SceneLM: 3D-Aware Language Models for Editable 3D Scene Synthesis

Xingbo Yao[1] and Xiaoyu Chen[1] and Doudou Zhang[1] and Mingzhi Sheng[1]
Boyuan Cao[3] and Yingcong Chen[1,2] and Hui Xiong[1,2]
[1]The Hong Kong University of Science and Technology (Guangzhou)
[2]The Hong Kong University of Science and Technology
[3]Fudan University

Abstract

Synthesizing an *editable* 3D scene from a single RGB image is central to content creation, embodied-agent data generation, and AR/VR, yet remains challenging to achieve both high-fidelity reconstruction and convenient interactive editing. Existing geometry-based pipelines produce high-quality 3D results but are typically hard to refine without rerunning the full process, while LLM-driven procedural systems enable interactive tool use but are mostly text-driven and lack *precise* metric 3D understanding from images. We present **SceneLM**, a language-model-based framework that grounds 3D scene synthesis in visual evidence by recovering an *executable metric 3D layout* directly from a single image. Given an RGB image (and camera intrinsics when available), SceneLM outputs a JSON-form layout specifying each object's category, 3D center, size, and discretized yaw, and then deterministically executes this layout with a tool suite to instantiate, place, and edit objects for iterative refinement. To train metric layout recovery at scale, we curate five datasets covering diverse indoor, outdoor, and tabletop scenes and convert heterogeneous 3D annotations into a unified instruction-tuning format. To improve numerical stability and metric accuracy while preserving the text interface, we augment autoregressive JSON generation with a lightweight geometry prediction branch and dual supervision. Experiments show that SceneLM substantially improves single-image 3D layout estimation over strong open and proprietary MLLM baselines, and yields higher-quality end-to-end scene generation in geometric consistency, physical plausibility, semantic alignment, and realism.

1 Introduction

Synthesizing an *editable* 3D scene from a single RGB image is a long-standing goal with broad impact in content creation, embodied-agent data generation, and AR/VR. It offers a direct path to instantiate diverse 3D scenes from everyday visual inputs. However, it remains difficult to simultaneously achieve high-fidelity scene synthesis and *convenient, interactive* editing.

Existing approaches largely fall into two categories. **Geometry-based** reconstruction and generation methods leverage explicit structures (e.g., layouts, scene graphs, intermediate geometric representations) to enable high-quality synthesis (Fang et al., 2025; Wang et al., 2025a; Feng et al., 2025; Yang et al., 2025d; Bokhovkin et al., 2025; Liu et al., 2025; Yang et al., 2025c; Kalischek et al., 2025; Huang et al., 2025; Yu et al., 2025; Schwarz et al., 2025; Xia et al., 2025; Hou et al., 2025; Yao et al., 2025a,b; **?**). However, these pipelines are often hard to edit: correcting an unsatisfactory output typically requires rerunning the whole pipeline, which limits iterative refinement. In contrast, **LLM-driven procedural** systems enable language-conditioned planning and tool use, making scene construction interactive and compositional (SongTang et al., 2025; Zhang et al., 2025; Hao et al., 2025; Bucher and Armeni, 2025; Sun et al., 2025b; Deng et al., 2025; Yang et al., 2025a; Sun et al., 2025a). Yet most of these systems remain *text-driven*. Since language is inherently ambiguous, it rarely specifies a unique, metric layout, and metric 3D geometry is typically handled as an external component. Using images as input can ground generation in visual evidence and reduce linguistic ambiguity, but it requires recovering *precise* object-level 3D geometry from a single view. Current VLMs generally provide only coarse 3D cues or relative spatial relations for monocular 3D reasoning and grounding (Cho et al., 2024; Man et al., 2025; Wang et al., 2025b; Yang et al., 2024; Mao et al., 2025), which can support conversational 3D grounding. However, these signals remain inadequate for *executable* scene construction that requires *precise metric layouts* with accurate centers, sizes, and orientations.

Findings of the Association for Computational Linguistics: ACL 2026, pages 42615–42624

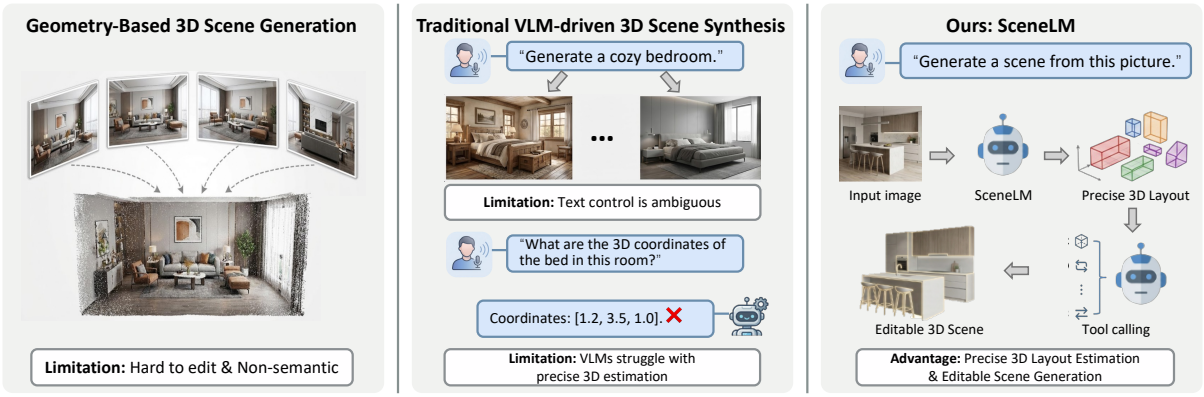

Figure 1: **Teaser.** (Left) Geometry-based pipelines reconstruct 3D structure from multi-view images but are difficult to edit. (Middle) Text-driven VLM approaches suffer from ambiguous prompts and unreliable metric 3D localization. (Right) SceneLM takes a single image, predicts a precise 3D layout (oriented boxes), and calls tools to synthesize an editable 3D scene.

To address the aforementioned issues, we propose **SceneLM**, a language-model-based framework for *editable* 3D scene synthesis from a single RGB image. Given an input image (and camera intrinsics when available), SceneLM *precisely* recovers an object-centric *metric* 3D layout by predicting each object's category, 3D center, size, and orientation yaw. To endow language models with this capability, we curate five large-scale datasets spanning diverse indoor, outdoor, and tabletop scenes, and cast heterogeneous 3D annotations into a unified, training-friendly instruction format that directly supervises metric layouts. Building on the predicted layout, we further develop a tool-based scene editing pipeline: SceneLM can invoke a set of deterministic scene-editing tools to instantiate assets, place them in 3D, and iteratively edit either the layout JSON or the resulting scene. Extensive experiments validate SceneLM on (i) single-image 3D layout estimation and (ii) end-to-end single-image 3D scene generation, demonstrating consistent improvements over strong open and proprietary MLLM baselines in metric localization accuracy, geometric consistency, and overall scene quality.

The key contributions are:

- We propose **SceneLM**, which recovers *precise* metric 3D bounding boxes from a single RGB image for executable scene construction.
- We curate **five large-scale datasets** covering diverse scene types and convert 3D spatial annotations into an instruction-tuning format that is scalable and extensible.
- We introduce a **tool-based scene synthesis and editing pipeline** that executes predicted layouts to generate editable 3D scenes and supports iterative refinement via explicit operations.
- We demonstrate **state-of-the-art performance** on single-image 3D layout estimation and improved end-to-end 3D scene generation quality under standard metrics.

2 Related Work

We group related work into two directions: (i) geometry-based 3D scene generation and reconstruction, which leverages explicit structure for high-quality synthesis, and (ii) LLM-driven procedural 3D scene synthesis, which enables language-conditioned planning and tool use.

2.1 Geometry Based 3D Scene Generation

Geometry-based methods generate 3D scenes with explicit geometric representations and intermediate layout structures. For indoor generation, SPATIALGEN performs layout-guided 3D indoor scene synthesis (Fang et al., 2025), and HLG constructs rooms via hierarchical layout generation (Wang et al., 2025a). CasaGPT targets interior design by predicting cuboid arrangements and assembling scenes (Feng et al., 2025). MMGDreamer introduces a mixed-modality graph for geometry-controllable 3D indoor generation (Yang et al., 2025d). SceneFactor studies factored latent 3D diffusion for controllable 3D scene generation through disentangled components (Bokhovkin et al., 2025).

Beyond indoor scenes, controllability has also been explored for outdoor and urban settings. Con-

trollable 3D Outdoor Scene Generation via Scene Graphs leverages scene graphs to guide the synthesis process (Liu et al., 2025). UrbanGen studies compositional and controllable neural fields for urban generation, focusing on factorized structure for large-scale outdoor environments (Yang et al., 2025c). A related stream builds 3D scenes from limited views by introducing intermediate representations such as panoramas or multi-plane structures. CubeDiff repurposes diffusion-based image models for panorama generation (Kalischek et al., 2025), and DreamCube generates 3D panoramas via multi-plane synchronization (Huang et al., 2025). WonderWorld studies interactive 3D scene generation from a single image (Yu et al., 2025), and A Recipe for Generating 3D Worlds From a Single Image explores complete world synthesis from single-view input (Schwarz et al., 2025). ScenePainter and BloomScene further explore semantically consistent or lightweight structured 3D Gaussian Splatting pipelines for cross-modal scene generation (Xia et al., 2025; Hou et al., 2025).

However, these geometry-based pipelines are typically end-to-end and *hard to edit*. Unsatisfactory results often require rerunning the pipeline or regenerating the scene, which hinders *iterative refinement*.

2.2 LLM-driven Procedural 3D Scene Synthesis

Recent research increasingly leverages LLMs and VLMs to generate 3D worlds by producing programs, action sequences, or structured plans that are executed by a renderer, asset library, or simulator. UnrealLLM and The Scene Language frame scene construction as code generation, using language to produce executable programs or explicit procedures (SongTang et al., 2025; Zhang et al., 2025). Beyond direct execution, several approaches focus on high-level reasoning and hierarchical planning. MesaTask formulates task-driven tabletop scene generation via explicit 3D spatial reasoning (Hao et al., 2025), while ReSpace employs LLMs to support controllable scene synthesis and editing with preference alignment (Bucher and Armeni, 2025). Hierarchically-structured indoor scene synthesis further explores open-vocabulary scene planning using pre-trained LLMs (Sun et al., 2025b). VLM-based systems also incorporate structured search and planning, such as global-local tree search, to iteratively refine indoor scene generation (Deng et al., 2025). Recent systems further adopt agentic designs that incorporate self-reflection and tool orchestration. SceneWeaver proposes an all-in-one 3D scene synthesis agent with an extensible toolset and iterative refinement (Yang et al., 2025a). Beyond single agents, 3D-Generalist targets vision-language-action learning for crafting 3D worlds with self-improvement signals (Sun et al., 2025a).

However, existing LLM-based 3D scene synthesis is still largely text-driven. Since language is inherently ambiguous, it rarely specifies a unique, metric layout. Using images as input can ground generation in the visual evidence and reduce this ambiguity, but it requires the model to recover object-level 3D geometry from a single view. Recent works have explored empowering VLMs to predict object-level 3D representations directly from monocular images. Cube-LLM studies inferring object-level 3D representations from monocular images using language-guided visual reasoning (Cho et al., 2024), while LocateAnything3D introduces chain-of-sight prompting for 3D localization (Man et al., 2025). N3D-VLM and LLMI3D further explore endowing VLMs with coarse 3D perception from a single view (Wang et al., 2025b; Yang et al., 2024).

Despite these efforts, current methods often recover only coarse 3D cues or relative spatial relations. This is sufficient for conversational 3D grounding, but inadequate for *executable* scene construction that requires *precise 3D layouts*.

3 Methods

3.1 Task Formulation

We study *single-image* editable 3D scene synthesis, formulated as *metric 3D layout estimation* followed by *tool-based scene synthesis* (Fig. 2). Given an RGB image I and camera intrinsics K, **SceneLM** predicts an object-centric *metric 3D layout* $\mathcal{L}$ as a structured JSON list:

$$\mathcal{L} = \Big[\{\texttt{class}, \texttt{center}, \texttt{size}, \texttt{yaw\_bin}\}\Big]_{i=1}^{M} \quad (1)$$

Here `class` denotes the semantic category of each object, $\texttt{center} \in R^3$ and $\texttt{size} \in R^3$ specify the 3D box center and dimensions in millimeters, and `yaw_bin` is a discretized yaw label with 15° per bin.

The predicted layout $\mathcal{L}$ serves as an executable intermediate representation. We design a tool suite that deterministically translates $\mathcal{L}$ into function

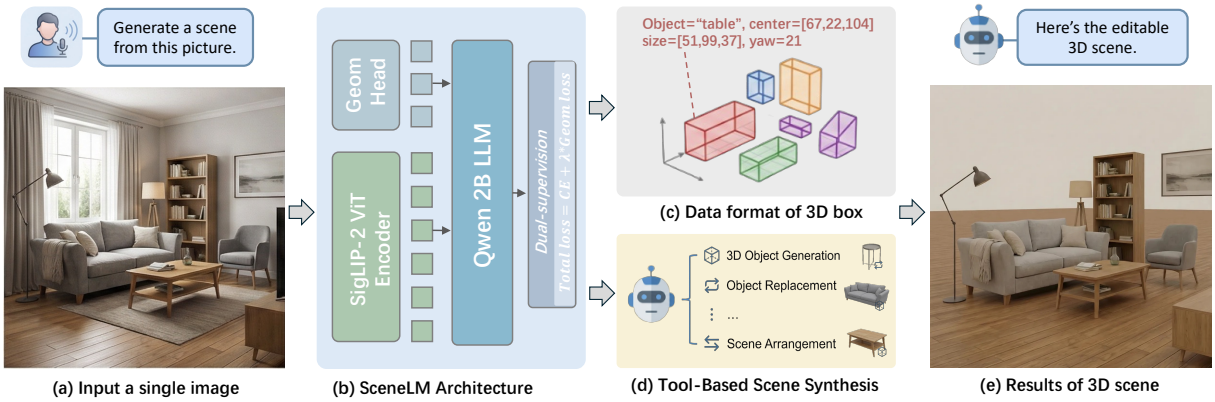

Figure 2: **Framework overview. SceneLM** recovers a metric 3D layout from a single image and executes it to synthesize an editable 3D scene. (a) Input image (with intrinsics). (b) SceneLM combines a SigLIP-2 encoder, a geometry-aware head, and a Qwen-VL-2B-Instruct LLM, trained with dual supervision to improve 3D layout accuracys. (c) The layout is an object list with `center` (mm), `size` (mm), and `yaw`. (d) SceneLM invokes tools for asset generation and scene assembly. (e) Final editable 3D scene.

calls for object generation, scene arrangement, and scene editing, producing an editable 3D scene $\mathcal{S}$:

$$\mathcal{S} = \mathcal{T}(\mathcal{L}). \tag{2}$$

Edits are applied by updating the layout fields and re-executing the corresponding tool calls, enabling iterative refinement without rerunning the full generation pipeline.

3.2 3D Layout Estimation from 2D Image

Challenge. Current VLMs mostly capture coarse relations (e.g., *left of*, *near*) and *struggle to infer accurate 3D layout* from a single image. *Without explicit projective reasoning*, their predictions are often inconsistent with perspective cues and camera intrinsics. Yet editable 3D scene synthesis requires object-level metric layouts with precise centers, sizes, and orientations that are directly executable by downstream tools.

SceneLM for layout estimation. As shown in Fig. 2(b), SceneLM predicts an executable metric 3D layout from an RGB image I and intrinsics K. We use **SigLIP-2 ViT** (Tschannen et al., 2025) to encode I into patch features $\mathbf{V} \in R^{N\times C}$, and a lightweight **vision projector** compresses $\mathbf{V}$ into M visual tokens $\mathbf{T}_v \in R^{M\times d}$ aligned with the LLM hidden size d. The visual tokens are concatenated with the text prompt and decoded by **Qwen-2B** (Bai et al., 2023) into a JSON list, where each object is represented by (`category`, `center_mm`, `size_mm`, `yaw_bin`) (Fig. 2(c)).

Geometry head for metric learning. Autoregressive JSON training relies on next-token prediction and can produce numerically unstable metric fields, especially orientations. To explicitly enforce metric correctness without changing the text interface, we attach a **differentiable geometry head** on the last-layer LLM hidden states. For each object, we extract an object-specific hidden vector $\mathbf{h}_o$ (e.g., at a dedicated anchor token in the target sequence) and predict

$$(\hat{\mathbf{c}}, \hat{\mathbf{s}}, \hat{y}) = \mathrm{MLP}(\mathbf{h}_o), \tag{3}$$

where $\hat{\mathbf{c}}, \hat{\mathbf{s}} \in R^3$ denote center/size in millimeters, and $\hat{y}$ is a 24-way `yaw_bin` classification with 15° per bin. This branch is auxiliary: at inference the model still outputs only JSON text.

Dual supervision. We jointly optimize language validity and metric accuracy:

$$\mathcal{L} = \mathcal{L}_{\mathrm{CE}} + \lambda\, \mathcal{L}_{\mathrm{geom}}. \tag{4}$$

$\mathcal{L}_{\mathrm{CE}}$ is the standard autoregressive cross-entropy on the JSON string. $\mathcal{L}_{\mathrm{geom}}$ applies Huber loss to $(\mathbf{c}, \mathbf{s})$ and cross-entropy to y:

$$\mathcal{L}_{\mathrm{geom}} = \mathrm{Huber}(\hat{\mathbf{c}}, \mathbf{c}) + \mathrm{Huber}(\hat{\mathbf{s}}, \mathbf{s}) + \mathrm{CE}(\hat{y}, y). \tag{5}$$

This dual supervision stabilizes metric fields (e.g., yaw modes) and improves downstream execution, while preserving the original JSON-generation behavior.

Executable output. The generated JSON layout (Fig. 2(c)) is parsed into a *metric 3D layout* and will be executed by our *scene-editing tools* to instantiate, place, and edit objects.

3.3 Data Preparation and Learning Strategy

Data preparation. A key challenge is the absence of a unified, executable supervision format for monocular metric 3D layout across diverse real-world scenes. We therefore curate a heterogeneous training corpus spanning indoor environments (Omni3D (Brazil et al., 2023), SUN RGB-D (Song et al., 2015), ARKitScenes (Baruch et al., 2021), Hypersim (Roberts et al., 2021)), outdoor driving scenes (KITTI (Geiger et al., 2013)), and multi-object tabletop scenes (Objectron (Ahmadyan et al., 2021)). Despite dataset-specific annotation conventions, each sample is normalized into a consistent *image-conditioned 3D layout* representation consisting of an RGB image, camera intrinsics, and per-object metric 3D bounding boxes. Each object is parameterized by `center`, `size`, and `yaw`.

To align with instruction-tuned VLM training, we serialize each sample into the LLaMAFactory message format (Fig. 3). The prompt provides the RGB image and intrinsics $K = (f_x, f_y, c_x, c_y)$ and requests an *executable JSON array* of all objects. The target follows `{category, bbox3d{center_mm, size_mm, yaw_bin}}`, where metric values are encoded as integer millimeters to ease tokenization and reduce floating-point artifacts. Since continuous angle regression is unstable in autoregressive decoding, yaw is discretized into 24 bins (15°/bin), i.e., `yaw_bin`$\in \{0, \ldots, 23\}$.

Learning strategy. Training uses token-level cross-entropy on the target JSON sequence to encourage syntactically valid and complete structured outputs. However, language-model supervision alone often produces numerically plausible yet geometrically inconsistent layouts. To explicitly enforce metric accuracy while preserving the original JSON generation interface, we attach a lightweight geometry head (MLP) to the last-layer hidden states to predict per-object $\{\hat{\mathbf{c}}, \hat{\mathbf{s}}, \hat{y}\}$ for `center_mm`, `size_mm`, and `yaw_bin`. The geometry head is supervised with a robust regression loss for center/size and a classification loss for yaw, and the overall objective is:

$$\mathcal{L} = \mathcal{L}_{CE} + \lambda_{geom}\mathcal{L}_{geom}.$$

3.4 Tool-Based Scene Synthesis

As illustrated in Fig. 2(d–e), the predicted JSON layout $\mathcal{L} = \{\ell_i\}_{i=1}^{M}$ is directly executed to construct an editable 3D scene. We decouple *asset creation* from *scene assembly*. For each object ℓ_i, we first obtain a canonical 3D asset using an external generator (e.g., Hunyuan3D) conditioned on its category (and optional text attributes). We then instantiate the asset and apply the metric transform specified by $\mathcal{L}$ (`center`, `size`, and `yaw_bin`), producing a coherent scene that supports subsequent edits by updating object transforms.

To robustly bridge structured outputs and tool execution, we use a tool-using agent to convert $\mathcal{B}$ into deterministic function calls. The agent parses the JSON, retrieves tool signatures, and executes a sequence of operations to create, place, and edit objects. This design makes failures inspectable and enables iterative refinement through explicit layout updates.

We expose a compact API aligned with Fig. 2(d), including `3d_object_generation`, `scene_arrangement`, and `object_replacement`, and additionally support standard scene-editing utilities such as `get_tools_info`, `add_object`, `set_transform`, `remove_object`, `replace_object`, `query_scene`, and `export_scene`. Our SceneLM can further invoke these tools to edit either the metric 3D layout JSON or the instantiated scene directly.

4 Experiments

4.1 Implementation Details

We start from the official Qwen3-VL-2B-Instruct checkpoint, with a SigLIP-SO400M ViT vision encoder (24 layers, patch size 16) and the default visual projector. We fine-tune on 4× NVIDIA A800 (80GB) in BF16 for 20 epochs, using max sequence length 2048 and micro-batch size 1 with gradient accumulation 16. We attach a two-layer MLP geometry head to the last-layer hidden states to predict per-object `center_mm`, `size_mm`, and 24-way `yaw_bin` (15°/bin). The geometry head is initialized with $\mathcal{N}(0, 0.02)$ weights and zero biases. For supervision alignment, we anchor each object at the first digit token following `"yaw_bin":` in the ground-truth JSON and optimize a dual objective $\mathcal{L} = \mathcal{L}_{CE} + 0.5\,\mathcal{L}_{geom}$, where $\mathcal{L}_{geom}$ applies SmoothL1 to center/size and cross-entropy to yaw classification.

4.2 3D Layout Estimation from a Single Image

We evaluate the *layout estimation* module in our pipeline (Fig. 2(b–c)), which predicts an executable

```
{
  "images": [".../image.jpg"],
  "messages": [
    {
      "role": "user",
      "content":
        "<image>
        Task: Monocular 3D detection. Camera intrinsics: fx=..., fy=..., cx=..., cy=...
        Output ALL objects as a JSON array. Units: millimeters (integers).
        Schema: [{\"category\":\"...\",\"bbox3d\":{\"center_mm\":[x,y,z],\"size_mm\":[w,h,l],\"yaw_bin\":0}}]
        Rules: sort by category; return ONLY valid JSON.
        size_mm: [width, height, depth] in millimeters. yaw_bin: 0-23 (15° per bin)
        Example output:
        [{\"category\":\"chair\",\"bbox3d\":{\"center_mm\":[500,200,2000],\"size_mm\":[600,800,600],\"yaw_bin\":12}}]
       "
    },
    {
      "role": "assistant",
      "content":
      "[{\"category\":\"sofa\",\"bbox3d\":{\"center_mm\":[...],\"size_mm\":[...],\"yaw_bin\":...}}, ...]"
    }
  ]
}
```

Figure 3: Unified LLaMAFactory-format training sample (two-column friendly). In practice, the instruction can include more detailed constraints (e.g., category list, ordering rules, unit conventions, and failure cases); we show a simplified template for illustration.

Methods	Volumetric IoU↑	Collided Pairs↓	CLIP Score↑	FID↓
SceneGen	0.35	**0.23**	0.72	42.6
Qwen3-VL-30B-A3B-Instruct	0.39	1.39	0.80	29.1
Ours (SceneLM)	**0.53**	0.37	**0.85**	**20.8**

Table 1: Single-image 3D scene generation (Fig. 2(a–e)). We evaluate geometric agreement (Volumetric IoU), physical plausibility (Collided Pairs), semantic alignment (CLIP Score), and realism (FID).

metric 3D layout (a set of 3D bounding boxes) from a single RGB image. **Evaluation Metrics.** Following SpatialLM (Mao et al., 2025), we use the 3D IoU between two boxes, defined as the intersection-over-union of their volumes. A predicted box is treated as a true positive if its IoU_{3D} with a ground-truth box exceeds a threshold τ, with one-to-one matching between predictions and ground truth. We then report **F1@IoU$_{3D}$=τ** for $\tau \in \{0.25, 0.5\}$, where $\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$, $\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}}$, and $\text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}}$.

Baselines. We compare against Qwen3-VL Instruct models at different scales (2B/8B/30B-A3B), prompted to output the same JSON-form 3D layout. For open models, we uniformly sample 1,000 test images across the five benchmarks. We additionally report results from two proprietary MLLMs (GPT-5.2 and Gemini 3 Pro) on a 100-image subset due to API cost.

Results. As shown in Table 2, SceneLM achieves the highest F1 at both IoU_{3D} thresholds while using a 2B backbone. Notably, the gain is larger at IoU_{3D}=0.5, indicating more accurate metric recovery of object extent and pose. In contrast, scaling Qwen3-VL improves F1@0.25 but yields limited improvement at F1@0.5, suggesting that next-token supervision on JSON alone does not reliably enforce precise 3D geometry. Proprietary MLLMs outperform open baselines, yet remain below SceneLM on both metrics.

4.3 3D Scene Generation from a Single Image

Evaluation Metrics. Following SceneWeaver and SceneGen (Yang et al., 2025b; Meng et al., 2026), we report *Volumetric IoU* (↑), *Collided Pairs* (↓), *CLIP Score* (↑), and *FID* (↓).

Volumetric IoU measures the volumetric overlap between the synthesized scene occupancy and the reference, capturing geometric agreement.

Collided Pairs counts the number of object pairs whose 3D bounding volumes intersect beyond a small tolerance, reflecting physical plausibility.

CLIP Score evaluates image-text alignment between the rendered scene and the conditioning description (or reference caption), measuring semantic consistency.

FID compares the distribution of rendered images to real images in a deep feature space, where lower values indicate more realistic renderings.

Baselines. We compare with SceneGen (Meng et al., 2026), a feed-forward single-image 3D scene generation model, and the open MLLM baseline Qwen3-VL-30B-A3B-Instruct, prompted to output the same executable layout JSON and executed with the same toolchain for fair comparison.

Methods	Param	F1@0.25↑	F1@0.5↑
Gemini 3 Pro	–	43.5	22.3
GPT-5.2	–	47.3	29.2
Qwen3-VL-2B-Instruct	2B	18.2	12.4
Qwen3-VL-8B-Instruct	8B	26.2	10.8
Qwen3-VL-30B-A3B-Instruct	30B	41.5	23.6
Ours (SceneLM)	2B	**58.5**	**36.1**

Table 2: Single-image 3D layout estimation (Fig. 2(b–c)). We report F1 at IoU_{3D} thresholds 0.25 and 0.5. GPT-5.2 and Gemini 3 Pro are evaluated on 100 images due to API cost.

Quantitative Results. As shown in Table 1, SceneLM achieves the best overall performance across all metrics. In particular, it substantially improves geometric consistency, increasing Volumetric IoU from 0.35/0.39 to 0.53, while also delivering the strongest semantic alignment (CLIP score 0.85) and the most realistic renderings (FID 20.8). Compared with Qwen3-VL-30B-A3B-Instruct executed with the same toolchain, the gains suggest that accurate, metric layout prediction is critical for downstream scene assembly and rendering quality. For *Collided Pairs*, SceneGen attains the lowest value (0.23) largely because it often synthesizes fewer objects and misses small instances, which reduces the chance of inter-object intersections. In contrast, SceneLM reconstructs more complete scenes, and still maintains a low collision rate (0.37), indicating improved physical plausibility despite generating richer object sets.

Qualitative results. Fig. 4 shows that Qwen3-VL-30B-A3B-Instruct, when paired with our execution tools (Sec. 3.4), often outputs *incomplete* JSON layouts and imprecise metric boxes; the resulting scenes exhibit missing objects, distorted scales, and occasional interpenetrations after placement. SceneGen typically generates *reasonable* placements for the objects it synthesizes, yet it frequently drops small or secondary instances, producing sparse reconstructions (e.g., tabletop items). In contrast, SceneLM recovers a more *complete* object set and more consistent 3D boxes, yielding scenes that better match both the content and spatial arrangement of the input while remaining directly editable.

4.4 Ablation Study

Setup. We ablate key design choices in the *layout estimation* stage of SceneLM (Fig. 2(b–c)), while keeping the backbone, prompts, training data, and optimization hyperparameters fixed. Evaluation is conducted on 1,000 images uniformly sampled from the test splits of our five datasets, using F1 at IoU_{3D} thresholds 0.25 and 0.5. In Fig. 2, module (b) denotes the image-to-layout predictor (VLM with our geometry-aware heads), and (c) is its executable output: a JSON list of per-object `category`, `center_mm`, `size_mm`, and `yaw_bin`.

Variant	F1@0.25↑	F1@0.5↑
Ours (SceneLM)	**58.5**	**36.1**
w/o GeomHead + dual supervision	47.6	28.4
w/o yaw_bin (24)	44.1	25.1
Qwen3-VL-2B + JSON CE	18.2	12.4

Table 3: Ablations on single-image 3D layout estimation.

Tuning	F1@0.25↑	F1@0.5↑
LoRA	51.3	32.7
Full	**58.5**	**36.1**

Table 4: Comparison between LoRA and full fine-tuning for SceneLM on single-image 3D layout estimation.

Results. Removing the GeomHead and dual supervision reduces F1 to 47.6 at IoU_{3D}=0.25 and 28.4 at IoU_{3D}=0.5. The larger gap at IoU_{3D}=0.5 is consistent with the role of the geometry branch in improving high-precision metric localization. Disabling yaw discretization further lowers F1 to 44.1/25.1, which highlights orientation as a dominant failure mode under pure autoregressive JSON learning; treating yaw as a 24-way classification target provides a more stable supervision signal. The plain Qwen3-VL-2B trained with JSON cross-entropy reaches only 18.2/12.4, which indicates that language-only supervision does not yield accurate monocular 3D layouts without explicit geometric constraints.

4.5 Full Fine-tuning vs. LoRA

We further compare LoRA and full fine-tuning under the same backbone, prompts, data, and evaluation protocol. Full fine-tuning yields consistently

Figure 4: **Qualitative comparison on single-image 3D scene generation.** We compare SceneLM with Qwen3-VL-30B-A3B-Instruct executed using our toolchain (Sec. 3.4) and SceneGen. Qwen30B+Tool often misses objects and produces inaccurate box sizes/poses, which can lead to interpenetrations after instantiation. SceneGen yields plausible placements for generated instances but frequently omits small or less salient objects, resulting in sparse scenes. SceneLM produces more complete layouts with more consistent metric geometry, leading to more faithful and editable reconstructions across indoor, outdoor, and tabletop scenes.

higher F1 at both IoU_{3D} thresholds, suggesting that precise metric 3D layout prediction benefits from updating the full model capacity rather than a low-rank adaptation. In practice, LoRA remains a viable choice when compute or memory is constrained, but full fine-tuning provides the best accuracy for executable scene synthesis.

5 Limitations

To comply with the ACL policy, we summarize the limitations of this work. First, SceneLM relies on camera intrinsics for metric layout recovery; when intrinsics are unavailable, we use default parameters, which may bias the estimated absolute scale and object sizes. Second, the final scene quality is bounded by external asset generation and execution (e.g., asset fidelity, scale alignment, and collision handling), and the model may still output invalid or incomplete JSON in complex scenes. We note that these limitations are acceptable in our evaluation setting and do not prevent normal usage under typical conditions.

6 Conclusion

We present **SceneLM**, a language-model-based framework for *editable* 3D scene synthesis from a single RGB image. SceneLM grounds generation in visual evidence by recovering an *executable metric 3D layout* as a JSON list of object categories with metric centers, sizes, and orientations, and deterministically executing it with a tool suite to instantiate, place, and edit objects. To equip language models with accurate monocular 3D perception, we introduce a geometry prediction branch with dual supervision and curate five large-scale datasets that convert heterogeneous 3D annotations into a unified instruction-tuning format. Extensive experiments show that SceneLM improves single-image 3D layout estimation and yields higher-quality end-to-end scene generation in geometric consistency, physical plausibility, semantic alignment, and realism. We believe SceneLM's image-grounded metric layout prediction, tool-based scene synthesis, and the accompanying datasets provide a promising foundation for scalable 3D scene understanding and generation from everyday images. Future work will support richer interactive editing with long-horizon refinement, and enable one-click export to embodied simulators such as Habitat for embodied-agent training and evaluation.

References

Adel Ahmadyan, Liangkai Zhang, Artsiom Ablavatski, Jianing Wei, and Matthias Grundmann. 2021. Objectron: A large scale dataset of object-centric videos in the wild with pose annotations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7822–7831.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, and 1 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, and 1 others. 2021. Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data. *arXiv preprint arXiv:2111.08897*.

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

Garrick Brazil, Abhinav Kumar, Julian Straub, Nikhila Ravi, Justin Johnson, and Georgia Gkioxari. 2023. Omni3d: A large benchmark and model for 3d object detection in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13154–13164.

Martin J.J. Bucher and Iro Armeni. 2025. Respace: Text-driven 3d indoor scene synthesis and editing with preference alignment. *arXiv preprint arXiv:2506.02459*.

Jang Hyun Cho, B. Ivanovic, Yulong Cao, Edward Schmerling, Yue Wang, Xinshuo Weng, Boyi Li, Yurong You, Philipp Krahenbuhl, Yan Wang, and Marco Pavone. 2024. Language-image models with 3d understanding. *arXiv preprint arXiv:2405.03685*.

Wei Deng, Mengshi Qi, and Huadong Ma. 2025. Global-local tree search in vlms for 3d indoor scene generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8975–8984.

Chuan Fang, Heng Li, Yixun Liang, Jia Zheng, Yongsen Mao, Yuan Liu, Rui Tang, Zihan Zhou, and Ping Tan. 2025. Spatialgen: Layout-guided 3d indoor scene generation. *arXiv preprint arXiv:2509.14981*.

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

Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. 2013. Vision meets robotics: The kitti dataset. *The international journal of robotics research*, 32(11):1231–1237.

Jinkun Hao, Naifu Liang, Zhen Luo, Xudong Xu, Weipeng Zhong, Ran Yi, Yichen Jin, Zhaoyang Lyu, Feng Zheng, Lizhuang Ma, and Jiangmiao Pang. 2025. Mesatask: Towards task-driven tabletop scene generation via 3d spatial reasoning. *arXiv preprint arXiv:2509.22281*.

Xiaolu Hou, Mingcheng Li, Dingkang Yang, Jiawei Chen, Ziyun Qian, Xiao Zhao, Yue Jiang, Jinjie Wei, Qingyao Xu, and Lihua Zhang. 2025. Bloomscene: Lightweight structured 3d gaussian splatting for crossmodal scene generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 3536–3544.

Yukun Huang, Yanning Zhou, Jianan Wang, Kaiyi Huang, and Xihui Liu. 2025. Dreamcube: 3d panorama generation via multi-plane synchronization.

Nikolai Kalischek, Michael Oechsle, Fabian Manhardt, Philipp Henzler, Konrad Schindler, and Federico Tombari. 2025. Cubediff: Repurposing diffusion-based image models for panorama generation.

Yuheng Liu, Xinke Li, Yuning Zhang, Lu Qi, Xin Li, Wenping Wang, Chongshou Li, Xueting Li, and Ming-Hsuan Yang. 2025. Controllable 3d outdoor scene generation via scene graphs. *arXiv preprint arXiv:2503.07152*.

Yunze Man, Shihao Wang, Guowen Zhang, Johan Bjorck, Zhiqi Li, Liang-Yan Gui, Jim Fan, Jan Kautz, Yu-Xiong Wang, and Zhiding Yu. 2025. Locateanything3d: Vision-language 3d detection with chain-of-sight. *arXiv preprint arXiv:2511.20648*.

Yongsen Mao, Junhao Zhong, Chuan Fang, Jia Zheng, Rui Tang, Hao Zhu, Ping Tan, and Zihan Zhou. 2025. Spatiallm: Training large language models for structured indoor modeling. In *Advances in Neural Information Processing Systems*.

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

Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M Susskind. 2021. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10912–10922.

Katja Schwarz, Denis Rozumny, Samuel Rota Bulò, Lorenzo Porzi, and Peter Kontschieder. 2025. A recipe for generating 3d worlds from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3520–3530.

Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. 2015. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 567–576.

SongTang SongTang, Kaiyong Zhao, Lei Wang, Yuliang Li, Xuebo Liu, Junyi Zou, Qiang Wang, and Xiaowen Chu. 2025. Unrealllm: Towards highly controllable and interactable 3d scene generation by llm-powered procedural content generation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 19417–19435.

Fan-Yun Sun, Shengguang Wu, Christian Jacobsen, Thomas Yim, Haoming Zou, Alex Zook, Shangru Li, Yu-Hsin Chou, Ethem Can, Xunlei Wu, and 1 others. 2025a. 3d-generalist: Self-improving vision-language-action models for crafting 3d worlds. *arXiv preprint arXiv:2507.06484*.

Weilin Sun, Xinran Li, Manyi Li, Kai Xu, Xiangxu Meng, and Lei Meng. 2025b. Hierarchically-structured open-vocabulary indoor scene synthesis with pre-trained large language model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7122–7130.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, and 1 others. 2025. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*.

Xiping Wang, Yuxi Wang, Mengqi Zhou, Junsong Fan, and Zhaoxiang Zhang. 2025a. Hlg: Comprehensive 3d room construction via hierarchical layout generation. *arXiv preprint arXiv:2508.17832*.

Yuxin Wang, Lei Ke, Boqiang Zhang, Tianyuan Qu, Hanxun Yu, Zhenpeng Huang, Meng Yu, Dan Xu, and Dong Yu. 2025b. N3d-vlm: Native 3d grounding enables accurate spatial reasoning in vision-language models. *arXiv preprint arXiv:2512.16561*.

Chong Xia, Shengjun Zhang, Fangfu Liu, Chang Liu, Khodchaphun Hirunyaratsameewong, and Yueqi Duan. 2025. Scenepainter: Semantically consistent perpetual 3d scene generation with concept relation alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 28808–28817.

Fan Yang, Sicheng Zhao, Yanhao Zhang, Hui Chen, Haonan Lu, Jungong Han, and Guiguang Ding. 2024. Llmi3d: Mllm-based 3d perception from a single 2d image. *arXiv preprint arXiv:2408.07422*.

Yandan Yang, Baoxiong Jia, Shujie Zhang, and Siyuan Huang. 2025a. Sceneweaver: All-in-one 3d scene synthesis with an extensible and self-reflective agent. *arXiv preprint arXiv:2509.20414*.

Yandan Yang, Baoxiong Jia, Shujie Zhang, and Siyuan Huang. 2025b. Sceneweaver: All-in-one 3d scene synthesis with an extensible and self-reflective agent. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Yuanbo Yang, Yujun Shen, Yue Wang, Andreas Geiger, and Yiyi Liao. 2025c. Urbangen: Urban generation with compositional and controllable neural fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Zhifei Yang, Keyang Lu, Chao Zhang, Jiaxing Qi, Hanqi Jiang, Ruifei Ma, Shenglin Yin, Yifan Xu, Mingzhe Xing, Zhen Xiao, and 1 others. 2025d. Mmgdreamer: Mixed-modality graph for geometry-controllable 3d indoor scene generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 9391–9399.

Xingbo Yao, Xuanmin Wang, Hao Wu, Chengliang Ping, Doudou Zhang, and Hui Xiong. 2025a. Magiccity: Geometry-aware 3d city generation from satellite imagery with multi-view consistency. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 25325–25334.

Xingbo Yao, Xuanmin Wang, and Hui Xiong. 2025b. Citysculpt: 3d city generation from satellite imagery with uv diffusion. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 9813–9821.

Hong-Xing Yu, Haoyi Duan, Charles Herrmann, William T Freeman, and Jiajun Wu. 2025. Wonderworld: Interactive 3d scene generation from a single image. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5916–5926.

Yunzhi Zhang, Zizhang Li, Matt Zhou, Shangzhe Wu, and Jiajun Wu. 2025. The scene language: Representing scenes with programs, words, and embeddings. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24625–24634.