DISTSCENE: OBJECT-TO-SCENE DISTILLATION FOR 3D SCENE GENERATION

Anonymous authors
Paper under double-blind review



Figure 1: Diverse 3D scenes generated by DistScene from images.

ABSTRACT

We present DistScene, a framework for single-image compositional 3D scene generation by jointly modeling the environment and individual objects. Unlike existing methods that represent scenes primarily as collections of objects, we model the environment as an explicit scene component to provide geometric context for object placement. Specifically, we introduce Scene-Frame Generation, which jointly generates separate environment and object components in a shared coordinate frame, allowing their geometry and relative placement to be learned together. Then we introduce Object-Centric Refinement to refine each object in a local frame with scene context. Finally, we develop Object-to-Scene Distillation to transfer pretrained object-generation priors to scene generation through automatically composed and rendered synthetic scenes. Evaluations on indoor and outdoor benchmarks demonstrate improved scene-level geometric accuracy over the evaluated baselines.

1 INTRODUCTION

Compositional 3D scene generation is essential for turning images into structured 3D worlds by recovering environments and independent objects, enabling applications in autonomous driving Lu et al. (2025); Liu et al. (2026), virtual reality Martinez-Gonzalez et al. (2020), robotics Wang et al. (2024a), and simulation Wu et al. (2024b). Yet this task is challenging because object decomposition, spatial placement, and 3D geometry generation must be resolved jointly from a single image.

Existing approaches Chen et al. (2026b); Yin et al. (2026) mainly adopt a modular and cascaded paradigm that address these problems in separate stages, using segmentation Carion et al. (2026) or detection DeTone et al. (2026) for object decomposition, depth Wang et al. (2025b) or layout estimation Wang et al. (2025a) for spatial placement, and object-level 3D generators Xiang et al. (2026); Hunyuan3D et al. (2025) for geometry recovery. By leveraging specialized foundation models

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.

for these individual capabilities, such pipelines have substantially improved the generalization and quality of compositional scene reconstruction. However, this cascaded paradigm often propagates errors across different intermediate predictions, leading to incomplete object geometry and inaccurate spatial relationships.

To reduce this cascading dependency, recent feed-forward methods Huang et al. (2025); Lin et al. (2026) generate multiple 3D instances and their spatial arrangements within a single learned process. Although this formulation reduces the need for explicit stage-wise processing, existing feed-forward methods typically represent a scene as a collection of object instances without explicitly modeling the surrounding environment. Consequently, object arrangements are formed without explicit constraints from the environmental geometry, making it difficult to produce globally coherent scene layouts.

Based on these observations, we present DistScene, which directly generates a decomposable 3D scene from a single image by jointly generating the environment and multiple independent objects. Specifically, we first design a **Scene-Frame Generation** stage to represent the environment and objects as separate components and jointly generates them in a shared scene frame, preserving component decomposability and scene-level spatial coherence. In this way, the environment component can provide a shared spatial context for organizing object placement, relative scales and relationships.

However, this formulation introduces two challenges: objects often occupy a limited portion of the whole scene space and may lose fine geometric details; training such a scene generator requires large-scale 3D scenes with decomposed environment and object supervision. To address the first challenge, we introduce **Object-Centric Refinement**, which refines each object in a normalized local space conditioned on the generated scene context. The refined object is then placed back into its original scene location, thereby improving local geometric detail while preserving the scene layout and component relationships. To address the second challenge, we introduce **Object-to-Scene Distillation**, which procedurally assembles environments and objects generated by a pretrained object generator into 3D scenes. This process transfers object-level generative priors into scalable, component-aligned supervision for training scene generation. Extensive experiments across indoor and outdoor settings show that DistScene produces more accurate scene geometry and more coherent spatial layouts than existing approaches, while retaining higher-quality local object details. Our main contributions are:

- We present **DistScene**, a compositional 3D scene generation framework that jointly generates the environment and independent objects in the scene from a single image.
- We introduce **Object-Centric Refinement** to recover fine-grained object geometry in local spaces while preserving the scene layout.
- We propose **Object-to-Scene Distillation** to train our scene generator by automatically constructing scene supervision from a pretrained object generator.

2 Related Work

Compositional 3D Scene Generation. Single-image compositional 3D scene generation (Nie et al., 2020; Liu et al., 2022; Wu et al., 2024a) aims to recover multiple independent 3D components while preserving their spatial organization. Existing methods mainly follow two directions. Component-aligned reconstruction methods Tang et al. (2025); Zhou et al. (2024); Ardelean et al. (2025) recover individual objects from an image and construct a compositional scene, as exemplified by CAST (Yao et al., 2025) and SAM 3D (Chen et al., 2026a). These methods typically combine scene decomposition (Carion et al., 2026; Ren et al., 2024; Deng et al., 2026b; Ravi et al., 2025), spatial placement Mao et al. (2025); DeTone et al. (2026) or object pose estimation (Ardelean et al., 2025; Shi et al., 2026), and monocular or multi-view geometry estimation (Wang et al., 2025b; 2024b; 2025a; 2026b;a; Lin et al., 2025a; Yang et al., 2024b;a; Viola et al., 2025; Wang et al., 2026c; Zhang et al., 2026; Xu et al., 2026) to reconstruct the scene components, provising a reconstruction-based paradigm for compositional 3D scene generation. Feed-forward methods Lin et al. (2025b); Chen et al. (2025) instead generate multiple 3D instances and their spatial arrangements within a single learned process. MIDI (Huang et al., 2025) and SceneGen (Meng et al., 2026) follow this direction and reduce the need for explicit stage-wise scene construction. However, these methods remain primarily object-centric and do not explicitly model the scene environment as a generated component. Consequently, the spatial organization of objects is not directly grounded in the geometry of the

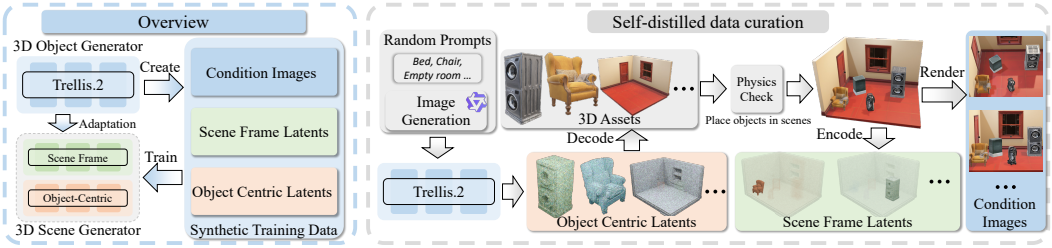

Figure 2: Overview of *Self-Distilled Data Curation*. We adopt Trellis.2 as the object generator from which training data are distilled. First, a VLM generates descriptions of 3D assets, including objects and environments; these descriptions are then used to generate the corresponding input images for Trellis.2. After the 3D assets are obtained, we place them into the generated environments with physics checks. Finally, we render condition images for each scene as conditioning inputs.

surrounding environment. In contrast, DistScene jointly generates the environment and multiple objects in a shared scene frame, and further refines each object in a local space while preserving its scene-level placement.

3D Object Generation. Recent image-to-3D methods have established strong generative priors for recovering object geometry and appearance from a single image. Methods based on Vecset representation (Lai et al., 2025; Li et al., 2024; 2025; Deng et al., 2026a; Yan et al., 2026; Zhang et al., 2024; Ye et al., 2025; Luo et al., 2026), and Structured 3D Latents (Xiang et al., 2025; 2026; Wu et al., 2026; He et al., 2025; Li et al., 2026) improve the quality and diversity of generated 3D assets. Part-aware methods such as PartCrafter (Lin et al., 2026) further represent an object as a set of structured components, enabling decomposable object generation. Despite these advances, existing object generators are primarily designed for individual objects in a canonical space. Their latent representations describe the geometry and appearance of each object but do not directly encode its position, scale, or spatial relationship within a surrounding environment. DistScene builds on these object-level priors and extends them to scene-frame generation, where the environment and individual objects are represented as separate components in a shared scene space.

3 Preliminary: Sparse-Voxel-Based 3D Generation

Sparse-voxel-based 3D generation represents an asset as active voxels on a regular grid, where each active voxel carries geometry and material features. A sparse VAE, typically built with sparse convolutional networks, compresses this representation into a compact latent space. On top of the latent space, a flow-matching generative model, typically instantiated as a sparse DiT, learns to synthesize assets conditioned on an image. Generation usually proceeds in stages: first predicting the sparse occupancy layout, then generating geometry latents within active voxels, and finally synthesizing appearance latents aligned with the geometry. Formally, this single-asset paradigm can be summarized as

$$\boldsymbol{f} = \{(f_i^{\text{shape}}, f_i^{\text{mat}}, p_i)\}_{i=1}^{L}, \quad z = \mathcal{E}(\boldsymbol{f}), \quad \hat{z} \sim \mathcal{G}(\cdot \mid I), \quad \hat{\boldsymbol{f}} = \mathcal{D}(\hat{z}), \tag{1}$$

where $\mathcal{E}$ and $\mathcal{D}$ denote the sparse VAE encoder and decoder, $\mathcal{G}$ is the image-conditioned flow-matching generator, and I is the input image. Our method follows this paradigm and extends it from a single asset to a complete compositional scene.

4 Method

Given a single image I, our goal is to generate a complete, decomposable compositional scene. We represent the scene in a unified sparse voxel space and jointly synthesize its objects and environment, followed by a scene-conditioned refinement stage that boosts the fidelity of each object. To overcome the scarcity of scene data, we further build a self-distilled data engine that infinitely produces physically plausible training scenes. Formally, the overall pipeline can be summarized as

$$S' = \mathcal{R}\big(\mathcal{G}(I)\big), \quad S = \mathcal{G}(I) = \{O_1, \ldots, O_N, E\}, \tag{2}$$

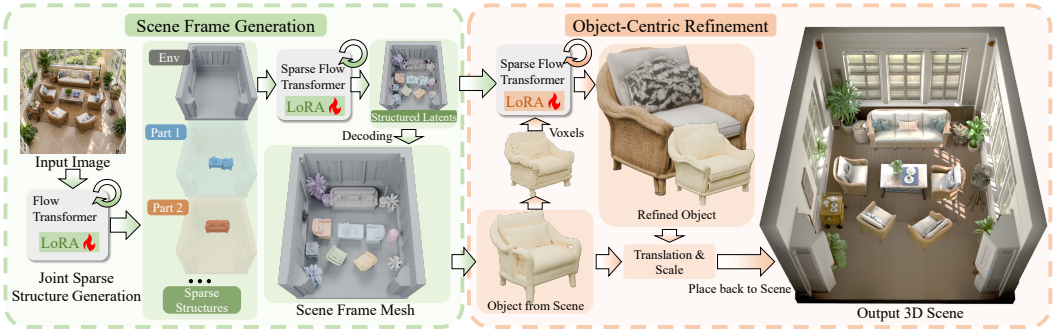

Figure 3: Overview of *Inference Pipeline*: Left: the environment and objects are jointly generated in a shared scene frame through two stages, sparse structure generation and geometry-latent generation. Right: each generated object is transformed to a local voxel support for scene-conditioned refinement and then placed back into the scene using the inverse transformation.

where $\mathcal{G}$ jointly generates the objects and the environment from I, and $\mathcal{R}$ refines each object to produce the final scene S'.

4.1 Self-Distilled Data Curation

Training a compositional scene generator requires decomposable scene data with complete meshes for every object. However, existing scene datasets such as 3D-FUTURE Fu et al. (2021) are limited in both scale and object diversity, which severely bottlenecks the generalization of scene generation models. To this end, we build a self-distilled data engine that can efficiently synthesize unlimited training scenes without relying on any existing scene dataset or human annotation.

Our engine generates a scene in three steps, as illustrated in Figure 2. First, an LLM produces a structured description of the scene, specifying the objects it contains and their approximate sizes. Second, an object generator synthesizes each object individually as a complete mesh. Crucially, the same generator also produces the environment, so the background is not an empty space but a geometrically detailed asset, eliminating the need for any separate background model Third, the generated objects are randomly placed into the generated environment, and the layout is iteratively adjusted by checking their contacts with the environment and collisions with each other. Formally, a scene $S = \{(O_i, T_i)\}_{i=1}^{N} \cup \{E\}$ is produced by

$$E = \mathcal{G}(d_E), \quad \{O_i\}_{i=1}^{N} = \{\mathcal{G}(d_i)\}_{i=1}^{N}, \quad \{T_i\}_{i=1}^{N} = \mathcal{A}\big(\{O_i\}, E\big), \quad \text{s.t.} \quad \mathcal{C}(S) = 1, \tag{3}$$

where d_i is the LLM-generated description of the i-th object, $\mathcal{G}$ is the object generator, $\mathcal{A}$ is the layout adjustment procedure, T_i denotes the pose of O_i, and $\mathcal{C}$ enforces physical plausibility, i.e., no inter-penetration and plausible contacts with the environment. For each generated scene, we render a large number of views and retain only those in which all objects are visible as conditioning images, so that the model learns a consistent image-to-scene mapping. By repeating this process, the engine can produce unlimited scenes with diverse objects and layouts, which in turn supervise the scene generator and form a self-distillation loop.

4.2 Scene Frame Generation

Existing approaches mainly adopt a modular and cascaded paradigm that address object composition, spatial placement and 3D generation in separate stages. However, this cascaded paradigm often propagates errors across different intermediate predictions. DistScene instead directly generates the environment and multiple independent objects together in a shared scene frame, preserving their component identities and spatial relationships without requiring external segmentation or reconstructed scene geometry at inference.

For representation, we adopt sparse voxels as the unified substrate for both objects and the environment. Given an input image I, we initialize $N + 1$ noise latents, corresponding to N objects and one environment, denoted as $\{z_i\}_{i=1}^{N+1}$ with z_{N+1} reserved for the environment. Crucially, the

environment latent acts as a bridge that anchors the positions of all objects and encodes their spatial interactions, thereby coupling the N object latents into a coherent scene. To jointly denoise these latents, we flatten each latent into a token sequence and concatenate the $N+1$ sequences along the token dimension into a single sequence $Z = [z_1; \ldots; z_{N+1}]$. Learnable type embeddings are added to distinguish object tokens and environment tokens, and the sparse DiT performs self-attention over Z, allowing all objects and the environment to exchange information during denoising. The generator $\mathcal{G}$ then jointly denoises Z conditioned on I following the same three-stage scheme as in Sec. 3. Formally,

$$Z = [z_1; \ldots; z_{N+1}], \quad \{x_i\}_{i=1}^{N+1} = \mathcal{G}(Z; I), \quad S = \mathcal{D}\big(\{x_i\}_{i=1}^{N+1}\big) = \{O_1, \ldots, O_N, E\}, \quad (4)$$

where $\{x_i\}$ are the generated latents, $\mathcal{D}$ is the sparse-voxel decoder, and S is the complete compositional scene.

4.3 Scene-Aware Object-Centric Refinement

After the scene is decomposed into N objects and an environment, each object occupies only a small fraction of the global sparse voxel grid. Consequently, the effective spatial resolution allocated to each object is far below the maximum that the representation can offer, which limits the fidelity of individual objects. To push the quality of every component to the extreme, we introduce a dedicated object refinement stage.

Given the decomposed scene $\{O_i\}_{i=1}^{N} \cup \{E\}$, we extract each object O_i, re-center and rescale it into a canonical space, and feed it into a sparse-voxel-based object refinement model $\mathcal{R}$. To make the refinement aware of the surrounding context, $\mathcal{R}$ is conditioned jointly on the input image I and the scene latents produced by $\mathcal{G}$. Specifically, we annotate the scene latents with a learnable embedding that marks the region corresponding to O_i, so that the refinement process knows which part of the scene information to interact with. The model then refines the object at a higher effective resolution, producing

$$\hat{O}_i = \mathcal{R}\big(O_i \mid I, \{x_j\}_{j=1}^{N+1}, e_i\big), \quad (5)$$

where $\{x_j\}$ are the scene latents and e_i is the embedding marking O_i's corresponding region. The refined objects are then placed back to their original positions, yielding the final scene

$$S' = \{\hat{O}_1, \ldots, \hat{O}_N, E\}. \quad (6)$$

Since the refinement is conditioned on the scene latents and the marker embedding, the refined object remains spatially and semantically consistent with the scene when placed back, while still benefiting from the higher effective resolution. This refinement stage operates entirely in the same sparse voxel domain and can be seamlessly integrated with the generator, allowing each object to be refined independently without affecting the global scene layout.

4.4 Training Objective and Network Architecture

We instantiate the scene generator $\mathcal{G}$ as a sparse DiT and fine-tune it with LoRA, since the underlying sparse-voxel generator already possesses strong image-to-3D priors and only needs to adapt to the compositional setting. Training follows the standard flow-matching objective used in sparse-voxel generation, applied separately to each object latent and to the environment latent:

$$\mathcal{L}_{\text{scene}} = \sum_{i=1}^{N+1} \mathbb{E}_{t, z_0^i, z_1^i} \left\| v_\theta(z_t^i, t) - (z_1^i - z_0^i) \right\|^2, \quad (7)$$

where $i = 1, \ldots, N$ indexes the N objects and $i = N+1$ indexes the environment, z_0^i and z_1^i are the noise and data latents, and v_θ is the predicted velocity. This loss jointly supervises the flow of every object and the scene, while LoRA adapters keep the pretrained backbone frozen, enabling efficient fine-tuning without sacrificing the strong generative prior.

For the object refinement model $\mathcal{R}$, we construct coarse-to-fine training pairs without any manual annotation. Given a high-quality object latent z_1, we downsample it, pass it through the sparse VAE encoder and decoder, and upsample it back to obtain a degraded coarse latent z_0. The refinement model is trained to recover z_1 from z_0 with the same flow-matching objective:

$$\mathcal{L}_{\text{refine}} = \mathbb{E}_{t, z_0, z_1} \left\| v_\phi(z_t, t \mid I, e) - (z_1 - z_0) \right\|^2, \quad (8)$$

Metric	Hunyuan3D 2.1	TRELLIS	EvoScene	Extend3D	TRELLIS.2	ours
CD-L1 ↓	0.1933	0.1397	0.1120	0.0832	0.0831	**0.0772**
F@0.05 ↑	0.411	0.556	0.543	0.680	0.701	**0.722**

Table 1: Geometry comparison on the outdoor UrbanScene3D benchmark. Lower CD-L1 and higher F-score indicate better reconstruction.

Methods	MIDI-test					Gen3DSR-test	
	CD_S ↓	FS_S ↑	CD_O ↓	FS_O ↑	IoU ↑	CD_S ↓	FS_S ↑
Gen3DSR	0.3795	26.45	0.2609	28.73	0.0712	0.6701	11.33
TRELLIS.2	0.1444	54.19	0.2475	31.07	0.1471	0.2099	44.44
MIDI	0.1550	53.46	0.2162	36.26	0.1639	0.2830	32.53
SAM3D	0.1486	59.79	0.1607	49.72	0.2102	0.2055	48.73
SceneGen	0.1502	51.40	0.1904	40.88	0.1217	0.2040	44.31
3D-Fixer	0.1295	65.08	0.1704	48.10	0.3527	0.1027	77.97
DistScene (ours)	**0.0877**	**71.59**	**0.1529**	**51.64**	**0.3688**	**0.0958**	**79.68**

Table 2: Geometry comparison on MIDI-test and Gen3DSR-test. S/O denote scene/object level; IoU measures bounding-box overlap. F-scores use a 0.1 threshold and are reported in percent.

where v_ϕ is the refinement velocity conditioned on the input image I and the marker embedding e that indicates the corresponding scene region. This allows the refinement module to learn detail recovery entirely from self-generated pairs, without requiring any external high-resolution data.

5 Experiments

5.1 Implementation Details

We use TRELLIS.2 as the pretrained object generator and adapt it with LoRA with a rank of 32. For Object-to-Scene Distillation, we construct $125k$ synthetic training scenes from $70k$ generated objects and $46k$ empty scenes, including $106k$ indoor scenes and $18k$ outdoor scenes. All synthetic training data are generated at a resolution of 512. Although the model is trained only at this resolution, we find that the adapted model supports inference at a resolution of 1024. Unless otherwise specified, all main experiments use 1024-resolution Scene-Frame Generation and Object-Centric Refinement. For evaluation, we follow 3D-Fixer to evaluate indoor scene generation on MIDI-test and Gen3DSR-test, reporting scene-level CD and F-score. On MIDI-test, we also report object-level CD and F-score, together with bounding-box IoU in the scene coordinate system. For outdoor scenes, we follow the settings of Extend3D and evaluate on UrbanScene3D, reporting CD-L1 and F-score.

5.2 Main Comparison

Outdoor scene generation. Table 1 compares outdoor scene geometry on UrbanScene3D. DistScene achieves the lowest CD-L1 (0.0772) and highest F-score (0.722) among the evaluated methods. Compared with Extend3D, CD-L1 decreases from 0.0832 to 0.0772, while F-score increases from 0.680 to 0.722. DistScene also improves both metrics over TRELLIS.2, showing the effectiveness of our object-scene distillation for outdoor scene generation.

Indoor scene generation. Table 2 reports results on MIDI-test and Gen3DSR-test. DistScene achieves the best scene-level CD and F-score on both benchmarks. On MIDI-test, it reduces scene CD from 0.1295 to 0.0877, a relative improvement of 32.2% over 3D-Fixer, and increases scene F-score from 65.08 to 71.59, an improvement of 6.51 percentage points. It also achieves the highest bounding-box IoU, as well as the best object-level CD and F-score among the compared methods. On Gen3DSR-test, DistScene obtains a scene CD of 0.0958 and an F-score of 79.68%, compared with 0.1027 and 77.97% for 3D-Fixer. These results demonstrate that DistScene improves both scene-level spatial alignment and object-level geometry on indoor scene generation.

Metric	TRELLIS.2	3D-Fixer	SAM3D	Extend3D	Ours
Image consistency	8.54	1.71	0.61	27.20	**61.95**
Geometry quality	9.15	3.29	0.61	19.15	**67.80**
Appearance quality	8.54	3.41	0.37	21.22	**66.46**
Average	8.74	2.80	0.53	22.52	**65.41**

Table 3: User study preference rates (%). Higher values indicate stronger participant preference.



Figure 4: Qualitative comparison with 3D-Fixer, MIDI, and SAM3D on Scannet.

Qualitative comparison. Figure 4 presents scene generation results using test images from Gen3DSR and real images from ScanNet. Compared with the input images, 3D-Fixer, MIDI, and SAM3D can recover the main objects, but often produce incomplete geometry or misaligned spatial relationships between objects. In contrast, DistScene jointly recovers the environment and multiple independent objects, while better preserving the relative layout and support relationships in the input image. These observations are consistent with the quantitative results in Table 2.

Figure 5 further presents examples using text-to-image-generated inputs. Despite the greater diversity in viewpoints, scene types, and object compositions, DistScene consistently generates environments and objects corresponding to the input content while preserving reasonable scene structures, object layouts, and component relationships.

User study. To evaluate perceptual quality, we conduct a user study with 43 participants on 20 randomly selected text-to-image-generated inputs. For each input, participants view anonymized outputs from five methods in randomized order and select the best result for image consistency, geometry quality, and appearance quality. We compute each method's winning probability as its fraction of selections and report the average across the three criteria. As shown in Table 3, DistScene achieves the highest preference rate on all three criteria and the highest average winning probability.

A qualitative resolution comparison is provided in the supplementary material (Figure 8).

5.3 Ablation Studies

We ablate the three key components of DistScene: environment generation in the shared scene frame, the distilled scene training data, and scene-conditioned object refinement. We further analyze the effect of generation resolution and report the runtime and memory cost of each configuration. In Table 4, SF denotes Scene-Frame Generation without local refinement, Env denotes the explicit environment component, and Ctx denotes the scene-frame context used during refinement. Unless otherwise specified, all variants operate at a resolution of 1024.

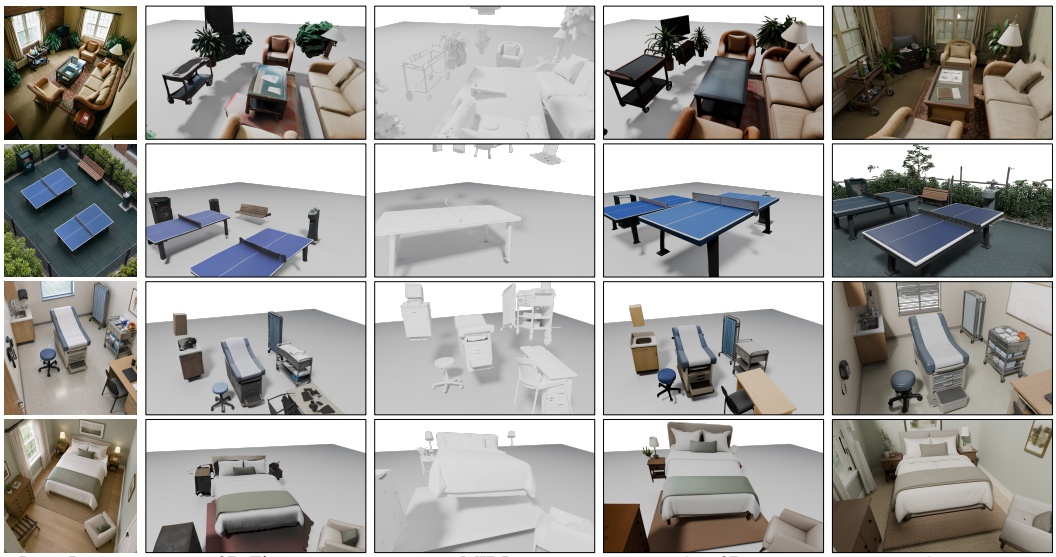

Figure 5: Qualitative comparison on text-to-image-generated inputs beyond the benchmarks.

Method	MIDI-test					Gen3DSR-test	
	$CD_S \downarrow$	$FS_S \uparrow$	$CD_O \downarrow$	$FS_O \uparrow$	IoU $\uparrow$	$CD_S \downarrow$	$FS_S \uparrow$
TRELLIS.2 (no mask)	0.1610	53.40	0.2222	34.46	0.1798	0.2177	41.45
TRELLIS.2 (mask)	0.1500	55.57	0.2280	34.12	0.1900	0.2099	44.44
SF w/o Env	0.0922	67.94	0.2444	29.98	0.1798	0.1367	62.04
SF w/o Env (MIDI)	0.0981	65.74	0.2347	33.03	0.2084	0.1936	46.68
SF w. Env	0.0928	68.86	0.2037	40.98	0.2544	0.1096	73.21
SF + Refine w/o Ctx	0.1055	66.54	0.1913	41.97	0.2964	0.1169	69.82
SF + Refine (full)	**0.0877**	**71.59**	**0.1529**	**51.64**	**0.3688**	**0.0958**	**79.68**

Table 4: Ablation study on MIDI-test and Gen3DSR-test. SF denotes Scene-Frame Generation, Env denotes environment generation, Refine denotes Object-Centric Refinement, and Ctx denotes scene-frame context.

Effect of environment generation. We first compare SF w/o Env and SF w. Env to evaluate the effect of jointly generating the environment with the objects. Adding the environment substantially improves object-level CD and F-score on MIDI-test, increasing the object F-score from 29.98 to 40.98, while improving bounding-box IoU from 0.1798 to 0.2544. It also reduces scene CD from 0.1367 to 0.1096 and raises the F-score from 62.04 to 73.21 on Gen3DSR-test. The MIDI scene CD remains nearly unchanged, while its scene F-score improves from 67.94 to 68.86. These results indicate that explicitly generating the environment provides useful scene-level context for organizing object geometry and spatial relationships. The qualitative results in Figure 6 show the same effect: without the environment component, the chair geometry in the first example is degraded, and the table lamp in the second indoor scene is missing.

Effect of Object-to-Scene Distillation. To isolate the effect of our distilled training data, we train the same object-only architecture on either our data or the MIDI training set, corresponding to SF w/o Env and SF w/o Env (MIDI), respectively. Both MIDI-test and Gen3DSR-test are out of domain for our synthetic training data, whereas MIDI-test is in domain for the MIDI training set. Despite this disadvantage, our data improves the scene-level metrics on MIDI-test and provides a much larger gain on Gen3DSR-test, reducing scene CD from 0.1936 to 0.1367 and increasing F-score from 46.68 to 62.04. The MIDI-trained variant retains an expected advantage on the object-level metrics and bounding-box IoU of MIDI-test. Compared with the pretrained TRELLIS.2 baseline supplied with object masks, SF w/o Env also achieves substantially stronger scene-level performance on both benchmarks. These comparisons show that our distilled data transfers the object generator toward scene-level generation and generalizes beyond its synthetic training distribution.

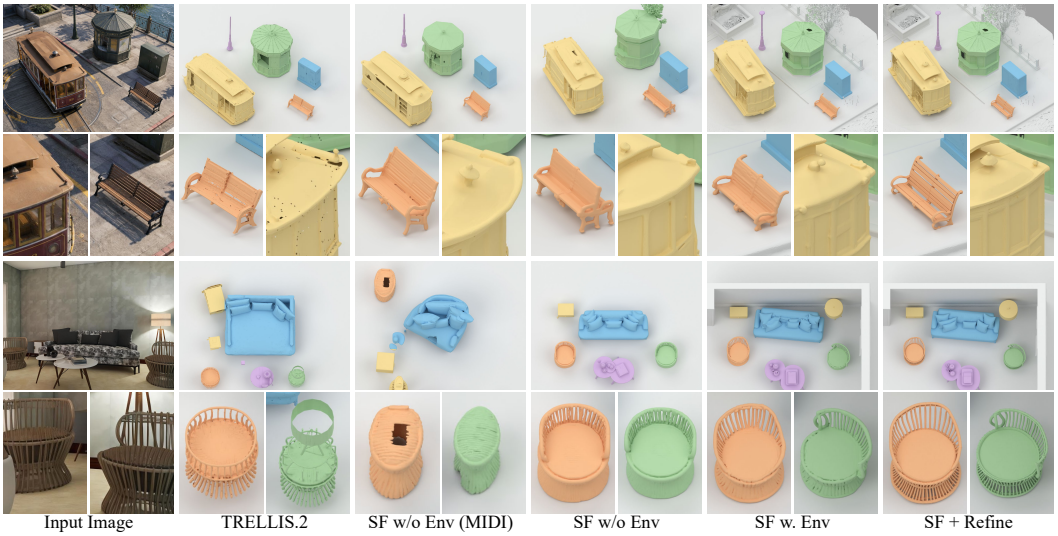

Figure 6: Qualitative ablation study of training supervision, environment generation, and refinement.

Method	MIDI-test		Gen3DSR-test		Time ↓		Memory (GB) ↓
	CD_S ↓	FS_S ↑	CD_S ↓	FS_S ↑	Scene (s)	Refine (s/obj)	
SF-512	0.1318	61.77	0.1077	71.96	**5.7**	–	**17.9**
SF-1024	0.0928	68.86	0.1096	73.21	36.4	–	21.4
SF-512 + Ref-512	0.0934	68.35	0.0974	78.35	**5.7**	**3.3**	18.0
SF-1024 + Ref-1024 (full)	**0.0877**	**71.59**	**0.0958**	**79.68**	36.4	38.5	21.8

Table 5: Ablation of resolution for refinement and efficiency.

Effect of Object-Centric Refinement. Comparing SF w. Env with the full model shows that Object-Centric Refinement consistently improves both scene- and object-level quality. On MIDI-test, it reduces object CD from 0.2037 to 0.1529, increases object F-score from 40.98 to 51.64, and improves bounding-box IoU from 0.2544 to 0.3688. On Gen3DSR-test, scene CD decreases from 0.1096 to 0.0958 and F-score increases from 73.21 to 79.68. We further remove the scene-frame latent context during refinement. This variant performs markedly worse than the full model across all metrics, including a drop in object F-score from 51.64 to 41.97 on MIDI-test. Without scene context, the refinement model has difficulty associating the target component with its corresponding object in the input image, demonstrating that scene-conditioned refinement is essential for recovering the correct local geometry.

Resolution and efficiency. Table 5 reports the quality, runtime, and peak memory of different scene-frame and refinement resolutions. Increasing the resolution consistently improves reconstruction quality at additional runtime and modest memory cost. The 512-resolution variant with refinement offers a favorable efficiency–quality trade-off, requiring 5.7 seconds for scene generation and 3.3 seconds per refined object with 18.0 GB peak memory, while the full 1024-resolution model achieves the best accuracy with 21.8 GB peak memory.

6 Conclusion

We introduced DistScene, which treats the environment as an explicit component of compositional 3D scene generation. This shared scene-frame formulation enables the environment and objects to be generated together, providing a common geometric context for their spatial arrangement. Object-Centric Refinement then restores fine object geometry in local coordinate frames, and Object-to-Scene Distillation supplies the aligned training supervision required by the framework. Extensive experiments on indoor and outdoor benchmarks show that DistScene produces more coherent layouts and more accurate scene geometry than existing methods.

AI USE STATEMENT

As part of the proposed data-generation pipeline, we use pretrained 3D generative models to produce object and environment assets and to construct synthetic scenes for training supervision. This use is part of the method described in the paper rather than an automated substitute for the authors' scientific decisions.

We also used OpenAI Codex solely for language-related assistance, including translating author-written Chinese text into English, correcting grammar and wording, improving readability, organizing the presentation of the paper, and formatting LaTeX content. The generative AI tool was not used to formulate the scientific contributions, design the model or training procedure, design or run experiments, generate or alter experimental results, interpret the experimental findings, implement the core model or training pipeline, create or modify scientific figures, or produce mathematical claims or proofs.

All AI-assisted text and LaTeX edits were reviewed and revised by the authors. The authors independently checked the technical descriptions, citations, methodological claims, experimental settings, and reported results against the underlying sources and records. The authors take full responsibility for the final manuscript, including all text, claims, experiments, code, data, and artifacts.

ETHICS STATEMENT

(This section is **recommended** and does not count toward the page limit.)

If authors feel that their paper submission raises questions regarding the Code of Ethics, they are encouraged to include a paragraph of Ethics Statement (at the end of the main text before references) to address potential concerns where appropriate. Topics include, but are not limited to, studies that involve human subjects, practices to data set releases, potentially harmful insights, methodologies and applications, potential conflicts of interest and sponsorship, discrimination/bias/fairness concerns, privacy and security issues, legal compliance, and research integrity issues (e.g., IRB, documentation, research ethics). This statement should not be more than 1 page.

REPRODUCIBILITY STATEMENT

(This section is **recommended** and does not count toward the page limit.)

It is important that the work published in ICLR is reproducible. Authors are strongly encouraged to include a paragraph-long Reproducibility Statement at the end of the main text (before references) to discuss the efforts that have been made to ensure reproducibility. This paragraph should not itself describe details needed for reproducing the results, but rather reference the parts of the main paper, appendix, and supplemental materials that will help with reproducibility. For example, for novel models or algorithms, a link to an anonymous downloadable source code can be submitted as supplementary materials; for theoretical results, clear explanations of any assumptions and a complete proof of the claims can be included in the appendix; for any datasets used in the experiments, a complete description of the data processing steps can be provided in the supplementary materials. Each of the above are examples of things that can be referenced in the reproducibility statement.

AUTHOR CONTRIBUTIONS

If you'd like to, you may include a section for author contributions as is done in many journals. This is optional and at the discretion of the authors.

ACKNOWLEDGMENTS

Use unnumbered third level headings for the acknowledgments. All acknowledgments, including those to funding agencies, go at the end of the paper.

REFERENCES

Andreea Ardelean, Mert Özer, and Bernhard Egger. Gen3dsr: Generalizable 3d scene reconstruction via divide and conquer from a single view. In *2025 International Conference on 3D Vision (3DV)*,

pp. 616–626. IEEE, 2025.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. In *International conference on learning representations*, volume 2026, pp. 138846–138923, 2026.

Minghao Chen, Roman Shapovalov, Iro Laina, Tom Monnier, Jianyuan Wang, David Novotny, and Andrea Vedaldi. Partgen: Part-level 3d generation and reconstruction with multi-view diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 5881–5892, 2025.

Xingyu Chen, Fu-Jen Chu, Pierre Gleize, Kevin J Liang, Alexander Sax, Hao Tang, Weiyao Wang, Michelle Guo, Thibaut Hardin, Xiang Li, et al. Sam 3d: 3dfy anything in images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7220–7232, 2026a.

Xingyu Chen, Fu-Jen Chu, Pierre Gleize, Kevin J Liang, Alexander Sax, Hao Tang, Weiyao Wang, Michelle Guo, Thibaut Hardin, Xiang Li, et al. Sam 3d: 3dfy anything in images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7220–7232, 2026b.

Ken Deng, Yuan-Chen Guo, Jingxiang Sun, Zi-Xin Zou, Yangguang Li, Xin Cai, Yan-Pei Cao, Yebin Liu, and Ding Liang. Detailgen3d: Generative 3d geometry enhancement via data-dependent flow. In *2026 International Conference on 3D Vision (3DV)*, pp. 1–31. IEEE, 2026a.

Ken Deng, Yunhan Yang, Jingxiang Sun, Xihui Liu, Yebin Liu, Ding Liang, and Yan-Pei Cao. Geosam2: Unleashing the power of sam2 for 3d part segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6367–6376, 2026b.

Daniel DeTone, Tianwei Shen, Fan Zhang, Lingni Ma, Julian Straub, Richard Newcombe, and Jakob Engel. Boxer: Robust lifting of open-world 2d bounding boxes to 3d, 2026. URL `https://arxiv.org/abs/2604.05212`.

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

Xianglong He, Zi-Xin Zou, Chia-Hao Chen, Yuan-Chen Guo, Ding Liang, Chun Yuan, Wanli Ouyang, Yan-Pei Cao, and Yangguang Li. Sparseflex: High-resolution and arbitrary-topology 3d shape modeling. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14822–14833. IEEE, 2025.

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23646–23657. IEEE, 2025.

Team Hunyuan3D, Shuhui Yang, Mingxin Yang, Yifei Feng, Xin Huang, Sheng Zhang, Zebin He, Di Luo, Haolin Liu, Yunfei Zhao, Qingxiang Lin, Zeqiang Lai, Xianghui Yang, Huiwen Shi, Zibo Zhao, Bowen Zhang, Hongyu Yan, Lifu Wang, Sicong Liu, Jihong Zhang, Meng Chen, Liang Dong, Yiwen Jia, Yulin Cai, Jiaao Yu, Yixuan Tang, Dongyuan Guo, Junlin Yu, Hao Zhang, Zheng Ye, Peng He, Runzhou Wu, Shida Wei, Chao Zhang, Yonghao Tan, Yifu Sun, Lin Niu, Shirui Huang, Bojian Zheng, Shu Liu, Shilin Chen, Xiang Yuan, Xiaofeng Yang, Kai Liu, Jianchen Zhu, Peng Chen, Tian Liu, Di Wang, Yuhong Liu, Linus, Jie Jiang, Jingwei Huang, and Chunchao Guo. Hunyuan3d 2.1: From images to high-fidelity 3d assets with production-ready pbr material, 2025. URL `https://arxiv.org/abs/2506.15442`.

Zeqiang Lai, Yunfei Zhao, Haolin Liu, Zibo Zhao, Qingxiang Lin, Huiwen Shi, Xianghui Yang, Mingxin Yang, Shuhui Yang, Yifei Feng, et al. Hunyuan3d 2.5: Towards high-fidelity 3d assets generation with ultimate details. *arXiv preprint arXiv:2506.16504*, 2025.

Weiyu Li, Jiarui Liu, Hongyu Yan, Rui Chen, Yixun Liang, Xuelin Chen, Ping Tan, and Xiaoxiao Long. Craftsman3d: High-fidelity mesh generation with 3d native generation and interactive geometry refiner. *arXiv preprint arXiv:2405.14979*, 2024.

Yangguang Li, Zi-Xin Zou, Zexiang Liu, Dehu Wang, Yuan Liang, Zhipeng Yu, Xingchao Liu, Yuan-Chen Guo, Ding Liang, Wanli Ouyang, et al. Triposg: High-fidelity 3d shape synthesis using large-scale rectified flow models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Zhihao Li, Yufei Wang, Heliang Zheng, Yihao Luo, and Bihan Wen. Sparc3d: Sparse representation and construction for high-resolution 3d shapes modeling. *Advances in Neural Information Processing Systems*, 38:118582–118600, 2026.

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

Yuchen Lin, Chenguo Lin, Panwang Pan, Honglei Yan, Yiqiang Feng, Yadong Mu, and Katerina Fragkiadaki. Partcrafter: Structured 3d mesh generation via compositional latent diffusion transformers. *arXiv preprint arXiv:2506.05573*, 2025b.

Yuchen Lin, Chenguo Lin, Panwang Pan, Honglei Yan, Feng Yiqiang, Yadong Mu, and Katerina Fragkiadaki. Partcrafter: Structured 3d mesh generation via compositional latent diffusion transformers. *Advances in Neural Information Processing Systems*, 38:35387–35415, 2026.

Haolin Liu, Yujian Zheng, Guanying Chen, Shuguang Cui, and Xiaoguang Han. Towards high-fidelity single-view holistic reconstruction of indoor scenes. In *European Conference on Computer Vision*, pp. 429–446. Springer, 2022.

Tianyu Liu, Weitao Xiong, Kunming Luo, Manyuan Zhang, Peng Li, Yuan Liu, and Ping Tan. Autoweather4d: Autonomous driving video weather conversion via g-buffer dual-pass editing. *arXiv preprint arXiv:2603.26546*, 2026.

Yifan Lu, Xuanchi Ren, Jiawei Yang, Tianchang Shen, Zhangjie Wu, Jun Gao, Yue Wang, Siheng Chen, Mike Chen, Sanja Fidler, et al. Infinicube: Unbounded and controllable dynamic 3d driving scene generation with world-guided video models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 27272–27283, 2025.

Kunming Luo, Hongyu Yan, Yuan Liu, Zihao Zhang, Manyuan Zhang, Wenping Wang, and Ping Tan. Ctr3d: Cross-view token reduction for dense multi-view generation. In *2026 International Conference on 3D Vision (3DV)*, pp. 956–966, 2026. doi: 10.1109/3DV69130.2026.00096.

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.

Pablo Martinez-Gonzalez, Sergiu Oprea, Alberto Garcia-Garcia, Alvaro Jover-Alvarez, Sergio Orts-Escolano, and Jose Garcia-Rodriguez. Unrealrox: an extremely photorealistic virtual reality environment for robotics simulations and synthetic data generation. *Virtual Reality*, 24(2):271–288, 2020.

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

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 *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 52–61. IEEE, 2020.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. In *International Conference on Learning Representations*, volume 2025, pp. 28085–28128, 2025.

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

Yukai Shi, Weiyu Li, Zihao Wang, Hongyang Li, Xingyu Chen, Ping Tan, and Lei Zhang. 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*, pp. 27146–27156, 2026.

Xiang Tang, Ruotong Li, and Xiaopeng Fan. Zeroscene: A zero-shot framework for 3d scene generation from a single image and controllable texture editing. *arXiv preprint arXiv:2509.23607*, 2025.

Massimiliano Viola, Kevin Qu, Nando Metzger, Bingxin Ke, Alexander Becker, Konrad Schindler, and Anton Obukhov. Marigold-dc: Zero-shot monocular depth completion with guided diffusion. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5359–5370. IEEE, 2025.

Hanqing Wang, Jiahe Chen, Wensi Huang, Qingwei Ben, Tai Wang, Boyu Mi, Tao Huang, Siheng Zhao, Yilun Chen, Sizhe Yang, et al. Grutopia: Dream general robots in a city at scale. *arXiv preprint arXiv:2407.10943*, 2024a.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5294–5306. IEEE, 2025a.

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. Vggt-$Omega$. *arXiv preprint arXiv:2605.15195*, 2026a.

Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. Moge: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5261–5271. IEEE, 2025b.

Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, and Jiaolong Yang. Moge-2: Accurate monocular geometry with metric scale and sharp details. *Advances in Neural Information Processing Systems*, 38:35928–35959, 2026b.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20697–20709. IEEE, 2024b.

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-equivariant visual geometry learning, 2026c. URL https://arxiv.org/abs/2507.13347.

Dong Wu, Zike Yan, and Hongbin Zha. Panorecon: Real-time panoptic 3d reconstruction from monocular video. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21507–21518. IEEE, 2024a.

Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Yikang Yang, Jiachen Qian, Siyu Zhu, Xun Cao, Philip Torr, Yao Yao, et al. Direct3d-s2: Gigascale 3d generation made easy with spatial sparse attention. *Advances in Neural Information Processing Systems*, 38:170778–170804, 2026.

Wayne Wu, Honglin He, Yiran Wang, Chenda Duan, Jack He, Zhizheng Liu, Quanyi Li, and Bolei Zhou. Metaurban: A simulation platform for embodied ai in urban spaces. *arXiv e-prints*, pp. arXiv–2407, 2024b.

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

Jianfeng Xiang, Xiaoxue Chen, Sicheng Xu, Ruicheng Wang, Zelong Lv, Yu Deng, Hongyuan Zhu, Yue Dong, Hao Zhao, Nicholas Jing Yuan, and Jiaolong Yang. Native and compact structured latents for 3d generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14419–14429, June 2026.

Gangwei Xu, Haotong Lin, Hongcheng Luo, Xianqi Wang, Jingfeng Yao, Lianghui Zhu, Yuechuan Pu, Cheng Chi_, Haiyang Sun, Bing Wang, et al. Pixel-perfect depth with semantics-prompted diffusion transformers. *Advances in Neural Information Processing Systems*, 38:174731–174755, 2026.

Hongyu Yan, Kunming Luo, Weiyu Li, Kaiyi Zhang, Yixun Liang, Jingwei Huang, Chunchao Guo, and Ping Tan. Posemaster: A unified 3d native framework for stylized pose generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 34292–34302, 2026.

Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10371–10381. IEEE, 2024a.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *Advances in neural information processing systems*, 37:21875–21911, 2024b.

Kaixin Yao, Longwen Zhang, Xinhao Yan, Yan Zeng, Qixuan Zhang, Lan Xu, Wei Yang, Jiayuan Gu, and Jingyi Yu. Cast: Component-aligned 3d scene reconstruction from an rgb image. *ACM Transactions on Graphics (TOG)*, 44(4):1–19, 2025.

Chongjie Ye, Yushuang Wu, Ziteng Lu, Jiahao Chang, Xiaoyang Guo, Jiaqing Zhou, Hao Zhao, and Xiaoguang Han. Hi3dgen: High-fidelity 3d geometry generation from images via normal bridging. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 01–12. IEEE, 2025.

Ze-Xin Yin, Liu Liu, Xinjie Wang, Wei Sui, Zhizhong Su, Jian Yang, and Jin Xie. 3d-fixer: Coarse-to-fine in-place completion for 3d scenes from a single image. *arXiv preprint arXiv:2604.04406*, 2026.

Chuhan Zhang, Guillaume Le Moing, Skanda Koppula, Ignacio Rocco, Liliane Momeni, Junyu Xie, Shuyang Sun, Rahul Sukthankar, Joëlle K. Barral, Raia Hadsell, Zoubin Ghahramani, Andrew Zisserman, Junlin Zhang, and Mehdi S. M. Sajjadi. Efficiently reconstructing dynamic scenes one d4rt at a time. In *CVPR*, 2026.

Longwen Zhang, Ziyu Wang, Qixuan Zhang, Qiwei Qiu, Anqi Pang, Haoran Jiang, Wei Yang, Lan Xu, and Jingyi Yu. Clay: A controllable large-scale generative model for creating high-quality 3d assets. *ACM Transactions On Graphics (TOG)*, 43(4):1–20, 2024.

Junsheng Zhou, Yu-Shen Liu, and Zhizhong Han. Zero-shot scene reconstruction from single images with deep prior assembly. *Advances in Neural Information Processing Systems*, 37:39104–39127, 2024.

A SUPPLEMENTARY MATERIAL

In this supplement, we first provide the network architecture detail in Sec. A.1. We also provide further data curation details in Sec. A.2 and further comparison in Sec. A.4. Finally, we provide discussion on limitation and more qualitative results in Sec. A.5. We encourage the readers to view our accompanying videos in the supplementary materials.

A.1 NETWORK ARCHITECTURE

Figure 7 illustrates the three geometry-generation modules of DistScene: joint sparse-structure generation, scene-frame geometry generation, and scene-conditioned object refinement. Colors indicate corresponding scene components across modules. The snowflake and flame symbols denote frozen pretrained transformer weights and trainable LoRA adapters, respectively.

Joint sparse-structure generation. We allocate a latent representation to each scene component, including the environment and the individual objects. Starting from noise, the flow transformer jointly generates the component structure latents, using positional encodings (PE) to distinguish their corresponding slots. Decoding these latents produces separate sparse voxel supports in the shared scene frame. Each support specifies the spatial extent and coarse structure of its component while retaining its identity.

Scene-frame geometry generation. The generated voxel supports provide the coordinates for the subsequent geometry-generation stage. Conditioned on the input image, a sparse flow transformer jointly generates geometry features on the active voxels of all components. These features form the structured geometry latents, denoted as Shape SLat in Figure 7. Since the voxel coordinates are defined in a shared scene frame, the decoded environment and object meshes can be assembled directly while remaining separate components.

Scene-conditioned object refinement. For each target object, we normalize its scene-frame mesh through translation and uniform scaling and voxelize the normalized mesh to construct a local sparse support. This transformation preserves the object's orientation and increases the effective spatial resolution available for its geometry. The refinement module generates a new local geometry latent on this support, conditioned on the input image and the fixed scene-frame geometry latents of the environment and all objects. Component correspondence is retained through the positional encodings, allowing the local target to be associated with its scene-frame component. During refinement, only the target object's local latent is generated; the scene-frame latents remain unchanged.

The refined latent is decoded into an object mesh and placed back into the scene using the recorded scene-space placement and scale. Repeating this procedure for each object yields a refined compositional scene while retaining the generated environment. Appearance generation is conditioned on the corresponding geometry and is omitted from Figure 7 for clarity.

A.2 TRAINING DATA CURATION

Asset generation and scene composition. We sample text prompts for object assets and empty environments, generate their conditioning images, and use the pretrained object generator to obtain object-centric latents and textured meshes. For each environment, we sample a small set of compatible objects, choose their scales from the asset and environment size ranges, and place them in the environment coordinate system using uniform scale and translation. Indoor and outdoor scenes additionally filter candidates by compatible placement surfaces and scene-size ratios. These constraints prevent degenerate compositions in which the environment is disproportionately large or small relative to the selected objects. The component identity and the resulting transform are retained for every placed object.

The composition is accepted only after geometry-based physical plausibility checks. We first discard environments with unusable floor structure, such as large floor holes or unsupported multi-story geometry. We then estimate a walkable floor mask and its distance to walls or other environmental obstacles. Candidate object centers are sampled from this free-floor region, subject to footprint-coverage and wall-clearance requirements. Each object is then dropped onto the estimated support

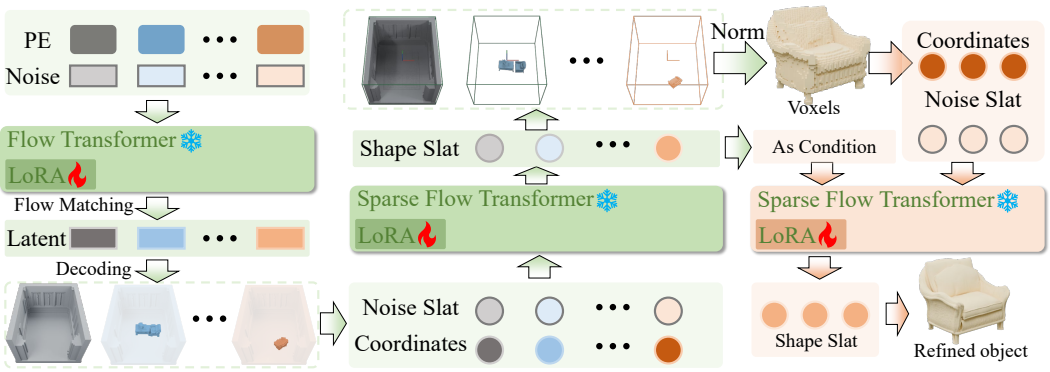

Figure 7: Detailed network architecture of DistScene. Left: a flow transformer jointly generates sparse-structure latents for the environment and individual objects, which are decoded into scene-frame voxel supports. Middle: a sparse flow transformer generates component geometry latents on these supports. Right: the target object is normalized to construct a local voxel support, on which a refinement transformer generates local geometry conditioned on the fixed scene-frame latents. Colors indicate component correspondence; snowflakes and flames denote frozen pretrained transformer weights and trainable LoRA adapters. Image conditioning and appearance generation are omitted for clarity.

Algorithm 1 Data Construction and Training

Require: Pretrained object generator $\mathcal{G}$
Ensure: Scene generator $\mathcal{F}_{\text{scene}}$ and refinement model $\mathcal{F}_{\text{ref}}$
1: Generate environment and object assets with $\mathcal{G}$
2: Compose valid scenes and render conditioning images
3: Encode scene-frame and object-centric latent targets
4: Initialize $\mathcal{F}_{\text{scene}}$ and $\mathcal{F}_{\text{ref}}$ from $\mathcal{G}$
5: Train $\mathcal{F}_{\text{scene}}$ on the scene-frame targets
6: Train $\mathcal{F}_{\text{ref}}$ on the object-centric targets

surface; poses that float above the floor or penetrate it beyond a small tolerance are discarded. Object–object intersections are filtered with a fast footprint test followed by an axis-aligned bounding-box test. To detect intersections with the environment, we sample non-floor environment surfaces and query the nearest 3D and height-band horizontal distances from the object surface. A placement is rejected when collision happens and only compositions with valid placements are retained.

A.3 Training and Inference Details

We use TRELLIS.2 as the pretrained object generator and adapt the trainable generation modules with LoRA of rank 32, while keeping the pretrained backbones frozen. We optimize the LoRA parameters with AdamW using a learning rate of 1×10^{-4} and a global batch size of 64. Each adapted module is trained for 20,000 iterations on the 512-resolution training targets. At test time, the same modules are run at 1024 resolution for the main experiments.

Due to the GPU memory limitation. We cap training scenes at 30 components, including the environment, and randomly pad each component sequence by appending a variable number of empty object latents. These empty slots are represented by empty supports and are included during training, allowing the model to be queried with a fixed component count larger than the number of visible objects. At inference, we use 20 component slots by default; slots that decode to empty supports are discarded from the final decomposable scene.

Algorithms 1 and 2 summarize the training and inference procedures of DistScene. We first construct component-aligned scene supervision from generated assets and use it to train Scene-Frame Generation and Object-Centric Refinement. At inference, the environment and objects are generated jointly and each object is subsequently refined in its local space.

Algorithm 2 Inference

Require: Input image I, $\mathcal{F}_{\text{scene}}$, and $\mathcal{F}_{\text{ref}}$
Ensure: Decomposable 3D scene
1: Jointly generate the environment and objects in the scene frame
2: Decode their scene-frame geometry
3: **for** each generated object **do**
4: Transform the object to its local space
5: Refine it using the generated scene as context
6: Decode its geometry and appearance
7: Transform the refined object back to the scene frame
8: **end for**
9: **return** Environment and refined object components

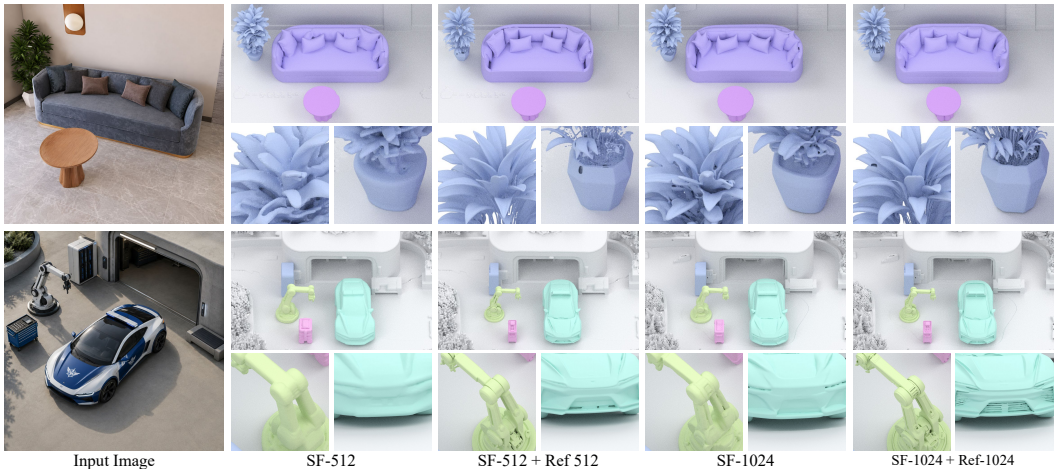

Figure 8: Qualitative comparison of generation resolution and local refinement. From left to right: input image, SF-512, SF-512 + Ref-512, SF-1024, and SF-1024 + Ref-1024. Each configuration is shown with a scene view and corresponding geometry crops. Matched views highlight differences in local detail while allowing comparison of the overall scene arrangement.

A.4 Resolution Comparison

Figure 8 compares four configurations: SF-512, SF-512 + Ref-512, SF-1024, and SF-1024 + Ref-1024, where SF and Ref denote scene-frame generation and object refinement, respectively. Matched scene views and detail crops allow comparison of both the overall arrangement and local geometry.

The examples show that similar overall arrangements can coexist with different levels of geometric detail. In the indoor example, the plant-leaf crops reveal differences in thin structures; in the outdoor example, the highlighted objects and vehicle surfaces expose differences in local shape detail. Refinement at resolution 1024 produces more clearly resolved structures in these examples. The quantitative quality–cost comparison is reported in Table 5.

A.5 Additional Results and Limitations

A.6 Failure Cases and Limitations

Figure 9 illustrates representative failure cases on indoor and outdoor inputs. For indoor scenes, our scene construction pipeline uses an image of an empty environment to generate the environment component before composing object assets. In the current data-generation setup, obtaining reliable fully enclosed empty rooms from pretrained 3D object generator is difficult; consequently, the indoor environment assets used for training generally omit the ceiling. This prior is reflected in some reconstructions, which may produce an open-top room even when the input image depicts a closed interior.

For outdoor scenes, the large spatial extent of the environment distributes a limited number of scene-frame voxels over a wide region. The resulting representation can therefore lose fine geometric detail and may contain holes in particularly large or complex environments. The outdoor training set is also currently smaller than the indoor set because of computationally resources. Expanding the outdoor corpus and improving large-scale scene representations are important directions for future work.

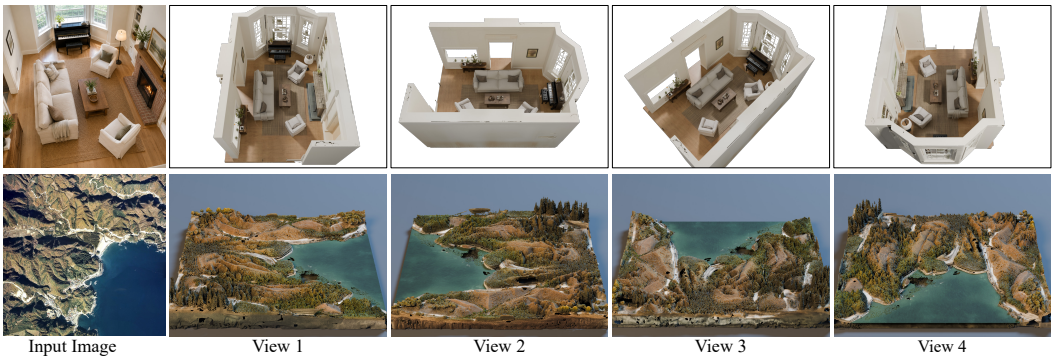

Figure 9: Representative failure cases. Indoor reconstructions may omit ceilings because the current environment construction relies on empty-scene images that rarely provide fully enclosed rooms. Outdoor reconstructions can exhibit holes or missing fine geometry when the scene spans a large spatial extent, reflecting the limited scene-frame resolution and the smaller outdoor training corpus.

A.7 Additional Qualitative Results

Figures 10 and 11 provide additional qualitative comparisons on indoor and outdoor scenes with diverse object configurations. DistScene consistently recovers a globally coherent layout that agrees with the input image, while generating both the surrounding environment and individual objects with detailed local geometry.

Extend3D optimizes a reconstructed scene using consistency between the input image and its estimated depth. This objective can preserve the visible appearance of the input, but provides limited constraints for heavily occluded regions, often resulting in incomplete geometry. Moreover, Extend3D does not explicitly represent the scene as independently addressable components, and therefore does not directly support object-level editing. TRELLIS.2 is an object-level 3D generator. When given a segmented multi-object image, it can synthesize several objects and recover recognizable local appearance. However, all objects are generated as a single fused object-centric representation rather than as independently indexed scene components. Without an explicit scene frame and component-wise spatial representation, the relative positions and scales of the generated objects are often inaccurate, while small objects receive limited spatial support and lose fine geometric details. 3D-Fixer and SAM3D can recover several individual objects, but their outputs may contain incomplete geometry or missing environmental context, leading to less complete object–environment relationships. In contrast, DistScene jointly generates the environment and multiple independent objects in a shared scene frame, which provides both coherent scene layout and directly editable component-level geometry.

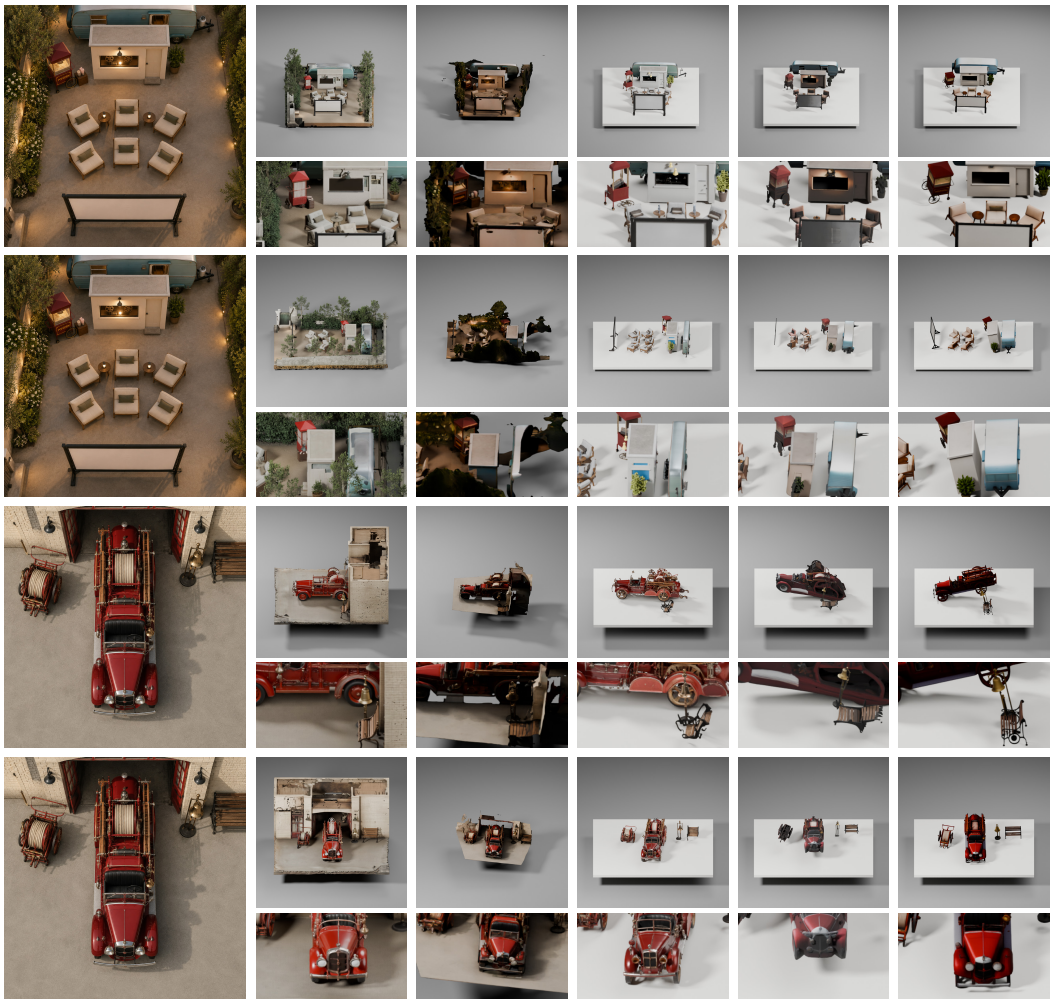

Figure 10: Additional qualitative comparisons between DistScene and Extend3D, TRELLIS.2, 3D-Fixer, and SAM3D. Each result is accompanied by a zoomed-in view for examining local geometric details. DistScene better preserves the scene layout and environmental context while producing more complete and detailed object geometry.



Figure 11: Additional qualitative comparisons between DistScene and Extend3D, TRELLIS.2, 3D-Fixer, and SAM3D. Zoomed-in views are provided for each result to highlight local geometric details. DistScene produces more coherent scene layouts, more complete environmental context, and finer object geometry.