LEARNING TO CODE THE WORLD: HARNESS EVOLUTION FOR COMPOSITIONAL 3D RECONSTRUCTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Reconstructing editable and compositional 3D scenes from a real image requires jointly reasoning about geometry, object structure, and spatial relationships. We introduce SceneEvolve, a framework that represents 3D scenes as executable code and improves reconstruction by evolving the harness around a small open-source agent. SceneEvolve combines SceneAct, a compositional semantic action interface built on parameterized primitives, hierarchical composition, and semantic spatial relations, with SceneHarness-RSI, which leverages a stronger teacher agent to iteratively evolve the harness using feedback from development-set reconstructions. Throughout harness evolution, the model parameters, action semantics, and execution backend remain fixed. The resulting harness provides a reusable reconstruction procedure that enables the small open-source model to reconstruct unseen scenes without parameter updates or teacher assistance. Across two in-domain datasets and one out-of-domain dataset, SceneEvolve with the comparatively compact Qwen3.8-27B model improves all four evaluated reconstruction metrics while reducing mean inference time by 47.0% relative to the base model. More notably, despite relying on a substantially smaller model, SceneEvolve achieves 32.3% higher reconstruction quality than the large proprietary GPT-5.6-Sol model, while reducing mean equivalent deployment cost by 59.8%.

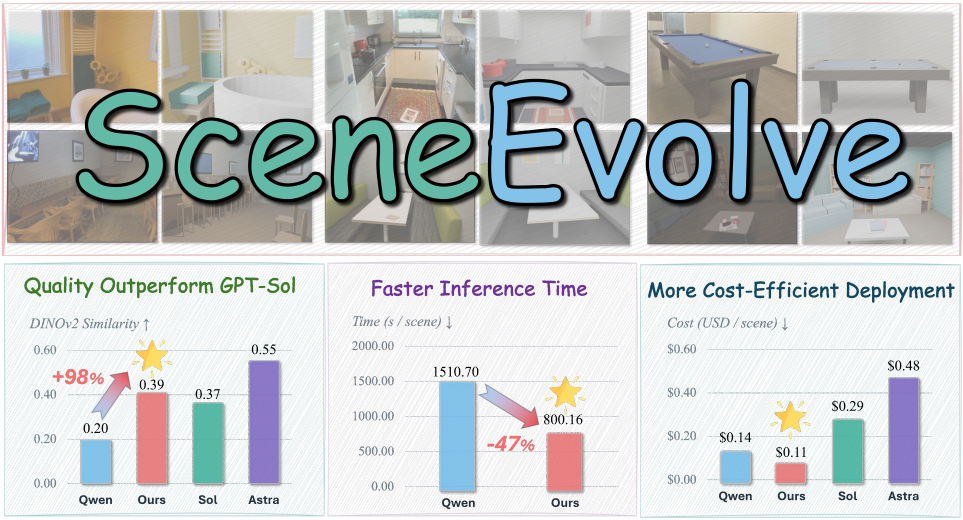

Figure 1: Our proposed SceneEvolve framework evolves the harness of a compact open-source code agent, enabling it to translate a single image into executable code that constructs an editable, compositional 3D scene. It improves reconstruction quality and reduces inference time compared with the base agent, while achieving higher reconstruction quality at lower cost than GPT-5.6-Sol.

1 INTRODUCTION

Compositional 3D scene reconstruction enables interactive and editable representations of real-world environments, with applications in robotics, embodied AI, AR/VR, and digital content cre-

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.

ation (Ma et al., 2026; Wang et al., 2026; Yeh et al., 2022). Given a single image of an indoor environment, the goal is to reconstruct an editable, compositional 3D scene that captures its geometry, constituent objects, spatial layout, and lighting (Ma et al., 2026; Yeh et al., 2022).

Existing approaches address this problem through several strategies. Procedural and generative methods synthesize diverse 3D environments (Raistrick et al., 2024; Yang et al., 2024), while multi-stage reconstruction pipelines divide the task into scene understanding, asset generation, and object placement (Ma et al., 2026). More recently, code-based agents have represented scenes as executable programs, enabling explicit manipulation of objects and spatial relations (Yin et al., 2026; Wang et al., 2026). Despite these advances, reconstructing a specific real-world scene with high fidelity often requires carefully engineered intermediate stages (Ma et al., 2026; Yeh et al., 2022) or repeated render-and-refine iterations (Yin et al., 2026; Huang et al., 2024).

Frontier models such as GPT-6-Astra (OpenAI, 2026b) suggest a more direct approach: generating editable 3D scene programs from a single image with limited iterative refinement. However, their deployment costs motivate us to investigate whether a substantially smaller open-source model can achieve comparable image-to-3D programming capabilities through harness design rather than parameter updates (Khattab et al., 2024; Agrawal et al., 2026). Our goal is to enable direct scene-code generation while reducing reliance on costly render-and-refine loops.

Achieving this goal presents two main challenges. First, compact models have limited visual and spatial reasoning capabilities, making it difficult to infer coherent 3D structures and translate visual observations into executable scene representations. Second, effective reconstruction also depends on the external harness: the instructions, tools, and control logic that govern how the model observes, plans, constructs, and executes a scene program (Yin et al., 2026; Khattab et al., 2024). Addressing these challenges requires both an interface that reduces the demands on geometric reasoning and a reusable procedure that guides reconstruction.

To address these challenges, we introduce **SceneEvolve**, as shown in Figure , a framework that combines a compositional scene interface, SceneAct, with a teacher-guided harness evolution framework, SceneHarness-RSI. SceneAct translates high-level scene intent into executable operations through parameterized geometric primitives, hierarchical object composition, and semantic spatial relations. These abstractions reduce the need for low-level geometric programming while preserving explicit scene structure. Building on this interface, SceneHarness-RSI uses a stronger teacher model to analyze the frozen student's development-set reconstructions, execution traces, and evaluation feedback, then propose revisions to the shared reconstruction harness. Candidate harnesses are evaluated under a common execution budget, and revisions are retained when they improve the development score. Throughout this process, the student parameters, action semantics, and execution backend remain fixed. The optimization target is the reusable reconstruction procedure, which is subsequently frozen and deployed on unseen scenes without teacher assistance.

Our contributions are summarized as follows:

(1) We formulate compositional 3D scene reconstruction as an executable harness-optimization problem, targeting direct scene-code generation without parameter updates or repeated iterations.

(2) We introduce SceneAct, a compositional interface that maps scene intent to executable 3D scenes using parameterized primitives, hierarchical composition, and semantic spatial relations.

(3) We propose SceneHarness-RSI, a teacher-guided recursive evolution method that optimizes reusable reconstruction harnesses for deployment on unseen scenes without teacher assistance.

(4) Across both in- and out-of-domain datasets, SceneEvolve with Qwen3.8-27B outperforms GPT-5.6-Sol *none*, while reducing cost by 59.8% and inference time by 47.0% relative to baseline.

2 Related Work

3D Generation for Compositional 3D Scene Reconstruction. Single-image reconstruction and generative models recover object geometry and appearance (Hong et al., 2024; Xu et al., 2024; Xiang et al., 2025), while scene-level methods model object layouts and relations using semantic priors (Lin & Mu, 2024; Lin et al., 2025) or jointly predict multiple objects and their spatial con-

figurations (Huang et al., 2025; Meng et al., 2026). SceneAct unifies object construction and scene composition in an executable program with primitives and spatial operators.

Multi-stage Pipelines for Compositional 3D Scene Reconstruction. Compositional reconstruction represents object geometry and scene layout as separately editable components. Prior work aligns CAD models or jointly estimates layout, poses, and geometry (Izadinia et al., 2017; Nie et al., 2020), with inverse rendering further recovering materials and lighting (Yeh et al., 2022). Recent methods combine single-image scene analysis, object generation, and spatial refinement (Ardelean et al., 2025; Ma et al., 2026; Zhang et al., 2026), leverage multi-view evidence for assembly (Xia et al., 2026; Qin et al., 2026), or jointly generate object geometry and spatial configurations (Huang et al., 2025; Meng et al., 2026). SceneEvolve instead realizes object-centric reconstruction as an executable scene program built from parameterized primitives, part hierarchies, and spatial relations, enabling a frozen agent to construct and edit the scene.

Programmatic Coding Agents for Compositional 3D Scene Reconstruction. Procedural and language-guided generators use reusable assets and spatial constraints for controllable scene synthesis (Deitke et al., 2022; Raistrick et al., 2024; Yang et al., 2024). Related representations encode hierarchical assemblies as executable programs (Jones et al., 2020) or predict structured scene descriptions from visual or point-cloud observations (Avetisyan et al., 2024; Mao et al., 2025). Visual coding agents further generate and refine graphics programs through rendered feedback, reusable libraries, and execution-guided iteration (Hu et al., 2024; Huang et al., 2024; Yin et al., 2026; Yang et al., 2026; Wang et al., 2026). SceneAct exposes a fixed set of semantic construction operators, while SceneHarness-RSI optimizes the reusable procedure.

3 METHODS

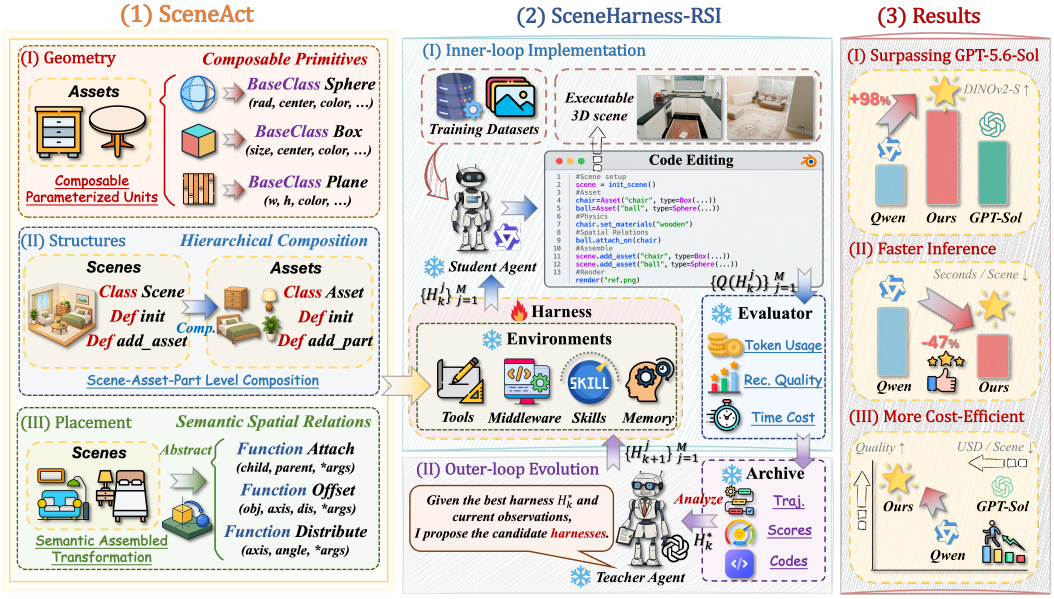

Figure 2: Illustration of the SceneEvolve framework for compositional image-to-3D reconstruction. The framework combines two components: (1) SceneAct translates scene intent into executable Blender operations through parameterized primitives, hierarchical composition, and semantic spatial relations. (2) SceneHarness-RSI couples an outer loop for teacher-guided harness evolution with an inner loop for candidate evaluation by a frozen student, retaining revisions that improve performance. The optimized harness is deployed on unseen scenes without parameter updates.

3.1 OVERVIEW

We propose SceneEvolve, a framework for compositional image-to-3D reconstruction of indoor scenes, as illustrated in Figure 2. The framework comprises two components: SceneAct, a compact

semantic interface that maps high-level scene intent to executable actions, and SceneHarness-RSI, a teacher-guided procedure that ens the reconstruction harness using end-to-end reconstruction feedback. SceneAct defines the executable interface through which the frozen student agent reconstructs 3D scenes. SceneHarness-RSI iteratively refines the harness through a propose–execute–evaluate–retain cycle to improve how the student uses SceneAct while keeping the student parameters fixed.

3.2 SceneAct: A Compact Semantic Interface from Intent to Actions

SceneAct provides a compact semantic interface between high-level scene reasoning and low-level Blender execution. It enables the student agent to use semantically meaningful actions that are deterministically translated into executable Blender operations, reducing the need to directly specify low-level geometric operations and numerical parameters. The action space comprises three complementary abstractions: *composable parameterized primitives*, *hierarchical composition*, and *semantic spatial relations*. Together, these abstractions support progressive scene construction from local geometry to object structure and scene-level layout.

Composable Parameterized Primitives. SceneAct exposes a compact set of parameterized geometric primitives as the basic units of scene construction. A SceneAct program can instantiate multiple primitives, with each primitive ρ_i represented as

$$\rho_i = \big(\kappa_i, \mathbf{g}_i, \mathbf{a}_i\big), \qquad \kappa_i \in \mathcal{G}, \tag{1}$$

where κ_i identifies a primitive class in the predefined library $\mathcal{G}$, and $\mathbf{g}_i$ and $\mathbf{a}_i$ denote its geometric and appearance parameters, respectively. This representation allows the student agent to construct geometry by instantiating and configuring primitive classes without directly manipulating mesh vertices, faces, or other low-level Blender representations.

The primitive classes share a common interface for translation, rotation, and scaling, allowing their instances to be composed into higher-level scene structures. For example, a tabletop can be represented by a thin *Box*, its supporting legs by *Box* or *Cylinder* instances, and a curved lamp arm by a *Curve*. These primitives can then be transformed and assembled into semantic parts and assets. This design expresses reconstruction as a sequence of compact, executable decisions and shifts the student's reasoning from low-level mesh programming to structured geometric construction. In this way, the student's geometric reasoning is reduced to two interpretable decisions: *which primitive best explains each visible part, and what geometric and appearance parameters characterize it?*

Hierarchical Composition. We organize the reconstructed scene using an explicit semantic hierarchy implemented in the scene program. The *Scene* class manages a collection of assets through interfaces such as *scene.add_asset(...)*. The reusable *Asset* base class encapsulates object-level structure and provides *asset.add_part(...)* for composing semantic parts. Each *Part* is constructed from one or more parameterized geometric primitives. This hierarchy separates scene-level composition, object-level structure, and part-level geometry: the *Scene* supports adding, removing, transforming, and duplicating complete assets, while each *Asset* controls the construction and refinement of its internal parts. For example, a table asset may contain a tabletop and several legs, each constructed independently from geometric primitives. This program structure supports modular construction and localized editing while preserving a direct correspondence between visually identifiable entities in the input image and their executable representations.

Semantic Spatial Relations. SceneAct specifies spatial structure through semantic relations, reducing the need to predict explicit coordinates. This design is motivated by the observation that VLMs are more reliable at inferring relations such as *under*, *attached to*, and *facing* than at predicting the precise 3D translations and rotations needed to realize them. Accordingly, each scene element exposes geometric features with semantic meaning, including *center*, *extent*, *support*, *anchor*, and *boundary*, together with relational operators such as *align(...)* and *distribute(...)*.

These operators encode spatial constraints relative to scene geometry. For example, four table legs can be positioned below a tabletop and aligned with its corners without predicting their coordinates independently. The SceneAct interpreter resolves such relations into deterministic transformations based on current geometry, reference frames, extents, and anchors. Since the constraints are defined structurally rather than in fixed world coordinates, they can be re-evaluated after edits, allowing

dependent elements to update automatically. This separates relational reasoning from numerical transformation computation, reducing direct coordinate prediction while preserving precision.

3.3 SCENEHARNESS-RSI: A TEACHER-GUIDED RECURSIVE HARNESS EVOLUTION

SceneHarness-RSI improves the execution harness of a frozen student through teacher-guided revisions. Starting from an initial harness H_0, the student reconstructs development-set scenes through SceneAct, producing Blender programs, rendered outputs, and execution traces. A teacher uses this evidence to identify limitations of the current harness and propose targeted revisions. The same frozen student then evaluates the revised harnesses under a fixed protocol, and only revisions that improve the recorded development-set reconstruction score are retained. Repeating this propose–execute–evaluate–retain cycle defines an outer-loop search over executable harnesses.

Harness Search Space and Evaluation. The harness defines the execution environment of the frozen student, including callable tool interfaces, middleware, reusable skills, and memory. SceneHarness-RSI searches over revisions to these components while keeping the SceneAct semantics, student parameters, and Blender execution backend fixed.

Let $\mathcal{D}_{\mathrm{dev}} = \{I_i\}_{i=1}^{N}$ denote a development set of N images. For each image I_i, the student executes under harness H to produce a reconstructed scene and its rendered image $\hat{I}_i(H)$. We define the development-set quality score $Q(H)$ as the average of three measures across development images:

$$Q(H) = \frac{1}{3N} \sum_{i=1}^{N} \left[\mathcal{S}\left(I_i, \hat{I}_i(H)\right) + \left(1 - \mathcal{L}\left(I_i, \hat{I}_i(H)\right)\right) + \frac{1 + \mathcal{D}\left(I_i, \hat{I}_i(H)\right)}{2} \right], \tag{2}$$

where $\mathcal{S}$, $\mathcal{L}$, and $\mathcal{D}$ denote SSIM (Wang et al., 2004), LPIPS (Zhang et al., 2018), and DINOv2 cosine similarity (Oquab et al., 2024), respectively. Before aggregation, we clip $\mathcal{S}$ and $\mathcal{L}$ to $[0, 1]$ and $\mathcal{D}$ to $[-1, 1]$. The resulting transformations place all three terms on a common $[0, 1]$ scale. Thus, a higher score $Q(H) \in [0, 1]$ indicates better reconstruction quality. We use this score to compare and select harness candidates.

Outer-Loop Teacher-Guided Harness Evolution. We search the discrete space of executable harnesses using teacher-guided proposals. At round k, the search maintains the retained harness H_k and an archive $\mathcal{A}_k$ containing all previously evaluated harnesses, their recorded development scores, and their execution traces. The teacher examines this history to identify recurring reconstruction and execution failures and proposes M candidate revisions:

$$H_{k+1}^{j} \sim P_{\phi}(\cdot \mid H_k, \mathcal{A}_k), \qquad j = 1, \ldots, M. \tag{3}$$

Here, P_ϕ denotes the proposal distribution induced by the teacher model with parameters ϕ. Revisions may include improved diagnostics, changes to tool handling, and reusable routines that apply SceneAct operators. These revisions change how the student performs reconstruction while preserving the operator semantics and student parameters.

Inner-Loop Student Evaluation and Harness Retention. The same frozen student evaluates each candidate harness on $\mathcal{D}_{\mathrm{dev}}$ under identical decoding configurations, rollout budgets, rendering settings, and scoring protocols. Let H_{k+1}^{*} denote a highest-scoring candidate in the current round. We replace the retained harness only when this candidate achieves a score strictly higher than the retained harness's recorded development score:

$$H_{k+1} = \begin{cases} H_{k+1}^{*}, & Q\big(H_{k+1}^{*}\big) > Q(H_k), \\ H_k, & \text{otherwise.} \end{cases} \tag{4}$$

All evaluated candidates, their scores, and their execution traces are added to the archive, allowing subsequent proposals to draw on both successful and unsuccessful revisions. This retention rule ensures that the recorded development scores of the retained harnesses form a non-decreasing sequence. It does not guarantee monotonic improvement in expected performance or performance on unseen scenes. Student reconstruction forms the inner execution loop, while teacher-guided harness revision forms the outer search loop. After K rounds, the method returns the retained harness H_K.

Table 1: Image-to-3D reconstruction on ScanNet and CA-1M (in-domain) and ARKitScenes (out-of-domain). In model rows, **bold** denotes column-best scores, violet shading highlights SceneEvolve, and Astra scores are shown in gray for reference only and excluded from ranking. The final two rows compare SceneEvolve with base Qwen3.8-27B and GPT-5.6-Sol *none*, the proprietary reference used in the main text. Green/Red denotes improvement/degradation.

Model	Inference effort	In-domain								Out-of-domain			
		ScanNet				CA-1M				ARKitScenes			
		PSNR↑	SSIM↑	LPIPS↓	DINOv2-S↑	PSNR↑	SSIM↑	LPIPS↓	DINOv2-S↑	PSNR↑	SSIM↑	LPIPS↓	DINOv2-S↑
GPT-6-Astra	High	10.063	0.386	0.684	0.590	10.309	0.382	0.705	0.556	9.561	0.407	0.748	0.519
GPT-6-Astra	Medium	10.143	0.383	0.684	0.579	10.433	0.378	0.711	0.557	10.177	0.423	0.732	0.528
GPT-6-Astra	Low	9.888	0.385	0.701	0.596	10.108	0.372	0.691	0.552	10.245	0.403	0.712	0.475
Claude-Opus-5	High	**9.174**	0.352	**0.735**	0.438	**8.781**	0.368	0.749	**0.441**	8.551	0.389	0.751	**0.417**
Claude-Opus-5	Medium	9.021	0.348	0.741	**0.444**	8.705	0.361	0.756	0.432	8.498	0.392	0.754	0.409
Claude-Opus-5	Low	8.742	0.337	0.747	0.416	8.566	0.351	0.753	0.419	8.287	0.371	0.764	0.391
Claude-Opus-5	None	8.463	0.329	0.759	0.398	8.318	0.341	0.769	0.423	8.146	0.358	0.771	0.356
GPT-5.6-Sol	High	9.042	0.337	0.742	0.417	8.562	0.351	0.748	0.428	8.312	0.371	0.755	0.401
GPT-5.6-Sol	Medium	8.731	0.319	0.754	0.375	7.821	0.347	0.765	0.411	8.027	0.340	0.752	0.408
GPT-5.6-Sol	Low	7.945	0.307	0.756	0.382	8.294	0.338	0.757	0.416	7.942	0.314	0.776	0.349
GPT-5.6-Sol	None	7.755	0.284	0.752	0.389	8.212	0.321	0.770	0.361	8.067	0.356	0.773	0.357
Gemini-Flash-3.8	High	8.126	0.318	0.774	0.358	8.113	0.324	0.781	0.365	7.984	0.337	0.785	0.341
Gemini-Flash-3.8	Medium	8.004	0.311	0.782	0.361	8.154	0.319	0.779	0.358	7.846	0.330	0.792	0.334
Gemini-Flash-3.8	Low	7.813	0.298	0.789	0.337	7.874	0.306	0.795	0.341	7.731	0.332	0.799	0.336
Qwen3.8-27B	xHigh	9.061	0.376	0.772	0.240	8.381	0.367	0.806	0.134	9.704	0.448	0.785	0.219
Qwen3.8-27B+Ours	xHigh	9.124	**0.378**	0.755	0.391	8.513	**0.440**	**0.742**	0.418	**9.862**	**0.453**	**0.721**	0.368
Δ vs. base Qwen3.8-27B xHigh		**+0.063**	**+0.5%**	**-2.2%**	**+62.9%**	**+0.132**	**+19.9%**	**-7.9%**	**+211.9%**	**+0.158**	**+1.1%**	**-8.2%**	**+68.0%**
Δ vs. GPT-5.6-Sol None		**+1.369**	**+33.1%**	+0.4%	**+0.5%**	**+0.301**	**+37.1%**	**-3.6%**	**+15.8%**	**+1.795**	**+27.2%**	**-6.7%**	**+3.1%**

4 EXPERIMENTS

4.1 EXPERIMENTAL SETTINGS

Datasets. We use three indoor scene datasets comprising 160 scenes in total: ScanNet (Dai et al., 2017) and CA-1M (Lazarow et al., 2025) for in-domain development and evaluation, and ARKitScenes (Baruch et al., 2021) for out-of-domain evaluation. The resulting split consists of 100 development scenes, 40 in-domain test scenes, and 20 out-of-domain test scenes.

In-domain Datasets. ScanNet (Dai et al., 2017) provides RGB-D sequences with camera poses, reconstructed indoor geometry, and semantic annotations, while CA-1M (Lazarow et al., 2025) contains handheld egocentric captures registered to high-quality laser scans with dense 3D object annotations. From each dataset, we randomly sample 70 scenes and partition them at the scene level into disjoint development and test sets of 50 and 20 scenes, respectively. All views and repeated captures of the same physical scene are confined to a single split. The development sets are used for recursive harness search, while the test sets are reserved exclusively for in-domain evaluation.

Out-of-domain Dataset. ARKitScenes (Baruch et al., 2021) contains mobile RGB-D indoor captures with laser-scanned geometry and 3D object annotations. We randomly sample 20 scenes for out-of-domain evaluation. The harness developed on ScanNet (Dai et al., 2017) and CA-1M (Lazarow et al., 2025) is transferred directly to these scenes with the student weights, harness code, and inference configuration fixed. No ARKitScenes (Baruch et al., 2021) data or feedback are used during harness development, enabling a clean evaluation of cross-dataset generalization.

Evaluation Metrics. We evaluate all methods along three dimensions: reconstruction quality, inference efficiency, and equivalent cost. Reconstruction quality is measured by PSNR (Wang & Bovik, 2009), SSIM (Wang et al., 2004), LPIPS (Zhang et al., 2018), and DINOv2 similarity (Oquab et al., 2024), capturing pixel, structural, perceptual, and semantic fidelity, respectively. Inference efficiency is measured by the average per-scene inference time under a unified protocol. We further report the average equivalent cost per scene under a consistent cost definition, enabling direct comparison of the trade-offs among quality, efficiency, and cost across model configurations.

Baselines. We benchmark proprietary multimodal models as zero-shot agents for compositional image-to-3D scene reconstruction, without task-specific fine-tuning. Our baseline suite includes GPT-5.6-Sol (OpenAI, 2026a) and GPT-6-Astra (OpenAI, 2026b), Claude-Opus-5 (Anthropic, 2026), and Gemini-Flash-3.8 (Google DeepMind, 2026). For models with configurable reasoning

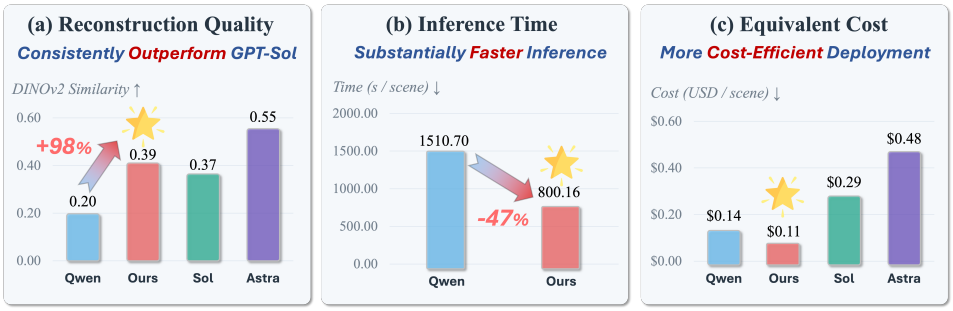

Figure 3: Comparison of SceneEvolve with Qwen3.8-27B at *xhigh*, GPT-5.6-Sol at *none*, and GPT-6-Astra at *none*. The average DINOv2-S ↑, inference time ↓, and equivalent cost ↓ are reported.

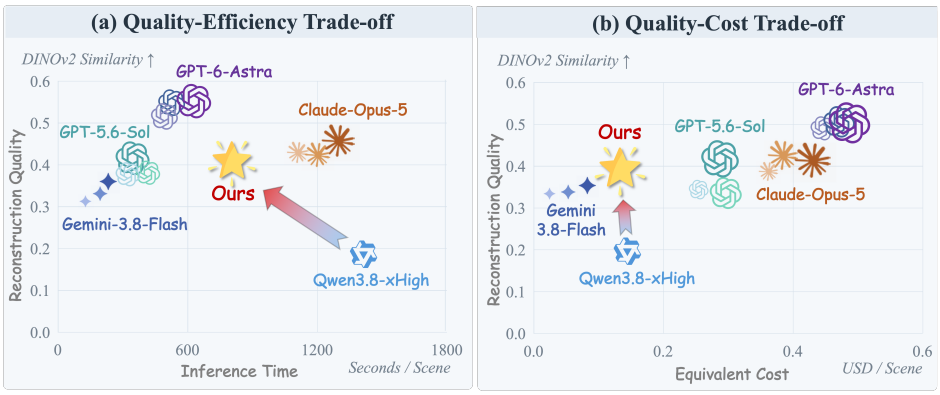

Figure 4: Comparison of reconstruction quality against inference time and equivalent deployment cost across model configurations. The *high*, *medium* and *low* settings denote reasoning modes from deeper to shallower inference. DINOv2-S ↑, inference time ↓, and equivalent deployment cost ↓ are reported averaged over three datasets.

effort, we evaluate all supported settings among *none*, *low*, *medium*, and *high*, treating each model–effort pair as a distinct baseline configuration. We also evaluate Qwen3.8-27B (Qwen Team, 2026) at *xhigh*, using the same model and reasoning configuration as the frozen student in our method.

Implementation details. We use Qwen3.8-27B (Qwen Team, 2026) at *xhigh* as the frozen student and GPT-6-Astra (OpenAI, 2026b) at *medium* as the harness proposer, running $K = 10$ rounds with $M = 10$ proposals each. Candidates are conditioned on archived development evidence and evaluated on equally weighted ScanNet (Dai et al., 2017) and CA-1M (Lazarow et al., 2025) scenes, with input frames selected by space-aware frame sampling strategy following Spatial-MLLM (Wu et al., 2025). Each episode is capped at 20 model calls, 12 renders, and 2,400 seconds, with fixed 512×512 rendering settings. The selected harness and inference configuration are frozen at test.

4.2 Quantitative Results

4.2.1 Comparison with the Base Model

Reconstruction Quality. As shown in Table 1, SceneEvolve consistently outperforms Qwen3.8-27B across all four reconstruction metrics and three datasets under fixed student weights and reasoning effort. Averaged across datasets, DINOv2-S increases from 0.198 to 0.392, while SSIM improves by 6.7%, LPIPS decreases by 6.1%, and PSNR increases by 0.118 dB; these gains also transfer to the unseen ARKitScenes dataset without target-specific adaptation.

Inference Time. As Figure 4 and Figure 5 illustrated, SceneEvolve reduces mean per-scene inference time by 37.5% on ScanNet, 55.1% on CA-1M, and 46.8% on ARKitScenes relative to the base Qwen3.8-27B *xhigh*. Across the three datasets, mean inference time decreases from 1,510.7 to 800.2 seconds, a reduction of 47.0%, alongside improvements in all four reconstruction metrics.

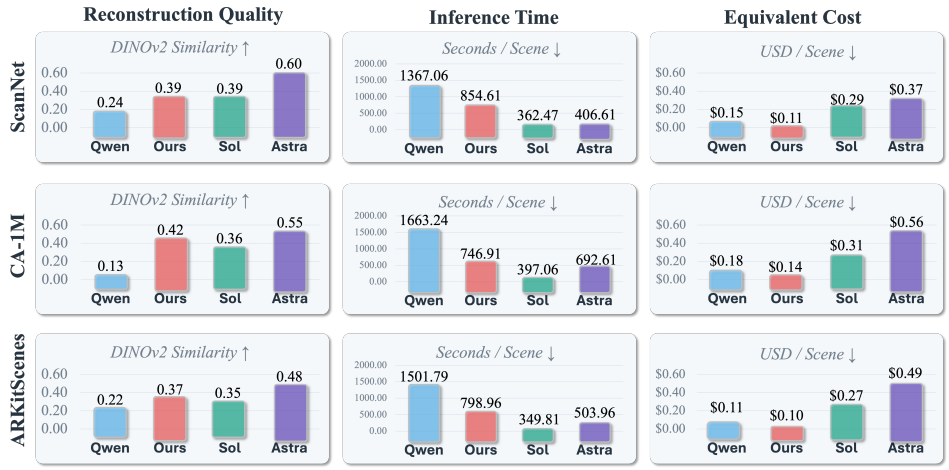

Figure 5: Comparison of SceneEvolve with Qwen3.8-27B at *xhigh*, GPT-5.6-Sol at *none*, and GPT-6-Astra at *none* on the two in-domain datasets ScanNet and CA-1M and the out-of-domain dataset ARKitScenes. The DINOv2 similarity, inference time, and deployment cost are reported for each.

Deployment cost. As shown in Figure 4 and Figure 5, SceneEvolve also achieves lower deployment costs across datasets, reducing the mean equivalent deployment cost by 20.5% relative to the Qwen3.8-27B xHigh baseline and by 22.7% on both ScanNet and CA-1M. Overall, SceneEvolve provides a more favorable trade-off between reconstruction quality, inference time, and cost.

4.2.2 Comparison with Frontier Models

Reconstruction Quality. SceneEvolve outperforms GPT-5.6-Sol with *none* reasoning effort (Sol-none) in eleven of twelve dataset–metric comparisons, improving mean SSIM by 32.3%, reducing LPIPS by 3.4%, increasing PSNR by 1.155 dB and increasing its DINOv2-S by 6.3%. All four metrics improve on CA-1M and ARKitScenes. Mean SSIM also exceeds Sol-low, Sol-medium, and Sol-high by 32.5%, 26.3%, and 20.0%, respectively. Against Astra-low and Astra-medium, mean SSIM improves by 9.6% and 7.3%, although DINOv2-S remains 27.5% and 29.3% lower.

Inference Time. SceneEvolve averages 800.2 seconds per scene, compared with 369.8 for Sol-none, 502.3 for Astra-low, and 526.8 for Astra-medium. It therefore requires 116.4%, 59.3%, and 51.9% more time, respectively; its inference speedup is relative to base Qwen3.8-27B.

Deployment Cost. Mean equivalent deployment cost decreases by 59.8% relative to Sol-none, from approximately $0.290 to $0.117 per scene. Costs are also 46.2%, 52.9%, 69.0%, and 70.5% lower than Sol-low, Sol-medium, Astra-low, and Astra-medium, respectively. Excluding offline harness search, SceneEvolve thus offers higher structural similarity at lower deployment cost.

Table 2: Component-ablation on Qwen3.8-27B, averaged across the three datasets. PSNR, SSIM, LPIPS, DINOv2 similarity, inference time, and equivalent cost are averaged across the three datasets, with time and cost measured per scene.

Configuration	PSNR↑	SSIM↑	LPIPS↓	DINOv2-S↑	Time↓	Cost↓
Base Qwen3.8-27B	9.049	0.397	0.788	0.198	1510.7	0.147
w/o SceneAct	9.102	0.405	0.771	0.305	1035.0	0.152
w/o SceneHarness-RSI	9.119	0.411	0.762	0.285	1424.0	0.155
SceneEvolve	**9.166**	**0.424**	**0.739**	**0.392**	**800.2**	**0.117**

4.3 Qualitative Results

Comparison with the Base Model. Figure 6 presents reference images and reconstructed scenes for visual comparison. The comparison focuses on object identity, relative proportions, room layout, and support relations, complementing the semantic and structural gains over base Qwen. These aspects directly reflect the object hierarchy and relation-based placement provided by SceneAct.



Figure 6: Qualitative comparisons of indoor scene reconstruction with GPT-6-Astra *none*, GPT-5.6-Sol *none*, and Qwen3.8-27B *xhigh* on Scannet, CA-1M and ARKitScenes datasets. SceneEvolve reconstructs editable 3D scenes that capture object geometry, relative proportions, and spatial relationships in the reference images, demonstrating improved structural fidelity.

Comparison with Frontier Models. The frontier comparisons emphasize global composition, large-object silhouettes, and local part structure. Quantitatively, SceneEvolve achieves higher mean SSIM while nearly matching Sol-none's DINOv2-S, suggesting competitive structural fidelity.

4.4 Ablation Study

We ablate SceneAct and SceneHarness-RSI with Qwen3.8-27B at *xhigh*. *w/o SceneAct* replaces SceneAct with direct Blender operations under the same optimization budget, while *w/o SceneHarness-RSI* keeps SceneAct but uses the initial, unevolved harness. The full method combines both.

Effect of SceneAct. As shown in Table 2, incorporating SceneAct increases DINOv2-S from 0.305 to 0.392 and reduces mean inference time by 22.7% compared with the *w/o SceneAct* variant. These results suggest that the structured interface, which supports hierarchical construction and semantic placement, improves both reconstruction quality and inference efficiency.

Effect of SceneHarness-RSI. As shown in Table 2, with SceneAct fixed, harness evolution increases DINOv2-S from 0.285 to 0.392, reduces mean inference time by 43.8%, and lowers the mean equivalent cost per scene from $0.155 to $0.117. These results support the value of optimizing a reusable reconstruction procedure alongside the structured action interface.

5 Conclusion

We introduced SceneEvolve, a framework for editable, compositional 3D scene reconstruction in code-generated scene settings that combines a structured scene action interface with teacher-guided evolution of a reusable reconstruction harness. Extensive experiments demonstrate that SceneEvolve improves over its base configuration and outperforms GPT-5.6-Sol with *none* reasoning effort, while reducing mean inference time by 47.0% and equivalent deployment cost per scene by 59.8%. These results demonstrate improved quality and efficiency for editable 3D scene generation.

AI USE STATEMENT

Generative AI tools were used solely to polish the manuscript's language, improving grammar, clarity, and concision. All scientific content and conclusions are the authors' own. The authors reviewed all AI-assisted edits and take full responsibility for the final manuscript.

REFERENCES

Lakshya A. Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J. Ryan, Meng Jiang, Christopher Potts, Koushik Sen, Alexandros G. Dimakis, Ion Stoica, Dan Klein, Matei Zaharia, and Omar Khattab. GEPA: Reflective prompt evolution can outperform reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2026.

Anthropic. Introducing Claude Opus 5. Official announcement, July 2026. URL `https://www.anthropic.com/news/claude-opus-5`. Published: 2026-07-24. Accessed: 2026-09-26.

Andreea Ardelean, Mert Özer, and Bernhard Egger. Gen3DSR: Generalizable 3D scene reconstruction via divide and conquer from a single view. In *International Conference on 3D Vision (3DV)*, pp. 616–626, 2025.

Armen Avetisyan, Christopher Xie, Henry Howard-Jenkins, Tsun-Yi Yang, Samir Aroudj, Suvam Patra, Fuyang Zhang, Duncan Frost, Luke Holland, Campbell Orme, Jakob Engel, Edward Miller, Richard Newcombe, and Vasileios Balntas. SceneScript: Reconstructing scenes with an autoregressive structured language model. In *European Conference on Computer Vision (ECCV)*, pp. 247–263, 2024.

Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, and Elad Shulman. ARKitScenes: A diverse real-world dataset for 3D indoor scene understanding using mobile RGB-D data. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021.

Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5828–5839, 2017.

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. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 5982–5994, 2022.

Google DeepMind. Gemini 3.8 Flash. Model card, September 2026. URL `https://deepmind.google/models/model-cards/gemini-3-8-flash/`. Published: 2026-09-02. Accessed: 2026-09-26.

Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. LRM: Large reconstruction model for single image to 3D. In *International Conference on Learning Representations (ICLR)*, 2024.

Ziniu Hu, Ahmet Iscen, Aashi Jain, Thomas Kipf, Yisong Yue, David A. Ross, Cordelia Schmid, and Alireza Fathi. SceneCraft: An LLM agent for synthesizing 3D scenes as Blender code. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 19252–19282, 2024.

Ian Huang, Guandao Yang, and Leonidas Guibas. BlenderAlchemy: Editing 3D graphics with vision-language models. In *European Conference on Computer Vision (ECCV)*, pp. 297–314, 2024.

Zehuan Huang, Yuan-Chen Guo, Xingqiao An, Yunhan Yang, Yangguang Li, Zi-Xin Zou, Ding Liang, Xihui Liu, Yan-Pei Cao, and Lu Sheng. MIDI: Multi-instance diffusion for single image to 3D scene generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23646–23657, 2025.

Hamid Izadinia, Qi Shan, and Steven M. Seitz. IM2CAD. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5134–5143, 2017.

R. Kenny Jones, Theresa Barton, Xianghao Xu, Kai Wang, Ellen Jiang, Paul Guerrero, Niloy J. Mitra, and Daniel Ritchie. ShapeAssembly: Learning to generate programs for 3D shape structure synthesis. *ACM Transactions on Graphics*, 39(6):1–20, 2020.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan A, Saiful Haq, Ashutosh Sharma, Thomas Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. DSPy: Compiling declarative language model calls into state-of-the-art pipelines. In *International Conference on Learning Representations (ICLR)*, 2024.

Justin Lazarow, David Griffiths, Gefen Kohavi, Francisco Crespo, and Afshin Dehghan. Cubify Anything: Scaling indoor 3D object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22225–22233, 2025.

Chenguo Lin and Yadong Mu. InstructScene: Instruction-driven 3D indoor scene synthesis with semantic graph prior. In *International Conference on Learning Representations (ICLR)*, 2024.

Chenguo Lin, Yuchen Lin, Panwang Pan, Xuanyang Zhang, and Yadong Mu. InstructLayout: Instruction-driven 2D and 3D layout synthesis with semantic graph prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(12):11040–11050, 2025.

Xiaoxuan Ma, Jiashun Wang, Nicolás Ugrinovic, Yehonathan Litman, and Kris Kitani. REST3D: Reconstructing physically stable 3D scenes from a single image. *arXiv preprint arXiv:2605.30338*, 2026.

Yongsen Mao, Junhao Zhong, Chuan Fang, Jia Zheng, Rui Tang, Hao Zhu, Ping Tan, and Zihan Zhou. SpatialLM: Training large language models for structured indoor modeling. *arXiv preprint arXiv:2506.07491*, 2025.

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

Yinyu Nie, Xiaoguang Han, Shihui Guo, Yujian Zheng, Jian Chang, and Jian Jun Zhang. 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 (CVPR)*, pp. 55–64, 2020.

OpenAI. GPT-5.6 Sol. OpenAI API documentation, 2026a. URL `https://developers.openai.com/api/docs/models/gpt-5.6-sol`. Accessed: 2026-09-26.

OpenAI. GPT-6 Astra. OpenAI API documentation, 2026b. URL `https://developers.openai.com/api/docs/models/gpt-6-astra`. Accessed: 2026-09-26.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Minghan Qin, Yuang Wang, Xiuyu Yang, Yushi Long, Yujian Zhang, Ruihuan Wang, Kai Ye, Yangang Zhang, and Hang Li. Lucida: Parse, generate, and place for composable real-to-sim scene modeling. *arXiv preprint arXiv:2608.30821*, 2026.

Qwen Team. Qwen3.8-27B. Hugging Face model card, 2026. URL `https://huggingface.co/Qwen/Qwen3.8-27B`. Accessed: 2026-09-26.

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

Puyi Wang, Yuhao Wang, Linjie Li, Zhengyuan Yang, Kevin Qinghong Lin, Yangguang Li, and Yu Cheng. SceneCode: Executable world programs for editable indoor scenes with articulated objects. *arXiv preprint arXiv:2605.19587*, 2026.

Zhou Wang and Alan C. Bovik. Mean squared error: Love it or leave it? a new look at signal fidelity measures. *IEEE Signal Processing Magazine*, 26(1):98–117, 2009.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004.

Diankun Wu, Fangfu Liu, Yi-Hsin Hung, and Yueqi Duan. Spatial-MLLM: Boosting MLLM capabilities in visual-based spatial intelligence. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, 2025.

Chong Xia, Kai Zhu, Zizhuo Wang, Fangfu Liu, Zhizheng Zhang, and Yueqi Duan. SimRecon: SimReady compositional scene reconstruction from real videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 42452–42463, 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21469–21480, 2025.

Jiale Xu, Weihao Cheng, Yiming Gao, Xintao Wang, Shenghua Gao, and Ying Shan. InstantMesh: Efficient 3D mesh generation from a single image with sparse-view large reconstruction models. *arXiv preprint arXiv:2404.07191*, 2024.

Yixuan Yang, Zhen Luo, Wanshui Gan, Jinkun Hao, Junru Lu, Jinghao Yan, Zhaoyang Lyu, and Xudong Xu. Code-as-Room: Generating 3D rooms from top-down view images via agentic code synthesis. *arXiv preprint arXiv:2605.18451*, 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, 2024.

Yu-Ying Yeh, Zhengqin Li, Yannick Hold-Geoffroy, Rui Zhu, Zexiang Xu, Miloš Hašan, Kalyan Sunkavalli, and Manmohan Chandraker. PhotoScene: Photorealistic material and lighting transfer for indoor scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18562–18571, 2022.

Shaofeng Yin, Jiaxin Ge, Zora Zhiruo Wang, Chenyang Wang, Xiuyu Li, Michael J. Black, Trevor Darrell, Angjoo Kanazawa, and Haiwen Feng. Vision-as-inverse-graphics agent via interleaved multimodal reasoning. In *European Conference on Computer Vision (ECCV)*, volume 17035 of *Lecture Notes in Computer Science*, pp. 557–575. Springer, 2026.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 586–595, 2018.

Shujie Zhang, Jingkun Yi, Weipeng Zhong, Zirui Zhou, Yangkun Zhu, Hanqing Wang, Xudong Xu, Weinan Zhang, and Chunhua Shen. RoboSnap: One-shot real-to-sim scene generation for generalizable robot learning and evaluation. *arXiv preprint arXiv:2607.06699*, 2026.