2.0391</u>
Scan2CAD	LDD	.2342	<u>.1816</u>	.2251	**.1340**

Separation from the retrieval corpus. Retrieved exemplars neither overlap with the evaluation references nor reappear in our layouts. None of the thirty scored scenes retrieved a Matterport3D region. In the feature space of LDD, a scored scene lies on average 6.69 from the three exemplars it retrieved and 6.96 from the other exemplars retrieved for the same room type, only slightly closer to the layouts it was conditioned on. Appendix D examines how retrieval shapes zoning and scene quality.

C.3 Cost and Efficiency

Our pipeline generates a layout in 2.7 minutes, about 27 times faster than SAGE (Table 7). Holodeck is faster still, but places fewer objects and leaves most of its scenes with collisions. Table 8 breaks down our time. Visual proposal generation takes the largest share (45.4%), followed by functional zoning (27.6%), while lifting with contract-guided compilation (14.5%) and wall and surface completion (12.4%) together take under a minute.

In language-model usage (Table 9), our pipeline spends 34,867 tokens and $0.0057 per scene, 8 times fewer tokens than SAGE and 60 times fewer than SceneWeaver; only Holodeck uses fewer. SAGE's runtime is dominated by its placement loop: it spends 54.2 of 61.3 total generation hours in `place_objects_in_room`, which accounts for 407 of 504 tool calls. Of 392 placement calls with recorded outcomes, 75 place every requested object, 221 partially succeed, and 96 place none; only 44 calls perform corrective moves. This insertion-heavy trajectory explains its high object density and long runtime.

Table 7: **Generation wall clock per scene.** Measured from each run's own artifacts, one scene at a time, over the baseline pools of Table 1 and a 50-scene run of our pipeline. The span is layout generation for every system: rendering and physical settling are excluded. Each job generates one scene on one H100-class GPU, except SceneWeaver, which runs on 8 CPU cores and no GPU: its released code ships the diffusion stage as precomputed samples rather than running inference, so its timing covers only the CPU-bound agent stage. SAGE's clock is insensitive to the accelerator (A100 65.2, H100 76.0, H200 75.7 min mean) and is dominated by its agent loop instead.

Method	n	#Obj	Mean (min)	Median (min)
Holodeck (Yang et al., 2024b)	50	18.0	1.6	1.5
SceneWeaver (Yang et al., 2026)	30	15.4	142.6	129.9
SAGE (Xia et al., 2026)	50	34.2	73.6	77.9
SceneBloom (ours)	50	23.6	2.7	2.6

Table 8: **Our layout-generation time breakdown.** Mean wall clock over a 50-scene production run with the same scene requests as Table 1. Components differ from the total by 0.2 seconds due to uninstrumented orchestration and rounding.

Stage	Mean (s)	Share
Functional zoning	45.3	27.6%
Visual proposal generation	74.6	45.4%
Lifting and contract-guided compilation	23.8	14.5%
Wall and surface completion	20.3	12.4%
Layout generation	164.2	100.0%

D Ablation Details

D.1 Retrieval

Retrieval improves how the VLM plans each room and, through it, the quality of the final scene (Table 10). We rerun the pipeline from retrieval onward with no exemplar supplied, which changes only the VLM's input. Without exemplars, the VLM plans fewer functional zones in four of the five room types, 3.14 against 3.94 on average, most sharply in bathrooms, from 4.40 to 1.80, and in dining rooms, from 2.90 to 1.90. The resulting scenes score lower on all four visual-quality axes, most in functionality, 5.25 against 6.28, and completeness, 5.87 against 6.76, and lie farther from both real references, especially Scan2CAD, $.2171$ against $.1340$. Retrieval thus informs how the VLM decomposes each room into functional zones, and the rest of the pipeline builds on that decomposition.

D.2 Proposal Source and Contract-Guided Compilation

Design. Both proposal sources receive the same prompt, zoning plan, and asset set, and differ only in whether a VLM states each object's placement (textual) or the diffusion proposal of our pipeline supplies it (visual).

Results. Table 11 lists the values plotted in Figure 5 together with the contract violation rate.

E Object-Goal Navigation Training and Evaluation

E.1 Training Scene Sources

SceneBloom (ours) uses the visual-proposal and contract-guided compiler pipeline of Section A.2 and Section A.3. *Randomized* starts from those same scenes and keeps every object and mesh, then resamples each floor object's position uniformly within the room and its yaw uniformly in $[0, 2\pi)$, rejecting samples whose footprint leaves the room or overlaps a floor object already placed. Surface

Table 9: **Language-model usage and cost per scene.** SAGE keeps the model pair specified in its paper; the other methods use Qwen3-VL-8B-Instruct for a controlled comparison.

Method	LLM backbone	n	#Obj	Tokens/scene	USD/scene
Holodeck	Qwen3-VL-8B-Instruct	50	18.0	6,986	$0.0016
SceneWeaver	Qwen3-VL-8B-Instruct	30	15.4	2,088,872	$0.3104
SAGE	GPT-OSS-120B + Qwen3-VL-30B-A3B-Instruct	50	34.2	293,499	$0.0151
SceneBloom (ours)	Qwen3-VL-8B-Instruct	50	23.6	34,867	$0.0057

Table 10: **Retrieval ablation.** Results on 50 scenes, ten per room type; LDD uses the 30 bedroom, dining-room, and living-room scenes. Zones is the mean number of functional zones per scene, and the visual quality of both rows is judged in a single run, so the with-retrieval row rescores the scenes of Table 1 and differs from it by at most 0.02. Bold marks the better value.

		Visual Quality ↑				LDD ↓	
Retrieval	Zones	Real.	Func.	Coh.	Compl.	MP3D	Scan2CAD
✔	3.94	**6.19**	**6.28**	**4.75**	**6.76**	**.1759**	**.1340**
✘	3.14	5.89	5.25	4.63	5.87	.1857	.2171

objects move rigidly with the furniture that supports them and wall objects do not move, so this source removes the arrangement while holding the object content fixed. *Agent-planned* replaces the whole placement stage with a single text-only call to Qwen3-VL-8B-Instruct at temperature zero. The call receives the same room box, the labels and extents of the same functional zones, and the room's asset library with nominal dimensions, and returns a textual layout that gives every object's position and yaw and, for wall and surface objects, the wall or support it belongs to.

Each source is given 120 generated scenes and the same 128,000-step PPO optimization, with the state-encoder learning rate scaled by 0.1, and each is trained with three seeds. Real-scan continuation applies the same optimization and seeds to the 142 HM3D training scans.

E.2 Perception Conditions

Fine-tuning on generated scenes uses ground-truth target masks, and every evaluation uses masks predicted by YOLOE (Wang et al., 2025), so the deployed perception interface is the predicted mask.

E.3 Navigable Space in the Training Scenes

All pools are measured with one agent geometry (radius 0.17 m, height 1.41 m, maximum climb 0.1 m, cell height 0.05 m). A scene is fragmented when its navigation mesh has more than one walkable island, and the floor-only rate (floor splits) sets aside raised islands, which we attribute to the furniture beneath them by sampling each island's height. Walk is the mean over scenes of the largest island's share of the walkable area. Our layouts are the least fragmented and have the highest Walk of all six pools (Table 12).

Fragmentation carries into training because episode construction admits a fragmented scene: it requires a finite geodesic between start and goal, so each episode stays on one island. For the randomized control, 41.1% of training episodes come from fragmented scenes, against 2.0% for ours, and these episodes are shorter, with a median geodesic length of 1.94 m against 2.27 m in its connected scenes.

F Qualitative Results

Figure 6 extends the main comparison to offices and bathrooms.

Table 11: **Ablation of proposal source and contract guidance.** Values plotted in Figure 5, on 50 furniture-only scenes with shared inventories, assets, and sizes. Viol. is the percentage of relations in the extracted contract that the compiled layout violates. Bold/underline mark best/second-best.

		Physical Plausibility			Visual Quality				Relations
Proposal	Contract	Coll. obj ↓	Coll. scene ↓	Stab. ↑	Real. ↑	Func. ↑	Coh. ↑	Compl. ↑	Viol. ↓
Textual	✘	0.67	4.00	98.44	5.44	5.47	4.29	3.80	39.07
Textual	✔	2.00	8.00	94.27	5.50	5.59	<u>4.37</u>	3.93	<u>10.75</u>
Visual	✘	**0.00**	**0.00**	**100.00**	<u>5.66</u>	<u>5.65</u>	4.31	<u>4.05</u>	37.45
Visual	✔	**0.00**	**0.00**	<u>99.74</u>	**5.83**	**5.85**	**4.65**	**4.07**	**4.25**

Table 12: **Navigation-mesh connectivity.** n counts the scenes with a navigation mesh. Our layouts and the two controls are measured on their training pools, the baselines on their pools of Table 1.

Scene pool	n	Fragmented ↓	Floor splits ↓	Walk ↑
Holodeck	50	16.0%	2.0%	0.9840
SAGE	45	20.0%	11.1%	0.9467
SceneWeaver	30	43.3%	23.3%	0.8389
SceneBloom (ours)	115	**2.6%**	**0.9%**	**0.9968**
Randomized	95	37.9%	36.8%	0.8823
Agent-planned	106	14.2%	14.2%	0.9663

G Prompt Templates

This appendix gives, in condensed form, the prompts of three model calls: functional zoning (Section G.1) and relation-contract extraction (Section G.2) in the generation pipeline, and the visual-quality judge used only for evaluation (Section G.3). The pipeline also calls the VLM to choose wall and surface objects and propose their positions (Section A.4). Bracketed fields are filled for each call with the inputs named in the box.

G.1 Functional Zoning

Functional zoning: system message

You are a furniture-layout expert specializing in indoor scene generation. You receive up to three retrieved indoor floor plans as spatial-layout references. Each reference provides a top-down render when available and a record of its room type, object count, and, for every object, category, ground-plane center and size in meters, height, and yaw. Use the references only as spatial-layout priors and produce a JSON zone plan for the room requested by the user. The zones must be non-overlapping axis-aligned rectangles that tile the room floor.

Return only valid JSON with this structure:

```
{"room_type": STRING, "room_width_m": FLOAT, "room_depth_m":
FLOAT, "zones": [{"label": STRING, "polygon_2d":
[[x1,z1],...,[x4,z4]], "allowed_categories": [STRING],
"category_counts": {STRING: INT}, "anchor_category": STRING or
null, "max_objects": INT, "priority": INT, "diffusion_prompt":
STRING, "wall_orientation": north, south, east, west, or null}]}
```

Apply the following rules:

1. Express all polygons in the local XZ frame, inside $[0, W] \times [0, D]$, where W and D are the room width and depth; zones must cover the floor without overlap or gaps.
2. Infer realistic room dimensions from the references and allocate enough area for the furniture assigned to each zone.



Figure 6: **Additional qualitative comparisons on offices and bathrooms.** We show Holodeck, SAGE, and our method on the two room types not covered by our SceneWeaver evaluation. SceneWeaver is omitted because its released PhyScene initializations provide the bedroom, dining-room, and living-room categories used in our executable evaluation, but not office or bathroom scenes. Each method retains its native asset source.

3. Use only the floor-furniture categories listed in the prompt, such as bed, sofa, desk, and dining_table. Wall-mounted, hanging, built-in, and surface objects are added by later stages.
4. Give each allowed category an integer count, by default 1, and use more only when the room type naturally has multiples. A circulation zone that absorbs leftover floor has no categories, no anchor, and `max_objects` of 0.
5. If present, `anchor_category` must be one of the zone's allowed categories.
6. Write the zone prompt (`diffusion_prompt`) in at most 25 words. Refer to each planned item by its exact category name with no adjective before it, state relative relations between items, and mention no unplanned object or absolute room position.
7. Set wall orientation from the zone boundary: north at $z = D$, south at $z = 0$, east at $x = W$, and west at $x = 0$. For a corner zone, use the wall the anchor backs onto; use null for an interior or empty zone.

Functional zoning: user message

The VLM receives, for each retrieved reference, its top-down render when available, its similarity score, and `[REFERENCE_JSON]`; then the available asset catalog, the original room instruction, the room type, and the required categories, followed by: "Produce the JSON zone plan for this room." When a zone's prompt and category counts disagree, a note listing each mismatch is appended and the VLM returns the complete plan again, at most twice.

G.2 Relation-Contract Extraction

Relation-contract extraction: system message

You are reading the original diffusion-generated image of one room zone. The accompanying JSON lists the reconstructed objects and the identifiers that must be used. List the qualitative spatial relations shown in the image, whether or not the current 3D reconstruction satisfies them, and report only relations that are clearly visible. Do not output coordinates or numeric

yaw, and do not invent identifiers. Use only the five relations of Table 4; `face` may be used only for subjects whose front is unambiguous, and `back_to` takes only the wall behind the zone as its target. Confidence expresses how clearly the image supports the relation: 0.9–1.0 is unmistakable, 0.7–0.85 is clear, and 0.4–0.6 is plausible but ambiguous.

Return at most 10 relations as JSON:

```
{"zone_assessment": {"function": STRING}, "constraints":
[{"subject_id": ID, "operator": RELATION, "target": {"type":
"object", "id": ID} or {"type": "wall"}, "confidence": FLOAT,
"reason": STRING}]}
```

Relation-contract extraction: user message

The VLM receives the original zone image, followed by the list of objects in the zone, `[RECONSTRUCTED_OBJECT_JSON]`.

G.3 Visual-Quality Evaluation

Visual-quality evaluation: user message

You are an expert interior-design reviewer scoring a computer-generated 3D `[ROOM_TYPE]` scene produced for `[ROOM_INSTRUCTION]`. You are shown a top-down render and a text list of every object's position, facing, and size, `[OBJECT_LIST]`. Score strictly: if a critical issue exists, including a missing key object, an obvious layout error, an unrealistic element, or a clearly wrong size, lower the score substantially even when other aspects are fine.

Score each criterion independently on an integer 0–10 scale, where 10 fully meets or exceeds a professional designer's expectation and needs no major fix, 5 partially meets it with obvious flaws, and 0 completely fails. Use the full range rather than defaulting to its midpoint. For every criterion use the same calibration bands:

- 8–10: the criterion is fully or substantially met; any deviations are minor;
- 4–7: the criterion is partially met; notable deficiencies are present;
- 0–3: the criterion is not met; deficiencies are severe.

The four criteria are:

1. **Realism**: how realistic and lived-in the room looks, ignoring texture, lighting, and doors.
2. **Functionality**: how well the room supports its intended activities, such as sleeping or working.
3. **Layout coherence**: whether furniture is arranged logically, in a good pose, and well sized.
4. **Completeness**: how complete and finished the room feels.

Reason briefly for each criterion, then output only:

```
{"realism": {"score": INT, "justification": STRING},
"functionality": {...}, "layout_coherence": {...}, "completeness":
{...}}
```