Scenix: Sparse-View 3D Scene Reconstruction via Executable Scene Programs

Anonymous submission

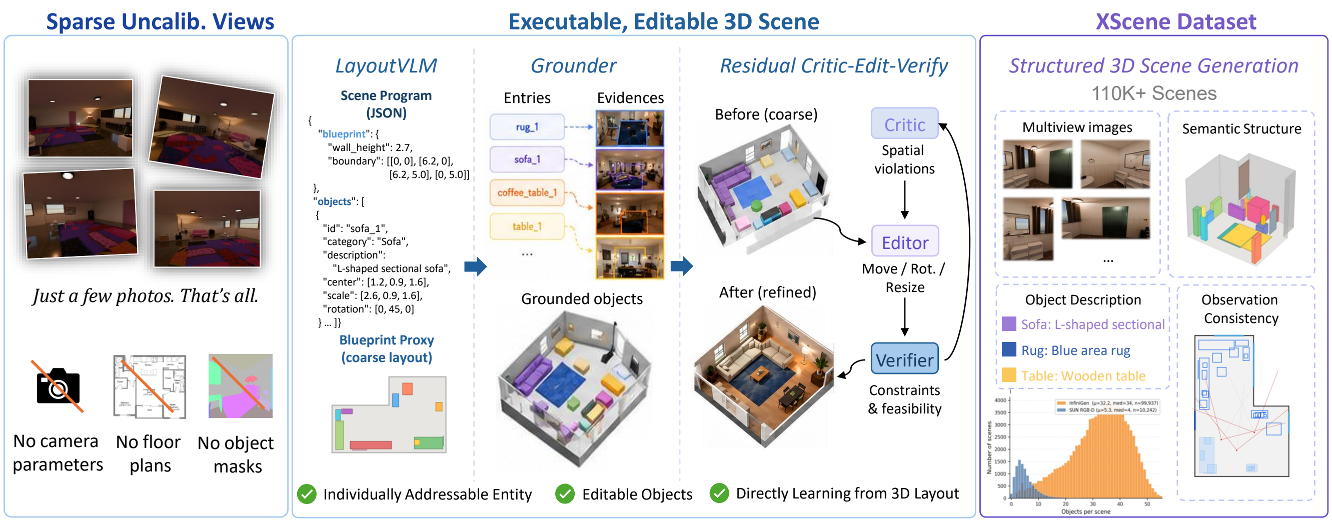


Figure 1: SCENIX directly transforms sparse uncalibrated RGB observations into an executable 3D scene program and realizes it as a structured, editable indoor environment.

Abstract

Synthesizing a structured and editable 3D indoor scene from a few uncalibrated RGB views requires more than generating high-quality individual assets: a system must infer the room structure, associate objects across incomplete observations, and recover a globally consistent spatial configuration. Previous methods mainly focus on 3D scene generation with text input or require continuous visual inputs with additional priors, e.g., human-annotated masks or accurate 3D layouts, which makes these methods labor demanding and hard to apply in general cases. We present SCENIX, a sparse-view 3D scene reconstruction framework via executable scene programs, a structured representation that can be directly instantiated into editable 3D scenes. Given sparse views, SCENIX predicts executable scene programs through perception-grounded asset instantiation and closed-loop spatial refinement. To support this task, we construct XSCENE, a dataset of approximately 110,000 synthetic and real indoor scenes with multiview imagery, room structures, object-centric descriptions, and metric spatial annotations. We further introduce observation-consistent supervision that aligns each target scene with the visual evidence available in its input views. Experiments on held-out XSCENE scenes, real indoor images, and out-of-distribution SpatialGen cases evaluate structured scene prediction, object grounding, and spatial refinement.

This is an anonymized submission for review purposes only. Distribution, citation, or public sharing of this manuscript is strictly prohibited. Copyright and publication details will appear in the final version if accepted.

1 Introduction

Recent generative models create high-quality 3D assets from a single image (Hu et al. 2026), yet reconstructing a complete, editable indoor scene from a few uncalibrated RGB views remains harder. Beyond recovering objects, a system must infer room structure, establish correspondences across sparse observations, and recover a globally consistent spatial configuration. These tightly coupled structural and semantic reasoning problems make sparse-view scene reconstruction fundamentally more difficult than single-object generation.

Existing paradigms address only part of this problem. Text-conditioned methods (Chen et al. 2024; Wan et al. 2026; Feng et al. 2023; Schneider and Dai 2026) generate plausible indoor scenes but cannot faithfully reconstruct a particular environment. Neural reconstruction methods recover scene geometry from images, yet typically require dense calibrated views or auxiliary geometric priors and produce represen-

Table 1: Comparison of input modalities and auxiliary conditions. $*$ The VLM framework supports text, image, and video inputs.

Method	Single	Multi.	Text	Mask	Conditions
Gen3DSR	✓	×	×	✓	Depth+Camera
3DFixer	✓	×	×	✓	Depth
SAM3D	✓	×	×	✓	Diff. Rendering
SpatialGen	✓	✓	×	×	3D Layout
NaLA	×	×	✓	×	N/A
LayoutGPT	×	×	✓	×	N/A
Ours	✓	✓	$*$	×	N/A

Table 2: Comparison of 3D datasets. BP: blueprint; P. & E.: positions and extents. †3D-FRONT images are rendered from provided scenes.

Dataset	Src.	#Scenes	#Imgs	BP	P. & E.	Ori.	Desc.
SUN RGB-D	real	–	10.3K	✓	✓	×	×
ScanNet	real	1,513	2.5M	×	✓	×	×
Matterport3D	real	90	10.8K	×	✓	×	×
ScanNet++	real	1,006	11.1M	×	✓	×	×
Structured3D	syn.	3,500	196.5K	✓	✓	×	×
Hypersim	syn.	461	77.4K	×	✓	×	×
SpatialGen	syn.	12,328	4.7M	✓	✓	×	×
3D-Front†	syn.	18,968	–	✓	✓	✓	×
XScene(Ours)	mix	110K	560K	✓	✓	✓	✓

tations that are difficult to edit at the object level. Recent open-vocabulary asset generators (Ye et al. 2025; Chen et al. 2026; Hunyuan3D et al. 2025; Xiang et al. 2025) have largely removed the restriction of predefined asset libraries. The remaining challenge is no longer asset creation, but *recovering a structured scene description* that guides *how those assets are instantiated and composed*.

This observation suggests that scene understanding should precede object realization. We therefore formulate sparse-view 3D scene reconstruction as executable scene program prediction. Rather than reconstructing monolithic geometry, it explicitly encodes the room envelope together with object identities, descriptions, positions, extents, orientations, and support relations. As an interface between perception and realization, it can be learned from images, instantiated with generated assets, verified through canonical projections, and edited at the object level.

Based on this formulation, Scenix reconstructs structured 3D scenes from uncalibrated RGB observations without external camera calibration, point clouds, floor plans, or manually annotated masks. LayoutVLM performs multimodal reasoning over a variable number of views to predict a scene program encoding room layout, object-level spatial attributes, and rich semantic descriptions. Asset Grounder realizes its entities with open-vocabulary 3D assets, followed by bounded Critic–Editor–Verify refinement that preserves structural consistency.

Training executable scene programs requires supervision unavailable in existing indoor datasets. We therefore construct the XScene dataset, comprising approximately 110,000 synthetic and real indoor scenes with room layouts, object descriptions, and metric spatial annotations. The synthetic portion is procedurally generated with InfiniGen (Raistrick et al. 2023) to provide scalable multiview supervision, while approximately 10,000 SUN RGB-D (Song, Lichtenberg, and Xiao 2015) images improve real-world coverage. To faithfully model sparse-view reconstruction, we further introduce an observation-consistent annotation protocol that retains only entities visible in the selected observations. Tab. 2 compares XScene with existing indoor datasets (Song, Lichtenberg, and Xiao 2015; Dai et al. 2017; Chang et al. 2017; Yeshwanth et al. 2023; Zheng et al. 2020; Roberts et al. 2021; Fang et al. 2026; Fu et al. 2021).

Extensive experiments on held-out XScene scenes, comparisons with single-view reconstruction baselines, and zero-shot transfer to SpatialGen demonstrate the effectiveness of our formulation. Scenix achieves more accurate scene layout reconstruction than existing approaches and consistently benefits from grounded residual refinement under distribution shift. Our contributions are threefold:

- We introduce Scenix, a new paradigm for sparse-view 3D scene reconstruction that directly infers executable structured scene programs from uncalibrated RGB images.
- We construct XScene, approximately 110,000 indoor cases with appearance-rich descriptions, complete placement parameters, and observation-consistent supervision.
- We demonstrate that executable scene programs serve as an effective representation for sparse-view reconstruction, yielding more accurate scene layouts, stronger generalization, and object-level editable 3D scenes.

2 Related Work

Structured indoor scene synthesis. SceneFormer, ATISS, and DiffuScene learn generative priors over object configurations (Wang, Yeshwanth, and Nießner 2021; Paschalidou et al. 2021; Tang et al. 2024), while LayoutGPT, Holodeck, AnyHome, and NaLA broaden control through language, relations, or foundation-model agents (Feng et al. 2023; Yang et al. 2024; Fu et al. 2024; Wan et al. 2026). Geometry-conditioned systems operate from proxy boxes or supplied layouts (Schult et al. 2024; Fang et al. 2026), and SceneWeaver and SAGE coordinate generators with semantic, visual, or physical evaluation (Yang et al. 2025; Xia et al. 2026). These methods synthesize from specifications rather than recover the identity, appearance, and metric placement of instances across sparse views.

Observation-conditioned scene recovery. Architectural methods recover room envelopes from perspective or panoramic RGB and polygonal floor plans from point clouds (Zou et al. 2018; Sun et al. 2019; Pintore, Agus, and Gobbetti 2020; Yue et al. 2023). Object-centric systems estimate cuboids and meshes, align CAD assets, or combine open-world perception with pose optimization (Nie et al. 2020; Gümeli, Dai, and Nießner 2022; Wu et al. 2025), but remain constrained by asset coverage and occlusion. Generative recovery reduces library dependence but

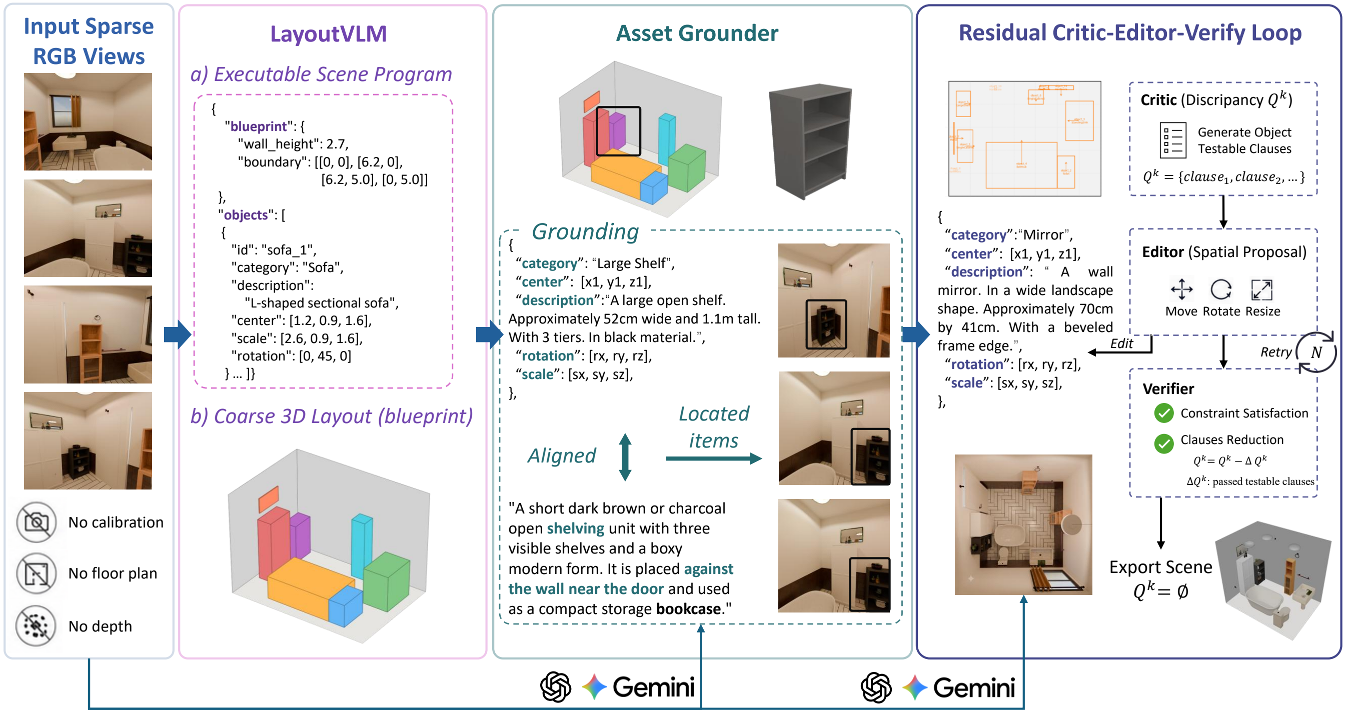


Figure 2: Overview of Scenix. LayoutVLM predicts an executable scene program from sparse RGB observations; Asset Grounder associates observed instances with scene entities and conditions open-vocabulary asset generation, and these visual evidences together with texts will be used to generate 3D assets; Residual Critic–Editor–Verify Loop performs closed-loop refinement under deterministic geometric constraints, and finally exports the full scene when the testable clause set is empty.

still requires masks, estimated geometry, de-occlusion, or separate pose stages (Meng et al. 2026; Yin et al. 2026; Shi et al. 2026); SceneScript decodes structured commands from walkthrough video (Avetisyan et al. 2024). Sparse-view radiance fields assume calibrated cameras (Niemeyer et al. 2022; Yang, Pavone, and Wang 2023); DUSt3R and MASt3R align pointmaps and dense correspondences (Wang et al. 2024; Leroy, Cabon, and Revaud 2024), while Spann3R, CUT3R, and VGGT predict cameras, depth, pointmaps, and tracks through spatial memory, recurrence, or feed-forward inference (Wang and Agapito 2025; Wang et al. 2025b,a). These approaches recover geometry rather than named, independently regenerable entities with editable relations. We instead predict an object-addressable scene program preserving category, appearance, and cross-view identity from sparse uncalibrated views.

3 Method

3.1 Problem Formulation

Given sparse RGB observations $\mathcal{I} = \{I_v\}_{v=1}^{V}$ of an indoor environment, our goal is to recover a structured and editable 3D scene without externally provided camera calibration, depth, point clouds, floor plans, or object masks. Conventional object-centric pipelines decompose this task into detection, geometric lifting, cross-view association, and optimization; we instead formulate reconstruction as *direct conditional generation of a 3D scene program*.

We represent a scene as $\mathcal{S} = (\mathcal{B}, \mathcal{O}, \mathcal{R})$. LayoutVLM first predicts an initial program $\widehat{\mathcal{S}}_0 = (\mathcal{B}, \widehat{\mathcal{O}}_0)$, where $\mathcal{B}$ denotes the induced blueprint used by subsequent stages; visual evidence then grounds the object set and relations. The blueprint $\mathcal{B}$ specifies polygonal room boundaries and wall parameters, while $\mathcal{O} = \{o_i\}_{i=1}^{N}$ contains individually addressable entities:

$$o_i = (c_i, d_i, \mathbf{t}_i, \mathbf{s}_i, \mathbf{r}_i, u_i), \quad (1)$$

where c_i and d_i denote category and appearance, $\mathbf{t}_i \in \mathbb{R}^3$ and $\mathbf{s}_i \in \mathbb{R}^3_+$ are the 3D center and extent, $\mathbf{r}_i$ is the orientation, and u_i records floor or object support. The relation set $\mathcal{R}$ captures object–object and object–room constraints. This representation is predictive and executable: it can be rendered as a canonical spatial proxy, edited explicitly, and populated with independently generated assets.

Figure 2 summarizes the pipeline. LayoutVLM predicts an initial scene program from all views in one step; Asset Grounder matches its entities to visual evidence and generates assets; the residual loop refines the spatial arrangement.

3.2 Direct 3D Scene Program Induction

Vision-to-structure generation. LayoutVLM shifts the task from staged geometric estimation to direct structured generation. Given a variable number of views in a single multimodal context, it autoregressively emits a serialized scene program rather than constructing per-view detections and lifting them into 3D. It jointly predicts room structure, object identities, semantic descriptions, and metric transformations:

$$\widehat{\mathcal{S}}_0 = \arg\max_{\mathcal{S}_0} p_\theta(\mathcal{S}_0 \mid I_1, \ldots, I_V) . \tag{2}$$

Here, p_θ is LayoutVLM's serialized-program distribution with parameters θ. The constrained schema separates blueprint and object records, each coupling semantic and appearance information with metric placement. It therefore expresses decisions normally made in separate coordinate systems within one compositional output space, while allowing weakly overlapping or non-overlapping views to contribute complementary evidence without explicit correspondence.

Observation-consistent learning. A sampled view subset rarely observes every entity, so supervising hidden objects would conflate reconstruction with unconstrained completion. We form $\mathcal{O}^{\mathrm{obs}}$ by retaining an object when the number of its oriented-box corners visible across the selected views reaches a fixed threshold κ; a corner is visible only if it lies inside a camera frustum and its ground-plane sight line does not intersect a blueprint wall. Camera parameters are used only to precompute targets and are unavailable at inference. For $\mathcal{S}_0^{\mathrm{obs}} = (\mathcal{B}^{\mathrm{gt}}, \mathcal{O}^{\mathrm{obs}})$ and $\mathbf{y}^{\mathrm{obs}} = \mathrm{Ser}(\mathcal{S}_0^{\mathrm{obs}})$, LayoutVLM minimizes

$$\mathcal{L}_{\mathrm{ind}} = -\sum_{t=1}^{T} \log p_\theta\big(y_t^{\mathrm{obs}} \mid y_{<t}^{\mathrm{obs}}, \mathcal{I}\big) . \tag{3}$$

Here, $\mathcal{B}^{\mathrm{gt}}$ is the annotated blueprint, Ser deterministic serialization, and T the target length; only assistant tokens are supervised, with user and system tokens masked. Targets are generated offline from random view subsets.

3.3 Asset Grounder

Direct induction provides a global spatial hypothesis, but ambiguous or out-of-distribution observations may still cause omissions, duplicates, or category errors. We therefore separate two authorities: the scene program supplies the room frame and initial poses, while visual evidence determines which instances exist and how they appear.

We construct an evidence-backed multiview inventory $\mathcal{E} = \{e_j\}_{j=1}^{M}$. Joint and per-view reasoning resolve cross-view identity, while a count audit handles repeated instances. Each item records category, appearance, support type, confidence, and evidence views.

An MLLM reconciles $\mathcal{E}$ with $\widehat{\mathcal{O}}_0$ using semantic and coarse spatial cues. Each seed receives one retention, replacement, or suppression decision, whereas cloning is reserved for an unmatched inventory item. These operations remove unsupported seeds, correct categories while retaining useful poses, and restore missing repeated instances without expanding $\mathcal{E}$. The grounded set $\widehat{\mathcal{O}}_g = \{o_i^g\}_{i=1}^{M}$ satisfies:

$$\begin{aligned} \widehat{\mathcal{O}}_g &= \mathrm{Recon}\Big(\widehat{\mathcal{O}}_0, \mathcal{E}\Big) , \\ |\widehat{\mathcal{O}}_g| &= M, \qquad \pi : \mathcal{E} \to \widehat{\mathcal{O}}_g \text{ is bijective}, \\ \mathrm{fam}(e_j) &= \mathrm{fam}(\pi(e_j)), \quad \forall e_j \in \mathcal{E}. \end{aligned} \tag{4}$$

Here, Recon is the reconciliation operator, π maps inventory items to active entities, and fam denotes semantic family. The bijection maps every inventory item to exactly one active entity. Reconciliation predicts no new poses; an image-guided planner localizes clones before boundary and collision checks validate all poses.

The grounded program determines *where* each object belongs, while observations determine *how* it appears. A promptable segmenter proposes regions and a VLM selects the best soft mask. The mask and description condition a clean single-object reference image, with text-only generation used when no mask is reliable. An image-conditioned 3D generator converts this reference into a mesh, while the program extent controls fitting during assembly.

3.4 Residual Critic–Editor–Verify Loop

Grounding and asset insertion may leave residual position, scale, and facing errors due to uncertain placement, metric size, or local front conventions. We therefore use a bounded Critic–Editor–Verify loop to refine, rather than reconstruct, the grounded layout.

Starting from $\mathcal{S}^0 = \widehat{\mathcal{S}}_g$, we render a metric bird's-eye proxy $\mathcal{P}(\mathcal{S}^k)$ with footprints and facing directions. An image generator produces an auxiliary top-down hypothesis $\mathcal{H}(\mathcal{I})$. Because $\mathcal{H}$ is synthesized, it supplies only coarse position and wall cues; the metric program and high-confidence relations remain authoritative.

At round k, the Critic converts visible discrepancies into object-specific clauses $\mathcal{Q}^k$, the Editor proposes a minimal translation, rotation, or footprint change, and the Verifier judges every clause on a fresh rendering:

$$\begin{aligned} \mathcal{Q}^k &= \mathrm{Critic}\big(\mathcal{P}(\mathcal{S}^k), \mathcal{H}(\mathcal{I}), \mathcal{R}\big) , \\ \mathcal{S}^{k+1} &= \mathrm{Edit}(\mathcal{S}^k, \mathcal{Q}^k), \\ \mathbf{a}^{k+1} &= \mathrm{Verify}\big(\mathcal{P}(\mathcal{S}^{k+1}), \mathcal{H}(\mathcal{I}), \mathcal{Q}^k, \mathcal{R}\big) . \end{aligned} \tag{5}$$

Each entry of $\mathbf{a}^{k+1}$ is a pass/fail verdict. Failed clauses and verifier feedback guide the next edit, while a deterministic footprint test introduces blueprint violations as mandatory constraints. Each edit is checked against the scene schema before application. The loop terminates when all clauses pass and all footprints lie within the blueprint, when no valid edit is applied, or when its fixed iteration budget is exhausted.

4 The XScene Dataset

Training LayoutVLM requires paired RGB observations from different perspectives and structured 3D scene programs. Each target must encode a room blueprint together with each object's long-form description, metric position, extent, and orientation. Since existing indoor datasets lack this complete supervision, we construct the Image–3D Scene Layout dataset (XScene) by procedurally generating randomized multiview scenes with InfiniGen and supplementing SUN RGB-D with the missing object annotations (Fig. 3).

4.1 Data Construction

Procedural multiview data. Using fixed InfiniGen seeds, we generate diverse indoor scenes across room types, such as kitchens, bathrooms, dining rooms, bedrooms, and living

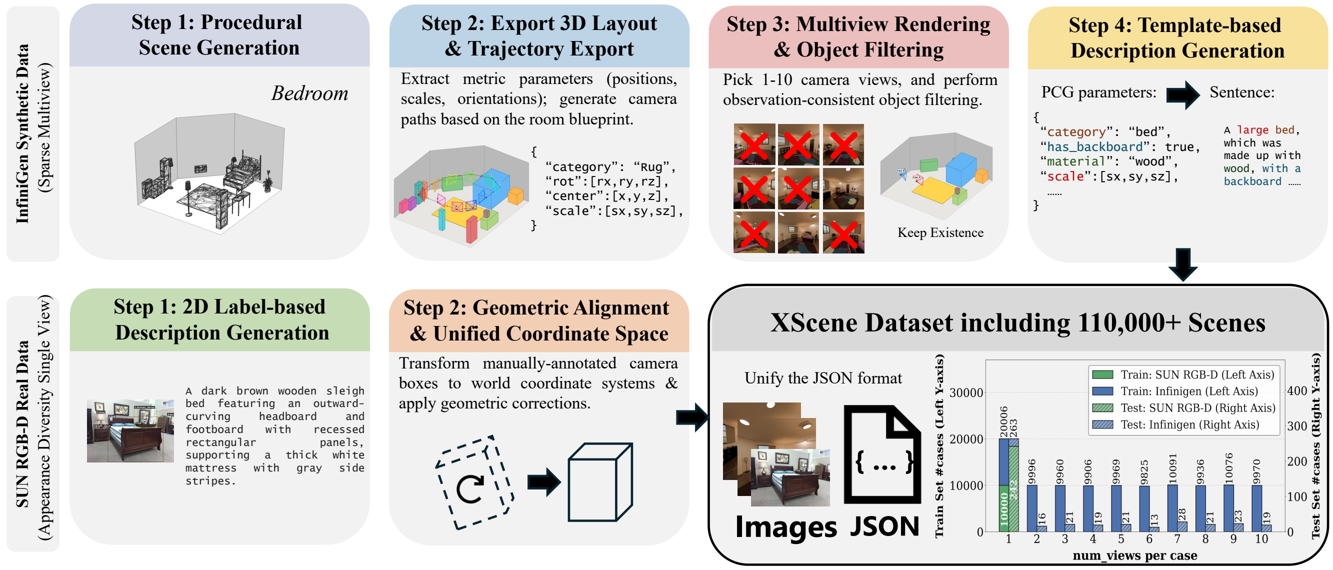


Figure 3: Construction of XScene. Synthetic and real-image sources provide complementary appearance and structural diversity, while cross-view visibility aggregation aligns each supervision target with the evidence available in its selected observations.

rooms, and export their room envelopes and object categories, positions, extents, and orientations. For each room, a blueprint-aware camera trajectory renders ten RGB images, from which we randomly retain one to ten views. Frustum intersection and wall occlusion retain only objects supported by the selected observations. InfiniGen's 79 procedural object generators expose readable parameters, which category-specific templates convert into long-form appearance descriptions.

Real-image data. Although these generators can theoretically yield unlimited assets, fixed sentence templates and parameterized property controls often produce structurally similar objects that underrepresent real-world diversity. We therefore add SUN RGB-D as single-view real-image supervision and use its 2D labels to prompt Hunyuan-turbo to describe the corresponding objects, attaching the text to their original 3D annotations. We transform the 3D boxes from camera coordinates to a world frame with non-negative x, y, and z coordinates. Because these boxes are manually annotated and may contain pose errors, we geometrically correct cases such as floor-supported objects whose boxes do not align with the ground plane. Finally, we convert the processed SUN RGB-D and InfiniGen subsets to a unified scene-program schema.

4.2 Dataset Statistics

We split XScene at the case level so that no room appears in both training and test. Table in 3 reports 110,179 cases and 560,184 images in total; InfiniGen provides scale and dense multiview supervision ($\sim$91% of training cases), while SUN RGB-D contributes real-world imagery and human-verified 3D annotations. InfiniGen cases are approximately uniformly distributed across one to ten views by construction, whereas SUN RGB-D occupies the single-view bin, together spanning the full spectrum from sparse to dense observations.

5 Experiments

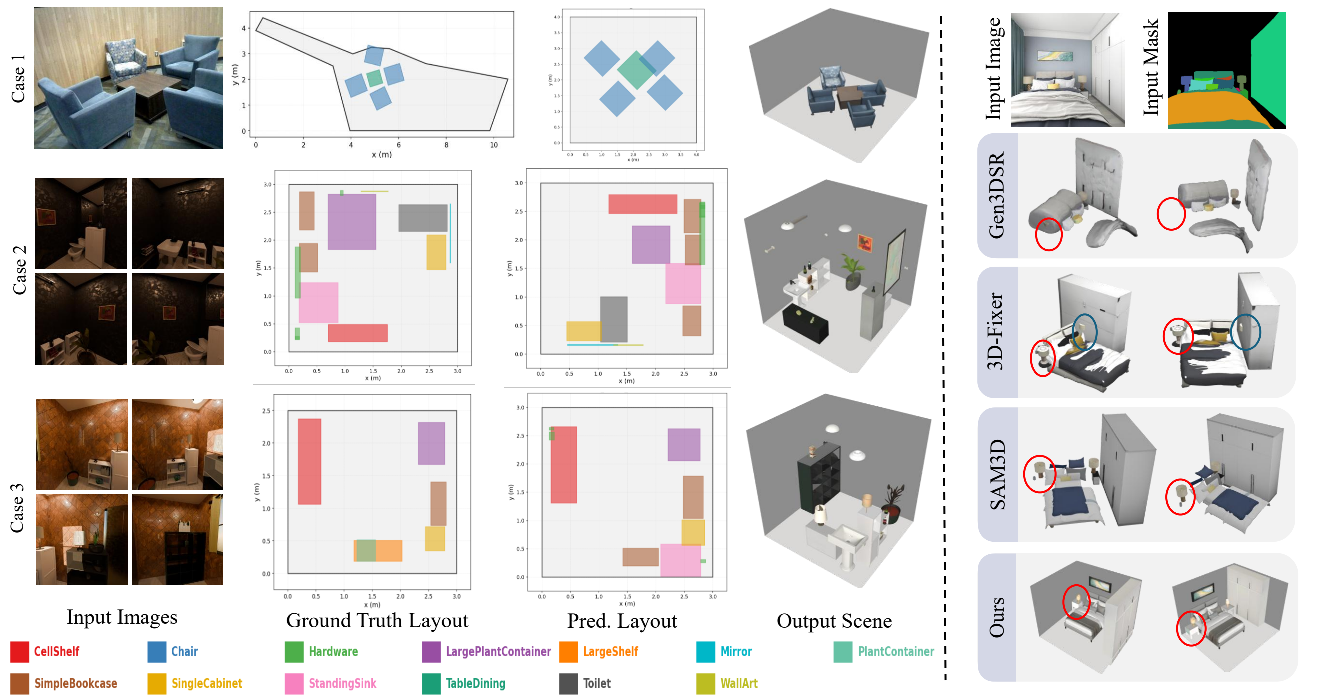
5.1 Experimental Setup

We train Qwen3.5-4B and Qwen3.5-9B LayoutVLMs by supervised fine-tuning. The XScene test split contains 202 InfiniGen cases with one to ten views and 242 single-view SUN RGB-D cases. For OOD evaluation, we subsample three to ten of each SpatialGen scene's 100 continuous views (Fang et al. 2026); a single-view subset supports comparisons with SAM-3D-object (SAM3D) (Chen et al. 2026), Gen3DSR (Ardelean, Özer, and Egger 2025), and 3D-Fixer (Yin et al. 2026).

5.2 Evaluation Metrics

Predictions and ground truth use independent coordinate frames, so we remove global translation and rotation before distance evaluation.

We report four complementary metric families. **(1) Localization F1.** We greedily match predicted and ground-truth objects of the same class by normalized center distance, counting a true positive below τ. Precision, Recall, and F1 are reported at $\tau \in \{5\%, 10\%\}$ of the room diagonal using macro scene averages and micro pooled counts. **(2) Class-only Metrics.** Ignoring position, we measure class-count overlap with $\mathrm{TP} = \sum_c \min(n_c^{\mathrm{pred}}, n_c^{\mathrm{gt}})$. Its gap from localization F1 separates recognizing *what* belongs in a room from placing it *where* it belongs. **(3) 3D IoU.** We compute oriented-box IoU for matched pairs and report its mean over matches, its mean over all ground-truth instances with misses counted as zero, and F1@{0.25,0.5}. Class-aware matching requires equal classes; class-agnostic matching provides a localization upper bound. **(4) Orientation Error (ICP-Rot).** Symmetry-aware yaw error folds the residual angle by each



(a) Our results on the XScene test set

(b) Single-view results on SpatialGen

Figure 4: Qualitative results of SCENIX across in-domain, real-image, and out-of-distribution scenes.

footprint's rotational symmetry (180° for rectangular and 90° for near-square footprints), avoiding penalties for equivalent orientations.

Unless noted otherwise, a fixed class-pattern filter excludes small decorative objects such as lights, tableware, books, wall ornaments, and hardware, which add disproportionate noise without reflecting room-scale layout errors.

5.3 Structured Scene Prediction

Table 3 reports raw, pre-refinement predictions on the two in-domain splits and zero-shot transfer to 48 OOD SpatialGen scenes. Results are macro-averaged over five runs and reported as mean $\pm$ population standard deviation where available.

In-domain behavior. SUN RGB-D produces substantially higher localization F1 and matched 3D IoU than InfiniGen, plausibly because its single-view scenes are smaller and less cluttered under normalized distance thresholds. InfiniGen instead retains stronger class-level coverage, consistent with its dominant share of the synthetic training distribution. These contrasting trends show that semantic inventory recovery, center localization, and volumetric agreement capture distinct errors. On InfiniGen, 9B also lowers yaw error, indicating better orientation reasoning despite similar matched IoU.

Model scale. Moving from 4B to 9B improves F1@5% and F1@10% on all splits, with the clearest gains on InfiniGen and consistent gains on SpatialGen. The improvement is not uniform: on InfiniGen, 4B retains slightly stronger class-level performance and matched 3D IoU, whereas 9B lowers orientation error. Capacity therefore primarily improves spatial localization rather than every aspect of reconstruction. 3D IoU is stricter than center-based Localization F1 because it penalizes errors along all three axes, especially for thin objects such as wall art.

Out-of-domain transfer. Both variants show a clear localization drop on SpatialGen, while class-level recall remains substantially higher than localization F1. Together with low raw class precision and predicted inventories of 12.2 (9B) and 14.1 (4B) objects against a 6.3 ground-truth mean, this gap indicates that LayoutVLM still recognizes plausible room categories but over-generates unsupported or duplicated entities and places them less reliably under distribution shift. OOD error therefore arises from both inventory calibration and metric placement, motivating the grounded reconciliation and spatial refinement evaluated below.

5.4 Ablation Studies

Table 4 evaluates refinement on 48 OOD SpatialGen scenes: Stage 1 adds Asset Grounder reconciliation and deterministic repair, and Stage 2 adds the Critic–Editor–Verify loop.

Stage 1 (Asset Grounder) provides the largest gain for both models. Reconciliation, blueprint containment, and axis snapping raise F1@5% from $0.244 \rightarrow 0.311$ for 9B and $0.234 \rightarrow 0.308$ for 4B (+27.5% and +31.6%) without the agentic BEV-Editor; F1@10% rises from $0.328 \rightarrow 0.401$ and $0.311 \rightarrow 0.398$. Predicted counts fall from $12.2 \rightarrow 6.5$ and $14.1 \rightarrow 6.7$, approaching the 6.3 ground-truth mean, with roughly 5–7 fewer false positives per scene. The accompanying precision gains show that unsupported, duplicated, or out-of-room entities dominate raw errors. Although pruning can reduce class recall, it improves the scene-level

Table 3: Raw structured scene prediction on in-domain (Infinigen, SUN RGB-D) and out-of-domain (SpatialGen) splits, macro-averaged over 5–6 inference runs (mean $\pm$ std). Cls. P/R/F1 is the location-agnostic class-count-matching precision/recall/F1 (micro-averaged); it disentangles semantic recognition from spatial placement.

Model	Dataset	F1@5%↑	F1@10%↑	Cls.P↑	Cls.R↑	Cls.F1↑	3D-IoU$_{match}$↑	ICP-Rot(°)↓
4B	Infinigen (in-domain)	$0.299 \pm .003$	$0.471 \pm .002$	$0.858 \pm .002$	$\mathbf{0.801} \pm .001$	$\mathbf{0.822} \pm .001$	$\mathbf{0.267} \pm .003$	28.3 ± 0.8
9B	Infinigen (in-domain)	$\mathbf{0.331} \pm .003$	$\mathbf{0.511} \pm .002$	$\mathbf{0.889} \pm .002$	$0.758 \pm .002$	$0.800 \pm .002$	$0.265 \pm .006$	$\mathbf{26.5} \pm 1.9$
4B	SUN RGB-D (in-domain)	$0.515 \pm .005$	$0.601 \pm .007$	$\mathbf{0.684} \pm .005$	$0.655 \pm .003$	$0.635 \pm .003$	$0.420 \pm .004$	–
9B	SUN RGB-D (in-domain)	$\mathbf{0.519} \pm .007$	$\mathbf{0.620} \pm .005$	$0.678 \pm .002$	$\mathbf{0.697} \pm .003$	$\mathbf{0.653} \pm .003$	$\mathbf{0.427} \pm .002$	–
4B	SpatialGen (OOD, raw)	$0.213 \pm .007$	$0.288 \pm .009$	$0.332 \pm .008$	$\mathbf{0.697} \pm .020$	$0.413 \pm .007$	$0.214 \pm .009$	–
9B	SpatialGen (OOD, raw)	$\mathbf{0.239} \pm .009$	$\mathbf{0.333} \pm .014$	$\mathbf{0.428} \pm .018$	$0.677 \pm .019$	$\mathbf{0.477} \pm .015$	$\mathbf{0.217} \pm .014$	–

Table 4: Stage-wise ablation on the SpatialGen out-of-domain multi-view split. Stage 1 = AssetGrounder; Stage 2 = + agentic Critic–Editor–Verify loop.

Model	Stage	F1@5%↑	F1@10%↑	Cls. F1↑	Cls. Pre.↑	Cls. Recall↑
9B	raw	0.244	0.328	0.484	0.446	**0.701**
	stage1	0.311	0.401	**0.571**	**0.576**	0.650
	stage2	**0.327**	**0.410**			
4B	raw	0.234	0.311	0.429	0.359	**0.745**
	stage1	0.308	0.398	**0.566**	**0.554**	0.673
	stage2	**0.347**	**0.432**			

Table 5: Comparison on 3D layout extraction across single-view methods.

Model	F1@5%	P@5%	R@5%	Cls. F1	Cls. P	Cls. R	3D IoU
SAM3D	0.207	**0.335**	0.171				0.1102
Gen3DSR	0.200	0.326	0.165	0.396	**0.618**	0.292	0.1340
3D-Fixer	0.192	0.333	0.150				0.0456
Ours (9B)	**0.355**	0.316	0.463	**0.457**	0.414	0.578	0.179
Ours (4B)	0.287	0.209	**0.564**	0.381	0.282	**0.705**	**0.181**

metrics overall.

Stage 2 (agentic loop) yields an additional, model-size-dependent gain. F1@5% improves by $+0.016$ for 9B ($0.311 \rightarrow 0.327$) and $+0.039$ for 4B ($0.308 \rightarrow 0.347$), while F1@10% rises to 0.410 and 0.432. Class precision is unchanged because the loop edits spatial parameters rather than inventory. The full pipeline improves F1@5% over raw predictions by 34.0% and 48.3%: Stage 1 contributes most, while closed-loop editing consistently adds value.

5.5 Single View Analysis

We compare with reconstruction methods on the single-view SpatialGen subset. These object-centric baselines reconstruct detected instances rather than predict a complete, editable scene program from an image alone. For a fair layout comparison, all use Gen3DSR's 2D front end: CropFormer (Qi et al. 2022) produces panoptic masks, OneFormer (Jain et al. 2023) separates foreground and background, and OVSAM (Yuan et al. 2024b,a) assigns labels.

Under this shared front end, the baselines inherit identical masks, labels, and class scores. Our 9B/4B variants achieve F1@5% of $0.355/0.287$ versus 0.192–0.207, localization recall of $0.463/0.564$ versus 0.150–0.171, and 3D IoU of 0.179–0.181 versus 0.046–0.134. The gain comes with lower precision, especially for 4B; 9B has the highest class F1 (0.457), whereas 4B favors class recall (0.705) over precision (0.282).

Why detection-based baselines fail. The shared front end exposes a compounding error chain. Rich textures can fragment one object into multiple masks with inconsistent labels; these errors propagate to the reconstructed inventory, while camera and monocular-depth errors accumulate during 3D lifting. The resulting scene remains constrained by its 2D detections and often misses hidden structure. Although preference-aligned models such as SAM3D can complete an occluded part of a detected object, they cannot recover a fully occluded object of another category or infer its scene-level relations. LayoutVLM instead predicts identities and spatial relations jointly as a room-level program, explaining its higher recall and geometric agreement.

Visualizations. Figure 4(a) shows three XScene test cases with ceilings and occluding walls removed. Predictions generally match the scenes; Case 2 (row 2) shows reflection, rotation, or diagonal-symmetry ambiguity because camera relationships are not modeled, while Case 3 shows category errors such as confusing a bookcase with a large shelf or predicting a standing sink.

Figure 4(b) compares single-view results. Even post-trained SAM3D misses the nightstand beneath the left lamp because it lacks visible pixels, whereas Scenix recovers it. More generally, segmentation can yield plausible regions yet assign incorrect semantics, causing asset-based pipelines to omit structural entities, misplace assets, and lose editability. Direct scene-program prediction instead decouples structure from asset realization, allowing each stage to improve independently and supporting human edits.

6 Conclusion

We introduced Scenix, a framework for editable 3D indoor scene synthesis from sparse, uncalibrated RGB views without external geometric priors. It directly generates executable scene programs using LayoutVLM for structural induction, Asset Grounder for perception-grounded asset generation, and an agentic Critic–Editor–Verify loop for spatial refinement. We also constructed XScene, over 110,000 scenes with observation-consistent supervision. Experiments show that Scenix bridges structured prediction and open-world 3D realization, outperforming single-view baselines while preserving object-level editability across diverse environments.

References

Ardelean, A.; Özer, M.; and Egger, B. 2025. Gen3DSR: Generalizable 3D Scene Reconstruction via Divide and Conquer from a Single View. ArXiv:2404.03421 [cs.CV].

Avetisyan, A.; Xie, C.; Howard-Jenkins, H.; Yang, T.-Y.; Aroudj, S.; Patra, S.; Zhang, F.; Frost, D.; Holland, L.; Orme, C.; Engel, J.; Miller, E.; Newcombe, R.; and Balntas, V. 2024. SceneScript: Reconstructing Scenes with an Autoregressive Structured Language Model. In *Computer Vision – ECCV 2024*, volume 15119 of *Lecture Notes in Computer Science*, 247–263.

Chang, A.; Dai, A.; Funkhouser, T.; Halber, M.; Niessner, M.; Savva, M.; Song, S.; Zeng, A.; and Zhang, Y. 2017. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*.

Chen, B.; Xu, Z.; Kirmani, S.; Ichter, B.; Sadigh, D.; Guibas, L.; and Xia, F. 2024. SpatialVLM: Endowing Vision-Language Models with Spatial Reasoning Capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14455–14465.

Chen, X.; Chu, F.-J.; Gleize, P.; Liang, K. J.; Sax, A.; Tang, H.; Wang, W.; Guo, M.; Hardin, T.; Li, X.; et al. 2026. Sam 3d: 3dfy anything in images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7220–7232.

Dai, A.; Chang, A. X.; Savva, M.; Halber, M.; Funkhouser, T.; and Nießner, M. 2017. ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5828–5839.

Fang, C.; Li, H.; Liang, Y.; Zheng, J.; Mao, Y.; Liu, Y.; Tang, R.; Zhou, Z.; and Tan, P. 2026. SpatialGen: Layout-guided 3D Indoor Scene Generation. In *2026 International Conference on 3D Vision (3DV)*, 1392–1402.

Feng, W.; Zhu, W.; Fu, T.-J.; Jampani, V.; Akula, A.; He, X.; Basu, S.; Wang, X. E.; and Wang, W. Y. 2023. LayoutGPT: Compositional Visual Planning and Generation with Large Language Models. In *Advances in Neural Information Processing Systems*, volume 36, 18225–18250.

Fu, H.; Cai, B.; Gao, L.; Zhang, L.-X.; Wang, J.; Li, C.; Zeng, Q.; Sun, C.; Jia, R.; Zhao, B.; and Zhang, H. 2021. 3D-FRONT: 3D Furnished Rooms With layOuts and semaNTics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10933–10942.

Fu, R.; Wen, Z.; Liu, Z.; and Sridhar, S. 2024. AnyHome: Open-Vocabulary Generation of Structured and Textured 3D Homes. In *Computer Vision – ECCV 2024*, volume 15097 of *Lecture Notes in Computer Science*, 52–70.

Gümeli, C.; Dai, A.; and Nießner, M. 2022. ROCA: Robust CAD Model Retrieval and Alignment From a Single Image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4022–4031.

Hu, Y.; Wang, L.; Liu, X.; Chen, L.-H.; Guo, Y.; Shi, Y.; Liu, C.; Rao, A.; Wang, Z.; and Xiong, H. 2026. Simulating the Real World: A Unified Survey of Multimodal Generative Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–20.

Hunyuan3D, T.; Yang, S.; Yang, M.; Feng, Y.; Huang, X.; Zhang, S.; He, Z.; Luo, D.; Liu, H.; Zhao, Y.; et al. 2025. Hunyuan3d 2.1: From images to high-fidelity 3d assets with production-ready pbr material. *arXiv preprint arXiv:2506.15442*.

Jain, J.; Li, J.; Chiu, M. T.; Hassani, A.; Orlov, N.; and Shi, H. 2023. OneFormer: One Transformer To Rule Universal Image Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2989–2998.

Leroy, V.; Cabon, Y.; and Revaud, J. 2024. Grounding Image Matching in 3D with MASt3R. In *Computer Vision – ECCV 2024*, 71–91.

Meng, Y.; Wu, H.; Zhang, Y.; and Xie, W. 2026. SceneGen: Single-Image 3D Scene Generation in One Feedforward Pass. In *2026 International Conference on 3D Vision (3DV)*, 543–553.

Nie, Y.; Han, X.; Guo, S.; Zheng, Y.; Chang, J.; and Zhang, J. J. 2020. Total3DUnderstanding: Joint Layout, Object Pose and Mesh Reconstruction for Indoor Scenes From a Single Image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 55–64.

Niemeyer, M.; Barron, J. T.; Mildenhall, B.; Sajjadi, M. S. M.; Geiger, A.; and Radwan, N. 2022. RegNeRF: Regularizing Neural Radiance Fields for View Synthesis From Sparse Inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5480–5490.

Paschalidou, D.; Kar, A.; Shugrina, M.; Kreis, K.; Geiger, A.; and Fidler, S. 2021. ATISS: Autoregressive Transformers for Indoor Scene Synthesis. In *Advances in Neural Information Processing Systems*, volume 34, 12013–12026.

Pintore, G.; Agus, M.; and Gobbetti, E. 2020. AtlantaNet: Inferring the 3D Indoor Layout from a Single 360° Image Beyond the Manhattan World Assumption. In *Computer Vision – ECCV 2020*, volume 12353 of *Lecture Notes in Computer Science*, 432–448.

Qi, L.; Kuen, J.; Wang, Y.; Gu, J.; Zhao, H.; Torr, P.; Lin, Z.; and Jia, J. 2022. Open world entity segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(7): 8743–8756.

Raistrick, A.; Lipson, L.; Ma, Z.; Mei, L.; Wang, M.; Zuo, Y.; Kayan, K.; Wen, H.; Han, B.; Wang, Y.; Newell, A.; Law, H.; Goyal, A.; Yang, K.; and Deng, J. 2023. Infinite Photorealistic Worlds using Procedural Generation. ArXiv:2306.09310 [cs].

Roberts, M.; Ramapuram, J.; Ranjan, A.; Kumar, A.; Bautista, M. A.; Paczan, N.; Webb, R.; and Susskind, J. M. 2021. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *Proceedings of the IEEE/CVF international conference on computer vision*, 10912–10922.

Schneider, M.-A.; and Dai, A. 2026. WorldMesh: Generating Navigable Multi-Room 3D Scenes via Mesh-Conditioned Image Diffusion. ArXiv:2603.22972 [cs].

Schult, J.; Tsai, S.; Höllein, L.; Wu, B.; Wang, J.; Ma, C.-Y.; Li, K.; Wang, X.; Wimbauer, F.; He, Z.; Zhang, P.; Leibe, B.; Vajda, P.; and Hou, J. 2024. ControlRoom3D: Room Generation using Semantic Proxy Rooms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6201–6210.

Shi, Y.; Li, W.; Wang, Z.; Li, H.; Chen, X.; Tan, P.; and Zhang, L. 2026. SceneMaker: Open-set 3D Scene Generation with Decoupled De-occlusion and Pose Estimation Model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 27146–27156.

Song, S.; Lichtenberg, S. P.; and Xiao, J. 2015. SUN RGB-D: A RGB-D Scene Understanding Benchmark Suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 567–576.

Sun, C.; Hsiao, C.-W.; Sun, M.; and Chen, H.-T. 2019. HorizonNet: Learning Room Layout With 1D Representation and Pano Stretch Data Augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1047–1056.

Tang, J.; Nie, Y.; Markhasin, L.; Dai, A.; Thies, J.; and Nießner, M. 2024. DiffuScene: Denoising Diffusion Models for Generative Indoor Scene Synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20507–20518.

Wan, C.; Mao, Y.; Wu, W.; Xie, Y.; Xiang, C.; Wang, R.; Zhang, X.; Liu, Z.; Dai, R.; and Liu, Y. 2026. NaLA: A 3D Native LLM Layout Agent for High-quality 3D Scene Generation. *arXiv preprint arXiv:2606.29395*.

Wang, H.; and Agapito, L. 2025. 3D Reconstruction with Spatial Memory. In *2025 International Conference on 3D Vision (3DV)*, 78–89.

Wang, J.; Chen, M.; Karaev, N.; Vedaldi, A.; Rupprecht, C.; and Novotny, D. 2025a. VGGT: Visual Geometry Grounded Transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5294–5306.

Wang, Q.; Zhang, Y.; Holynski, A.; Efros, A. A.; and Kanazawa, A. 2025b. Continuous 3D Perception Model with Persistent State. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10510–10522.

Wang, S.; Leroy, V.; Cabon, Y.; Chidlovskii, B.; and Revaud, J. 2024. DUSt3R: Geometric 3D Vision Made Easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20697–20709.

Wang, X.; Yeshwanth, C.; and Nießner, M. 2021. SceneFormer: Indoor Scene Generation with Transformers. In *2021 International Conference on 3D Vision (3DV)*, 106–115.

Wu, Q.; Iliash, D.; Ritchie, D.; Savva, M.; and Chang, A. X. 2025. Diorama: Unleashing Zero-shot Single-view 3D Indoor Scene Modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8896–8907.

Xia, H.; Li, X.; Li, Z.; Ma, Q.; Xu, J.; Liu, M.-Y.; Cui, Y.; Lin, T.-Y.; Ma, W.-C.; Wang, S.; Song, S.; and Wei, F. 2026. SAGE: Scalable Agentic 3D Scene Generation for Embodied AI. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 22358–22368.

Xiang, J.; Lv, Z.; Xu, S.; Deng, Y.; Wang, R.; Zhang, B.; Chen, D.; Tong, X.; and Yang, J. 2025. Structured 3D Latents for Scalable and Versatile 3D Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21469–21480.

Yang, J.; Pavone, M.; and Wang, Y. 2023. FreeNeRF: Improving Few-Shot Neural Rendering With Free Frequency Regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8254–8263.

Yang, Y.; Jia, B.; Zhang, S.; and Huang, S. 2025. SceneWeaver: All-in-One 3D Scene Synthesis with an Extensible and Self-Reflective Agent. In *Advances in Neural Information Processing Systems*, volume 38, 140319–140351.

Yang, Y.; Sun, F.-Y.; Weihs, L.; VanderBilt, E.; Herrasti, A.; Han, W.; Wu, J.; Haber, N.; Krishna, R.; Liu, L.; Callison-Burch, C.; Yatskar, M.; Kembhavi, A.; and Clark, C. 2024. Holodeck: Language Guided Generation of 3D Embodied AI Environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16227–16237.

Ye, C.; Wu, Y.; Lu, Z.; Chang, J.; Guo, X.; Zhou, J.; Zhao, H.; and Han, X. 2025. Hi3dgen: High-fidelity 3d geometry generation from images via normal bridging. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 25050–25061.

Yeshwanth, C.; Liu, Y.-C.; Nießner, M.; and Dai, A. 2023. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 12–22.

Yin, Z.-X.; Liu, L.; Wang, X.; Sui, W.; Su, Z.; Yang, J.; and Xie, J. 2026. 3D-Fixer: Coarse-to-Fine In-place Completion for 3D Scenes from a Single Image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12753–12763.

Yuan, H.; Li, X.; Qi, L.; Zhang, T.; Yang, M.-H.; Yan, S.; and Loy, C. C. 2024a. Mamba or RWKV: Exploring High-Quality and High-Efficiency Segment Anything Model. *arXiv preprint*.

Yuan, H.; Li, X.; Zhou, C.; Li, Y.; Chen, K.; and Loy, C. C. 2024b. Open-Vocabulary SAM: Segment and Recognize Twenty-thousand Classes Interactively. In *ECCV*.

Yue, Y.; Kontogianni, T.; Schindler, K.; and Engelmann, F. 2023. Connecting the Dots: Floorplan Reconstruction Using Two-Level Queries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 845–854.

Zheng, J.; Zhang, J.; Li, J.; Tang, R.; Gao, S.; and Zhou, Z. 2020. Structured3D: A Large Photo-realistic Dataset for Structured 3D Modeling. In *Computer Vision – ECCV 2020*, volume 12354 of *Lecture Notes in Computer Science*, 519–535.

Zou, C.; Colburn, A.; Shan, Q.; and Hoiem, D. 2018. LayoutNet: Reconstructing the 3D Room Layout From a Single RGB Image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2051–2059.