# WorldAct: Activating Monolithic 3D Worlds into Interactive-Ready Object-Centric Scenes

**Jichen Hu**[1]*, **Jiawei Guo**[1]*, **Jiazhong Cen**[1]*, **Chen Yang**[2], **Sikuang Li**[1], **Wei Shen**[1]†
[1]MoE Key Lab of Artificial Intelligence, AI Institute,
School of Computer Science, Shanghai Jiao Tong University [2]Huawei Inc.

## Abstract

Recent 3D world modeling systems based on generative scene synthesis, such as Marble, can create coherent and explorable 3D environments, yet their outputs are typically static monolithic assets with limited editability and physical interaction. This restricts their use in immersive content creation and embodied simulation, where generated worlds must be actively modified and manipulated. To tackle this challenge, we present **WorldAct**, a framework that converts static generated 3D worlds into editable and interaction-ready scenes. WorldAct uses a multimodal agent to guide scene decomposition, identify actionable objects, reconstruct geometrically aligned object-level meshes for interaction, and restore the residual background via 3D inpainting. The resulting scenes support object-level editing, collision-aware manipulation, and embodied task execution while preserving global scene coherence. Experiments show that WorldAct enables richer interaction scenarios than the original generated scenes, suggesting a practical path toward editable and interactive 3D world models.

## 1 Introduction

Recent advancements in generative modeling have enabled the creation of immersive 3D worlds [78, 18, 49, 10, 14, 11, 51, 62] from simple text or image prompts. These models synthesize large-scale, spatially coherent environments, serving as a foundational tool for virtual simulation and digital content creation.

Despite these advances, editability and interactivity remain critical limitations. Existing 3D generative world models typically produce static, monolithic 3D representations, where objects are fused into a single structure and cannot be individually selected, moved, or replaced. This limits their use in creative workflows such as game design and interior decoration, where fine-grained scene editing is essential. It also restricts embodied AI simulation, as agents cannot manipulate specific entities in an unstructured scene. Without explicit semantic and physical object decoupling, generated worlds remain inert, serving only as visually plausible environments.

To address the lack of interactivity in existing 3D generative world models, we present **WorldAct**, a framework that converts monolithic 3D Gaussian Splatting (3DGS) [20] scenes into editable and physically interactive worlds. Given a generated 3DGS scene, WorldAct first uses a vision-language agent to find objects that can be manipulated and select useful viewpoints for scene analysis. The selected views are then segmented in 2D, projected back to 3D, and combined to separate individual objects from the original scene. After removing these objects, WorldAct fills the missing background regions and rebuilds high-quality object assets, which are then placed back into the repaired scene. To support physical interaction, WorldAct also builds simplified collision geometry from the scene,

*Equal contribution.
†Corresponding author: Wei Shen, wei.shen@sjtu.edu.cn

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

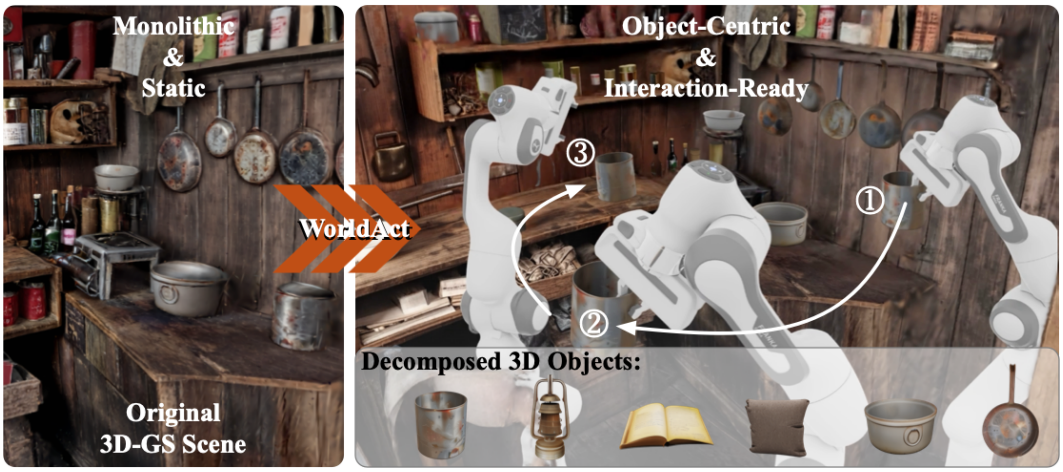


Figure 1: WorldAct converts a monolithic 3DGS scene into a decomposable, object-centric, and interaction-ready environment. By separating individual 3D objects and augmenting them with structures required for physical interaction, our framework enables downstream simulation tasks such as robotic manipulation and scene rearrangement.

enabling stable placement, collision-aware manipulation, and embodied tasks. In this way, WorldAct turns static monolithic generated worlds into structured scenes where individual objects can be edited, moved, and interacted with.

Our key contributions are summarized as follows:

- **Interactive 3D World Modeling.** We propose a framework that converts monolithic 3D generated scenes into decomposed, interaction-ready environments, enabling object-level editing and manipulation.
- **Agent-Driven Automation.** We design an agent-looped pipeline that automatically identifies operable objects, decomposes the scene, restores the background, and reconstructs object assets without manual annotation.
- **Application-Oriented Evaluation.** We evaluate the generated scenes in editing and interaction tasks, demonstrating their visual quality, efficiency, and practical value for downstream applications.

## 2 Related Works

### 2.1 3D Scene Generation

The evolution of 3D representations, from NeRFs [38] to 3D Gaussian Splatting (3DGS) [20], has enabled efficient and photorealistic rendering of complex scenes. Building on these advances, recent generative methods such as LucidDreamer [11], Text2Room [14], Marble [62], and HY-World [18] can synthesize complete 3D worlds from text or images. However, these approaches produce static, monolithic representations in which all scene elements are fused together, limiting object-level editing and interaction.

To address this limitation, compositional approaches generate scenes by assembling individual objects. Some methods [13, 48, 73, 61, 82] generate objects independently before placing them, while others [16, 37, 50] jointly model object generation and layout. Agent-based methods [12, 29, 72, 66] further leverage asset retrieval for scene construction. While these approaches enable object-level controllability and interaction, they typically rely on limited-view inputs or predefined assets, making it difficult to generate large-scale, multi-view consistent environments with high photorealism.

### 2.2 Scene Decomposition and Restoration

Decomposing a fused 3D scene into individual objects is a key step toward interaction. Recent advances in 2D segmentation, such as SAM [21], and vision-language models, such as CLIP [46], have inspired a line of methods that lift 2D masks into 3D, including LangSplat [44], Feature3DGS [83], and related works [75, 77, 3, 35, 2]. These methods provide useful object-level partitions, but the extracted objects are often incomplete, as they mainly consist of visible Gaussians and lack occluded geometry or clean mesh representations. Meanwhile, removing objects from the scene leaves holes in the background, which can be partially addressed by 3D inpainting methods [7, 58, 39, 33, 59, 15]. However, completing large missing regions while preserving scene consistency remains challenging.

### 2.3 Object-Level 3D Generation.

Object-level 3D generation has evolved from SDS-based text-to-3D optimization with frozen 2D diffusion priors [43, 57, 27, 5, 60, 53, 76, 54] to image-conditioned asset generation and reconstruction from single or multi-view inputs [36, 68, 55, 34, 31, 69, 23]. Although effective, these 2D-prior-based methods often face limited 3D consistency or expensive optimization. Recent native 3D generative models instead learn directly over 3D representations such as point clouds, voxels, meshes, 3D Gaussians, and neural fields [40, 56, 79, 47, 67, 70], enabling more efficient geometry generation and textured asset synthesis [8, 64, 65, 26, 74, 80, 9, 17, 22, 24, 84, 28, 63]. In particular, SAM3D [6] improves object asset generation under occlusion, making it useful for reconstructing clean objects from complex indoor scenes.

## 3 Preliminaries

### 3.1 3D Gaussian Splatting

3D Gaussian Splatting (3DGS) [20] represents a continuous 3D scene as an explicit set of unstructured colored Gaussian primitives. Let $\mathcal{G} = \{\mathbf{g}_i\}_{i=1}^{N}$ denote a 3DGS scene with $N$ Gaussians, where each primitive is parameterized as

$$\mathbf{g}_i = (\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i, \alpha_i, \mathbf{c}_i). \tag{1}$$

Here, $\boldsymbol{\mu}_i \in \mathbb{R}^3$ is the 3D center, $\boldsymbol{\Sigma}_i \in \mathbb{R}^{3\times3}$ is the anisotropic covariance, $\alpha_i \in [0, 1]$ is the opacity, and $\mathbf{c}_i$ denotes the color feature. For rendering, Gaussians are projected to the image plane and accumulated by differentiable alpha blending:

$$\mathbf{C}(\mathbf{p}) = \sum_{i=1}^{K} \mathbf{c}_i \alpha_i'(\mathbf{p}) \prod_{j=1}^{i-1} \left(1 - \alpha_j'(\mathbf{p})\right), \tag{2}$$

where $\mathbf{p}$ is a pixel, $K$ is the number of depth-ordered Gaussians overlapping $\mathbf{p}$, and $\alpha_i'(\mathbf{p})$ is the effective opacity of the $i$-th projected Gaussian.

In this work, we mainly consider 3D worlds represented by 3DGS. Unless otherwise specified, a generated 3D world is denoted as a Gaussian set $\mathcal{G}$, which serves as the renderable visual representation of the scene.

### 3.2 3D World Models

Recent 3D world models aim to generate large-scale, navigable, and spatially coherent 3D environments from sparse conditions such as text, images, videos, or panoramas [78, 18, 49, 10, 14, 11, 51, 62]. Such models can be abstracted as a conditional generator $\Phi : \mathcal{X} \to \mathcal{W}$, where $\mathcal{X}$ denotes the input condition and $\mathcal{W}$ denotes the generated 3D world. Generally, the generated world is represented as a 3DGS scene.

Although existing 3D world models have shown impressive visual fidelity, their outputs are still monolithic visual assets rather than interactive-ready environments. Ideally, a 3D world should provide object-level entities, surface or proxy geometry. For a 3DGS-based world, object-level entities can be viewed as a partition of Gaussian primitives:

$$\mathcal{G} = \mathcal{G}_{\text{bg}} \cup \mathcal{G}_1 \cup \cdots \cup \mathcal{G}_M, \quad \mathcal{G}_m \cap \mathcal{G}_n = \varnothing, \tag{3}$$

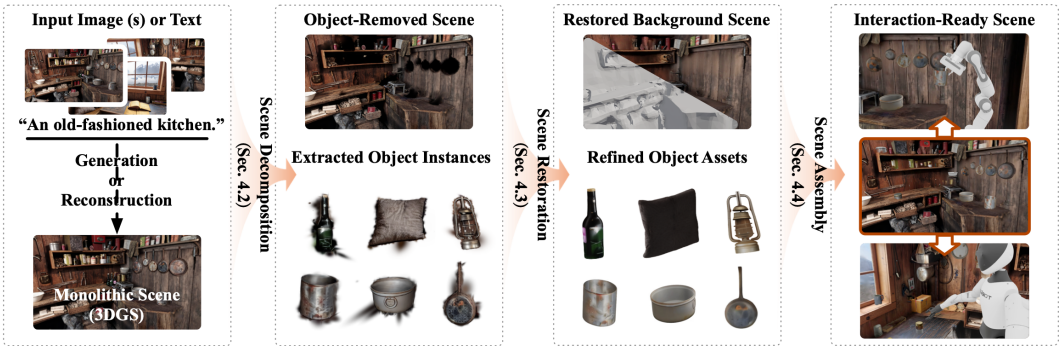


Figure 2: WorldAct first decomposes a generated or reconstructed 3DGS scene into an object-removed background and a set of extracted object instances. It then restores the incomplete background, reconstructs scene-level collision geometry, and refines the extracted instances into clean object assets. Finally, WorldAct assembles these assets back into the restored scene, producing an interaction-ready environment with independent object representations.

where $\mathcal{G}_{\text{bg}}$ denotes the background and each $\mathcal{G}_m$ corresponds to an independently editable object. However, standard 3D world models do not directly provide such primitive-to-entity assignments. Moreover, raw 3DGS scenes do not explicitly encode watertight surfaces, collision proxies, or physical properties such as mass, friction, and support relations. Therefore, despite being visually plausible, existing generated 3D worlds are not directly suitable for downstream tasks like embodied simulation.

# 4 Method

In this section, we first present the overall pipeline of WorldAct, followed by a detailed explanation of each stage.

## 4.1 Pipeline Overview

As shown in Figure 2, WorldAct converts a monolithic 3D Gaussian Splatting (3DGS) scene $\mathcal{G}$, either generated from text/images or reconstructed from multi-view observations, into an interaction-ready, object-decomposed environment. It first decomposes the scene into an object-removed background and extracted object instances via agent-guided multi-view segmentation and 2D-to-3D mask lifting. The background is then restored with scene-level collision geometry, while the extracted instances are refined into clean object assets. Finally, WorldAct aligns and assembles these assets into a restored scene, where the background and objects are independently represented for editing, manipulation, and embodied interaction.

## 4.2 Scene Decomposition

Given a monolithic 3DGS scene $\mathcal{G}$ produced by a 3D world model, WorldAct first renders a camera trajectory to obtain multi-view observations for object discovery and segmentation. Specifically, we define a camera trajectory $\mathcal{T} = \{\mathbf{T}_t\}_{t=1}^{T}$ that navigates through the scene, capturing a video sequence of RGB frames $\{\mathbf{I}_t\}_{t=1}^{T}$ along with their camera poses.

### 4.2.1 Agent-Driven Interactable Object Discovery

To automate object discovery without manual annotation, we employ a vision-language agent (*e.g.*, Qwen3.6-Plus [45]) that analyzes a sparse set of keyframes sampled from the trajectory. As shown in Figure 3, the agent proposes manipulable objects visible in these observations and generates a text prompt list $\mathcal{P} = \{p_1, \ldots, p_N\}$, where each $p_n$ corresponds to a distinct object prompt such as "jar" or "pillow". The agent also filters out objects that are semantically irrelevant for interaction.

For each prompt $p_n$ in $\mathcal{P}$, we perform video segmentation using SAM3 [1], a promptable segmentation foundation model. We prompt SAM3 with the object's semantic label. The model processes each

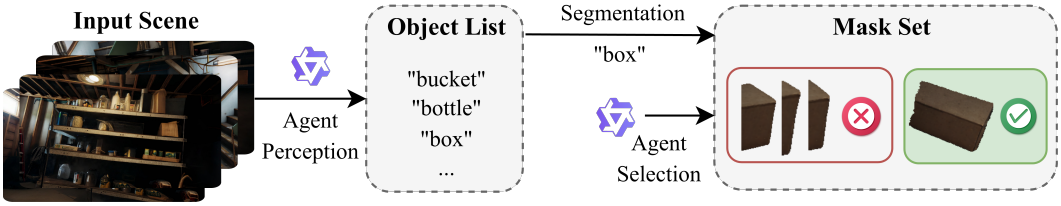


Figure 3: Agent-driven object discovery and best frame selection in WorldAct. The agent identifies interactable objects from rendered scene observations and selects the best object view from multi-view masks for reliable 3D asset generation.

frame $\mathbf{I}_t$ to produce a binary mask $\mathbf{M}_{t,m} \in \{0,1\}^{H\times W}$ indicating the pixel region occupied by the object corresponding to $p_m$. After processing all prompts, we obtain an object list $\mathcal{O} = \{o_1, \ldots, o_M\}$. The output of this stage is a set of multi-view mask sequences $\{\mathbf{M}_{t,m}\}_{t=1}^{T}$ for each object $o_m$, which serve as the input to the subsequent 3D decomposition stage.

#### 4.2.2 Object-Level 3DGS Segmentation

Given multi-view masks for each object, WorldAct decomposes the input 3DGS scene into object-level Gaussian subsets and a residual background. We denote the input scene as $\mathcal{G} = \{g_i\}_{i=1}^{N}$, where each Gaussian $g_i$ contains its geometry, opacity, and appearance attributes. For object $o_m$, we estimate a Gaussian subset

$$\mathcal{G}_m = \{g_i \in \mathcal{G} \mid z_{i,m} = 1\}, \tag{4}$$

where $z_{i,m} \in \{0,1\}$ indicates whether $g_i$ belongs to $o_m$.

Following SA3D [3], we propose a learnable soft assignment score $s_{i,m} \in [0,1]$ for each Gaussian and optimize it through mask inverse rendering. For a view $t \in \mathcal{V}_m$, let $\mathbf{M}_{t,m} \in \{0,1\}^{H\times W}$ be the 2D mask of object $o_m$. The rendered soft mask is computed as

$$\hat{\mathbf{M}}_{t,m}(\mathbf{r}) = \sum_{i=1}^{N} w_i^t(\mathbf{r})\, s_{i,m}, \tag{5}$$

where $\mathbf{r}$ is a pixel ray and $w_i^t(\mathbf{r})$ is the 3DGS alpha-compositing weight of Gaussian $g_i$ on this ray. We optimize $s_{i,m}$ with the projection loss

$$\mathcal{L}_{\text{seg}}^{m} = \sum_{t\in\mathcal{V}_m} \sum_{\mathbf{r}\in\mathcal{R}(\mathbf{I}_t)} \left[-\mathbf{M}_{t,m}(\mathbf{r})\hat{\mathbf{M}}_{t,m}(\mathbf{r}) + \lambda\big(1-\mathbf{M}_{t,m}(\mathbf{r})\big)\hat{\mathbf{M}}_{t,m}(\mathbf{r})\right], \tag{6}$$

where the first term encourages foreground consistency and the second term suppresses false positives in background regions. During optimization, the 3DGS parameters are fixed and only the assignment scores are updated. After convergence, we binarize the scores by a threshold $\tau$:

$$z_{i,m} = \begin{cases} 1, & s_{i,m} > \tau, \\ 0, & \text{otherwise}, \end{cases} \qquad \mathcal{G}_m = \{g_i \in \mathcal{G} \mid z_{i,m} = 1\}. \tag{7}$$

After all objects are processed, the background is defined as $\mathcal{G}_{\text{bg}} = \mathcal{G} \setminus \bigcup_{m=1}^{M} \mathcal{G}_m$. Since $\mathcal{G}_m$ may still be noisy or incomplete due to occlusion and segmentation errors, we use it only as a spatial proxy for object localization and regenerate clean object assets in the following stage.

### 4.3 Scene Restoration

#### 4.3.1 Background Completion

After removing the object Gaussians, the residual background $\mathcal{G}_{\text{bg}}$ contains missing regions at the removed object locations. To complete the background, we first build temporally and geometrically consistent removal masks. Given the multi-view object masks $\{\mathbf{M}_{t,m}\}$, we fuse them into a 3D mask representation through Gaussian splatting reprojection and render it back to each view, obtaining complete masks $\{\mathbf{M}_t^{\text{comp}}\}$ along the trajectory. We then apply Diffueraser [25] to the rendered video

$\{\mathbf{I}_t\}$ with the complete masks $\{\mathbf{M}_t^{\text{comp}}\}$, producing inpainted frames $\{\mathbf{I}_t^{\text{inp}}\}$. To lift the inpainted content back to 3D, we select sparse keyframes, estimate their depths using DepthLab [32], and initialize new Gaussians from the predicted depths. Following Infusion [33], these Gaussians are then optimized to match the inpainted keyframes, yielding a complete background representation $\mathcal{G}_{\text{bg}}^{\text{comp}}$.

To enable physical interaction, we further construct a lightweight collision proxy from $\mathcal{G}_{\text{bg}}^{\text{comp}}$. We extract a watertight mesh using Poisson reconstruction [19], then simplify the mesh and regularize major planar structures using plane detection. Specifically, we perform iterative RANSAC to identify planes from uniformly sampled mesh points, classify them by normal orientation (floors/walls/ceilings), and project nearby vertices onto the detected planes to enforce planarity. The resulting low-polygon mesh $\mathcal{M}_{\text{bg}}$ approximates the background geometry and is used for stable placement and collision-aware simulation.

#### 4.3.2 Agent-Driven Object Generation

After background repair, we focus on generating high-quality assets for 3D objects. Due to occlusion and incomplete observations in the original scene, the isolated Gaussians $\mathcal{G}_m$ are often incomplete and not directly usable for interaction. Instead, we adopt SAM3D [6], a feed-forward model that generates complete 3DGS and mesh assets from single-view RGB images and masks. However, not all viewpoints are equally suitable for generation, as occlusion or unfavorable angles can degrade the output. To address this, we employ an agent to automatically select the optimal viewpoint for each object by evaluating visibility, occlusion levels, and semantic confidence across all frames in the trajectory, as illustrated in Figure 3. The agent then feeds the selected RGB image and its corresponding mask into SAM3D, which produces a clean 3DGS representation $\mathcal{G}_m^{\text{gen}}$ and a textured mesh $\mathcal{M}_m^{\text{gen}}$ for object $o_m$.

### 4.4 Scene Assembly

Although SAM3D provides an estimated pose for each generated object, we observe that the predicted pose is often inaccurate and may not align well with the restored scene. To place each generated object into the completed background $\mathcal{G}_{\text{bg}}^{\text{comp}}$, we use a two-stage alignment procedure.

First, we estimate an initial pose using the extracted object Gaussians $\mathcal{G}_m$ as spatial anchors. Given the generated object mesh $\mathcal{M}_m^{\text{gen}}$, we perform Iterative Closest Point (ICP) between $\mathcal{M}_m^{\text{gen}}$ and the point set derived from $\mathcal{G}_m$ under multiple candidate transformations. For each candidate pose, we render the placed object and compare it with the original object observations using DINOv2 [42] features. The pose with the highest feature similarity is selected as the initialization.

Second, we refine the object pose through differentiable rendering. For each object $o_m$, we optimize its translation $\mathbf{t}_m \in \mathbb{R}^3$, rotation represented in 6D form $\mathbf{r}_m \in \mathbb{R}^6$, and scale $s_m \in \mathbb{R}^+$. The optimization minimizes

$$\mathcal{L}_{\text{align}} = \mathcal{L}_{\text{mask}} + w_c \mathcal{L}_{\text{contact}} + w_p \mathcal{L}_{\text{penetration}}, \tag{8}$$

Here, $\mathcal{L}_{\text{mask}}$ combines soft Dice and focal losses between the rendered object silhouettes and the original object masks across sampled views. $\mathcal{L}_{\text{contact}}$ applies a smooth L1 penalty to small positive vertical gaps between the object and nearby background surfaces, encouraging support contact. $\mathcal{L}_{\text{penetration}}$ identifies penetrating object vertices using nearest background surface samples and their oriented normals, and penalizes the distances to those samples. We use $w_c = 0.01$ and $w_p = 0.02$.

After alignment, the final scene consists of the completed background $\mathcal{G}_{\text{bg}}^{\text{comp}}$ with its collision mesh $\mathcal{M}_{\text{bg}}$, together with a set of generated object assets $\{(\mathcal{G}_m^{\text{gen}}, \mathcal{M}_m^{\text{gen}})\}_{m=1}^{M}$ placed in the scene. This decomposed representation supports object-level editing, manipulation, and embodied task execution.

## 5 Experiments

### 5.1 Implementation Details

All experiments are conducted on a single NVIDIA RTX 3090 GPU. Converting a 3DGS scene typically takes around 1 hour, varying with scene complexity.




Figure 4: Qualitative comparison with input scenes. For each scene, we show three different viewpoints, each with the original input and our decomposed interactive output. Our method preserves visual fidelity while enabling object-level decomposition and interaction.

We evaluate on six indoor scenes generated by Marble [62], including kitchens, restrooms, and storage rooms. Marble accepts text, single-image, or multi-image inputs and produces monolithic 3DGS scenes. We additionally evaluate on four scenes generated by HY-World 2.0 [18] to assess performance across foundation models.

As ground-truth decomposed assets and completed backgrounds are unavailable, we primarily use reference-free metrics to assess conversion quality. ReMOVE [4, 81] measures foreground-background consistency after object removal, while MANIQA [71] evaluates perceptual image quality. For decomposition, we report *Interactable Object Recall*, the fraction of manually annotated interactable objects that are successfully extracted. We further evaluate scene and object quality through a Mean Opinion Score (MOS) study with 20 participants using a 5-point Likert scale across four dimensions: overall quality, surface completeness, boundary cleanliness, and naturalness. As an auxiliary evaluator, GPT-5.5 [41] rates matched scene and object renderings from the original and converted Marble scenes using the same criteria. As a supplementary reference-based comparison, we report PSNR, SSIM, and LPIPS on selected keyframes, using the original 3DGS renderings from the same viewpoints as references.

**Qualitative Results.** Figure 4 compares the input scenes with the converted scenes from multiple viewpoints. After decomposition and regeneration, objects are placed back into their original locations within the restored background. The results retain the overall scene layout and show coherent object boundaries across views. At the object level, regeneration completes some missing surfaces and reduces visible deformations and blurriness, while allowing the resulting assets to be edited independently. Figure 5 shows the extracted meshes alongside the inpainted backgrounds. The



Figure 5: Qualitative results of background mesh reconstruction. Top: background renderings after object removal and inpainting. Bottom: the corresponding reconstructed meshes used as background collision geometry.

Table 1: **Interactable Object Recall Rate.** Object discovery recall on Marble and HY-World. Higher is better.

| Method | Marble (%) ↑ | HY-World (%) ↑ |
|---|---|---|
| Ours w/o Agent | 23.77 | 8.10 |
| Ours | **78.80** | **70.97** |

Table 2: **Perceptual quality.** ReMOVE [4, 81] and MANIQA [71] at different conversion stages. Higher is better.

| | Marble (6) | | HY-World (4) | |
|---|---|---|---|---|
| Stage | ReMOVE | MANIQA | ReMOVE | MANIQA |
| Original 3DGS | 0.7933 | 0.3338 | 0.7393 | 0.3924 |
| Object re-insertion | **0.7934** | **0.3622** | **0.7501** | **0.3951** |
| Inpainting (Gaussian Grouping) | 0.7948 | 0.3227 | 0.7567 | 0.3729 |
| Inpainting (ours) | **0.7951** | **0.3254** | **0.7583** | **0.3740** |

Table 3: **Reference-based comparison.** Results on six Marble and four HY-World scenes, using one keyframe per scene.

| | Marble (6) | | | HY-World (4) | | |
|---|---|---|---|---|---|---|
| Method | PSNR ↑ | SSIM ↑ | LPIPS ↓ | PSNR ↑ | SSIM ↑ | LPIPS ↓ |
| SAM3D | 24.33 | 0.890 | 0.107 | 25.18 | 0.920 | 0.060 |
| MIDI | 21.72 | 0.841 | 0.179 | 23.90 | 0.920 | 0.078 |
| Ours | **25.26** | **0.903** | **0.089** | **25.53** | **0.925** | **0.058** |

meshes align well with the restored scene geometry and exhibit relatively smooth surfaces, providing collision geometry for downstream simulation.

**Quantitative Results.** Table 1 reports the Interactable Object Recall Rate. Agent guidance increases recall from 23.77% to 78.80% on Marble and from 8.10% to 70.97% on HY-World, showing its effectiveness in identifying interactable objects across both datasets. Objects missed during discovery remain in the background and are not converted into independently editable assets.

Table 2 summarizes perceptual quality at different conversion stages on Marble and HY-World. Object re-insertion improves MANIQA on both datasets. We also include Gaussian Grouping as a reference for evaluating background inpainting.

WorldAct converts existing 3D scenes into interactive environments, a task that differs from 3D generation and makes direct comparison challenging. Nevertheless, we provide a supplementary reference-based comparison with SAM3D [6] and MIDI [16] on keyframes from six Marble and four

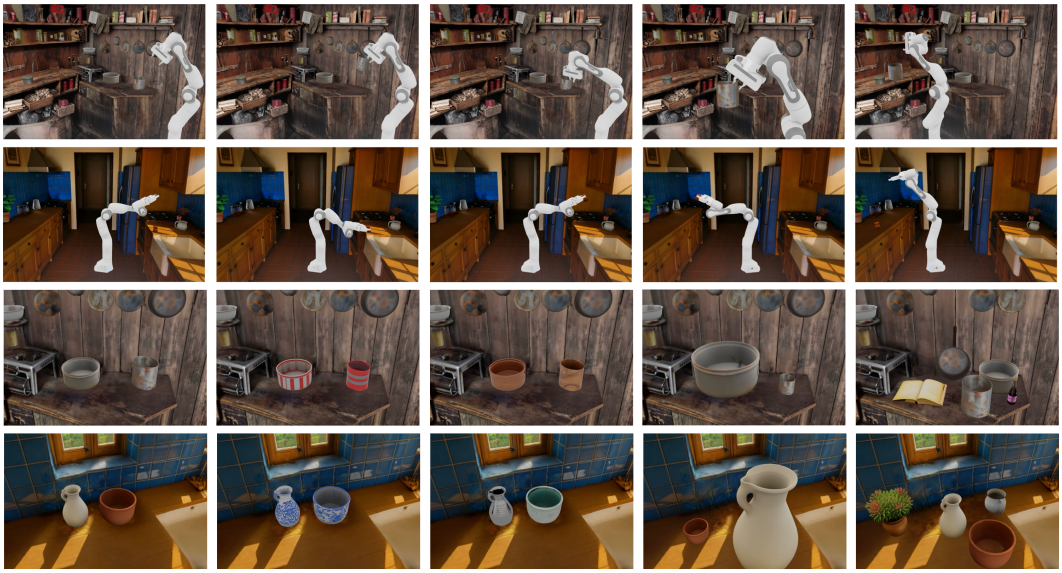

Figure 6: Interactive examples with WorldAct. By decomposing a generated 3DGS scene into editable object assets and a restored background, WorldAct enables object-level interaction in 3D worlds. Users can add, place, remove, and modify objects, including size, texture, and material. These capabilities support embodied simulation, scene rearrangement, and interactive content creation.

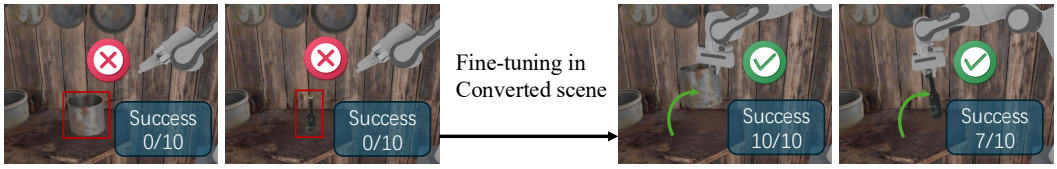


Figure 7: Grasping before and after policy fine-tuning in our converted scene. The pretrained policy fails to grasp the cup and thin bottle, while the fine-tuned policies successfully grasp and lift them.

HY-World scenes. Baseline object renderings are composited with the original background. Table 3 reports PSNR, SSIM, and LPIPS computed against the input renderings from the same viewpoints.

### 5.2 Interactive Experiments

**Application to Embodied Simulation.** WorldAct converts a generated 3D scene into an interaction-ready environment for embodied simulation. As shown in the first and second rows of Figure 6, after decomposing the scene into explicit object assets and a restored background, our framework supports object-level physical interaction, like grasping and placement. We evaluate language-conditioned grasping in our constructed Isaac Sim environment using a Franka arm and a SmolVLA policy [52] initialized from a LIBERO-trained checkpoint [30]. Given instructions such as "Pick up the cup," the pretrained policy achieves 0/10 successful grasps for both the cup and the thin bottle. For each object, a scripted controller automatically collects 24 successful expert trajectories, split into 20 for training and 4 for validation. We then fine-tune a separate policy for each object for 4,000 steps. Evaluated under the same initial conditions as the pretrained policy, the fine-tuned policies achieve 10/10 successful grasps for the cup and 7/10 for the thin bottle (Fig. 7). These results illustrate how the constructed environment supports automated demonstration collection and adaptation of pretrained manipulation policies.

**Application to High-Quality 3D Scene Editing.** WorldAct supports adding, removing, and relocating objects, as well as editing their scale, texture, and material (Figure 6, third and fourth rows). The examples show restored backgrounds after removal and coherent placement of inserted or modified assets.

Table 4: **Human and GPT-5.5 ratings.** Each cell reports *Human / GPT* scores on a 1–5 scale, where higher is better. The human study includes 20 participants; the GPT evaluation uses 50 matched pairs per level. Both evaluations cover the same six Marble scenes and are aggregated separately.

| Evaluated Target | Overall Quality | Surface Comp. | Boundary Clean. | Naturalness |
|---|---|---|---|---|
| *Scene-Level Evaluation* | | | | |
| Original Scene | 4.41 / 3.04 | 4.44 / 3.62 | 4.28 / 2.90 | 4.20 / 3.06 |
| Converted Scene | 4.13 / 2.78 | 4.31 / 3.26 | 4.02 / 2.60 | 3.78 / 2.78 |
| *Object-Level Evaluation* | | | | |
| Original Object | 2.49 / 1.78 | 2.58 / 2.24 | 2.00 / 1.46 | 2.14 / 1.88 |
| Regenerated Object | 3.75 / 2.50 | 3.93 / 3.20 | 3.95 / 2.42 | 3.37 / 2.62 |

### 5.3 User Study

We conduct a Mean Opinion Score (MOS) study with 20 participants on the six Marble scenes. Participants rate anonymized scene and object renderings on a five-point scale across four criteria: Overall Quality, Surface Completeness, Boundary Cleanliness, and Naturalness. We additionally use GPT-5.5 to evaluate 50 matched scene pairs and 50 matched object pairs from the same scenes, using the same criteria and scale. Human ratings serve as the primary subjective evaluation, with GPT ratings providing an auxiliary assessment. The evaluation protocols and GPT prompt are provided in Appendix C.

Table 4 reports both human and GPT ratings. Both evaluations show higher scores for the regenerated objects across all four criteria, alongside lower scores for the converted scenes. For example, object surface completeness increases from 2.58 to 3.93 in the human study and from 2.24 to 3.20 in the GPT evaluation. Scene naturalness decreases from 4.20 to 3.78 and from 3.06 to 2.78, respectively. These results indicate that object regeneration improves the quality of individual assets, while the conversion introduces some loss in scene-level perceptual quality.

## 6 Conclusion

In this paper, we present WorldAct, a framework that converts monolithic 3DGS scenes into object-decomposed environments for editing and interaction. WorldAct identifies objects in a generated scene, separates them from the background, repairs the remaining scene, regenerates cleaner object assets, and aligns them with simple collision geometry. Our experiments show that the pipeline preserves the visual appearance of the original scene while enabling basic object-level editing, placement, and embodied interaction.

**Limitations.** Physical parameters such as mass and friction are currently specified manually, and interactions are limited to rigid objects, without support for articulated objects or more complex dynamics. The pipeline also requires multiple processing stages. Future work will focus on estimating physical properties, extending interaction capabilities, and integrating agent reasoning more closely with 3D generation, while streamlining the pipeline to construct interactive environments more efficiently.

## Acknowledgments and Disclosure of Funding

This work was supported by the NSFC under Grants 62322604 and 62576207. The authors express gratitude to the anonymous reviewers for their valuable feedback.

## References


[1] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. In *ICLR*, 2026.

[2] Jiazhong Cen, Jiemin Fang, Chen Yang, Lingxi Xie, Xiaopeng Zhang, Wei Shen, and Qi Tian. Segment any 3d gaussians. In *AAAI*, 2025.

[3] Jiazhong Cen, Zanwei Zhou, Jiemin Fang, Chen Yang, Wei Shen, Lingxi Xie, Dongsheng Jiang, Xiaopeng Zhang, and Qi Tian. Segment anything in 3d with nerfs. In *NeurIPS*, 2023.

[4] Aditya Chandrasekar, Goirik Chakrabarty, Jai Bardhan, Ramya Hebbalaguppe, and Prathosh AP. Remove: A reference-free metric for object erasure. In *CVPR*, 2024.

[5] Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. In *ICCV*, 2023.

[6] 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. *arXiv preprint arXiv:2511.16624*, 2025.

[7] Yiwen Chen, Zilong Chen, Chi Zhang, Feng Wang, Xiaofeng Yang, Yikai Wang, Zhongang Cai, Lei Yang, Huaping Liu, and Guosheng Lin. Gaussianeditor: Swift and controllable 3d editing with gaussian splatting. In *CVPR*, 2024.

[8] Yiwen Chen, Zhihao Li, Yikai Wang, Hu Zhang, Qin Li, Chi Zhang, and Guosheng Lin. Ultra3d: Efficient and high-fidelity 3d generation with part attention. *arXiv preprint arXiv:2507.17745*, 2025.

[9] Zhaoxi Chen, Jiaxiang Tang, Yuhao Dong, Ziang Cao, Fangzhou Hong, Yushi Lan, Tengfei Wang, Haozhe Xie, Tong Wu, Shunsuke Saito, et al. 3dtopia-xl: Scaling high-quality 3d asset generation via primitive diffusion. In *CVPR*, 2025.

[10] Jisheng Chu, Wenrui Li, Rui Zhao, Wangmeng Zuo, Shifeng Chen, and Xiaopeng Fan. Roamscene3d: Immersive text-to-3d scene generation via adaptive object-aware roaming. *arXiv preprint arXiv:2601.19433*, 2026.

[11] Jaeyoung Chung, Suyoung Lee, Hyeongjin Nam, Jaerin Lee, and Kyoung Mu Lee. Luciddreamer: Domain-free generation of 3d gaussian splatting scenes. *TVCG*, 2025.

[12] Tianyuan Dai, Josiah Wong, Yunfan Jiang, Chen Wang, Cem Gokmen, Ruohan Zhang, Jiajun Wu, and Li Fei-Fei. Automated creation of digital cousins for robust policy learning. In *CoRL*, 2024.

[13] Wenqi Dong, Bangbang Yang, Zesong Yang, Yuan Li, Tao Hu, Hujun Bao, Yuewen Ma, and Zhaopeng Cui. Hiscene: creating hierarchical 3d scenes with isometric view generation. In *ACMMM*, 2025.

[14] Lukas Höllein, Ang Cao, Andrew Owens, Justin Johnson, and Matthias Nießner. Text2room: Extracting textured 3d meshes from 2d text-to-image models. In *ICCV*, 2023.

[15] Sheng-Yu Huang, Zi-Ting Chou, and Yu-Chiang Frank Wang. 3d gaussian inpainting with depth-guided cross-view consistency. In *CVPR*, 2025.

[16] 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 *CVPR*, 2025.

[17] Team Hunyuan3D, Shuhui Yang, Mingxin Yang, Yifei Feng, Xin Huang, Sheng Zhang, Zebin He, Di Luo, Haolin Liu, Yunfei Zhao, et al. Hunyuan3d 2.1: From images to high-fidelity 3d assets with production-ready pbr material. *arXiv preprint arXiv:2506.15442*, 2025.

[18] Team HY-World, Chenjie Cao, Xuhui Zuo, Zhenwei Wang, Yisu Zhang, Junta Wu, Zhenyang Liu, Yuning Gong, Yang Liu, Bo Yuan, et al. Hy-world 2.0: A multi-modal world model for reconstructing, generating, and simulating 3d worlds. *arXiv preprint arXiv:2604.14268*, 2026.

[19] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *SGP*, 2006.

[20] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023.

[21] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *ICCV*, 2023.

[22] 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.

[23] Jiahao Li, Hao Tan, Kai Zhang, Zexiang Xu, Fujun Luan, Yinghao Xu, Yicong Hong, Kalyan Sunkavalli, Greg Shakhnarovich, and Sai Bi. Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model. *arXiv preprint arXiv:2311.06214*, 2023.

[24] Weiyu Li, Xuanyang Zhang, Zheng Sun, Di Qi, Hao Li, Wei Cheng, Weiwei Cai, Shihao Wu, Jiarui Liu, Zihao Wang, et al. Step1x-3d: Towards high-fidelity and controllable generation of textured 3d assets. *arXiv preprint arXiv:2505.07747*, 2025.

[25] Xiaowen Li, Haolan Xue, Peiran Ren, and Liefeng Bo. Diffueraser: A diffusion model for video inpainting. *arXiv preprint arXiv:2501.10018*, 2025.

[26] Zhihao Li, Yufei Wang, Heliang Zheng, Yihao Luo, and Bihan Wen. Sparc3d: Sparse representation and construction for high-resolution 3d shapes modeling. *arXiv preprint arXiv:2505.14521*, 2025.

[27] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *CVPR*, 2023.

[28] 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*, 2025.

[29] Lu Ling, Chen-Hsuan Lin, Tsung-Yi Lin, Yifan Ding, Yu Zeng, Yichen Sheng, Yunhao Ge, Ming-Yu Liu, Aniket Bera, and Zhaoshuo Li. Scenethesis: Combining language and visual priors for 3d scene generation. In *ICLR*, 2026.

[30] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

[31] Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. Syncdreamer: Generating multiview-consistent images from a single-view image. In *ICLR*, 2024.

[32] Zhiheng Liu, Ka Leong Cheng, Qiuyu Wang, Shuzhe Wang, Hao Ouyang, Bin Tan, Kai Zhu, Yujun Shen, Qifeng Chen, and Ping Luo. Depthlab: From partial to complete. *arXiv preprint arXiv:2412.18153*, 2024.

[33] Zhiheng Liu, Hao Ouyang, Qiuyu Wang, Ka Leong Cheng, Jie Xiao, Kai Zhu, Nan Xue, Yu Liu, Yujun Shen, and Yang Cao. Infusion: Inpainting 3d gaussians via learning depth completion from diffusion prior. *arXiv preprint arXiv:2404.11613*, 2024.

[34] Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, et al. Wonder3d: Single image to 3d using cross-domain diffusion. In *CVPR*, 2024.

[35] Weijie Lyu, Xueting Li, Abhijit Kundu, Yi-Hsuan Tsai, and Ming-Hsuan Yang. Gaga: Group any gaussians via 3d-aware memory bank. *TMLR*, 2026.

[36] Luke Melas-Kyriazi, Iro Laina, Christian Rupprecht, and Andrea Vedaldi. Realfusion: 360deg reconstruction of any object from a single image. In *CVPR*, 2023.

[37] Yanxu Meng, Haoning Wu, Ya Zhang, and Weidi Xie. Scenegen: Single-image 3d scene generation in one feedforward pass. In *3DV*, 2026.

[38] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *CACM*, 2021.

[39] Ashkan Mirzaei, Tristan Aumentado-Armstrong, Konstantinos G Derpanis, Jonathan Kelly, Marcus A Brubaker, Igor Gilitschenski, and Alex Levinshtein. Spin-nerf: Multiview segmentation and perceptual inpainting with neural radiance fields. In *CVPR*, 2023.

[40] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*, 2022.

[41] OpenAI. GPT-5.5 System Card. Technical report, OpenAI, 4 2026.

[42] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[43] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *ICLR*, 2023.

[44] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3d language gaussian splatting. In *CVPR*, 2024.

[45] Qwen Team. Qwen3.6-Plus: Towards real world agents, April 2026.

[46] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[47] Xuanchi Ren, Jiahui Huang, Xiaohui Zeng, Ken Museth, Sanja Fidler, and Francis Williams. Xcube: Large-scale 3d generative modeling using sparse voxel hierarchies. In *CVPR*, 2024.

[48] Tobias Sautter, Jan-Niklas Dihlmann, and Hendrik Lensch. 3d-re-gen: 3d reconstruction of indoor scenes with a generative framework. *arXiv preprint arXiv:2512.17459*, 2025.

[49] Katja Schwarz, Denis Rozumny, Samuel Rota Bulò, Lorenzo Porzi, and Peter Kontschieder. A recipe for generating 3d worlds from a single image. In *ICCV*, 2025.

[50] 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. *arXiv preprint arXiv:2512.10957*, 2025.

[51] Jaidev Shriram, Alex Trevithick, Lingjie Liu, and Ravi Ramamoorthi. Realmdreamer: Text-driven 3d scene generation with inpainting and depth diffusion. In *3DV*, 2025.

[52] Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, et al. Smolvla: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844*, 2025.

[53] Jingxiang Sun, Bo Zhang, Ruizhi Shao, Lizhen Wang, Wen Liu, Zhenda Xie, and Yebin Liu. Dreamcraft3d: Hierarchical 3d generation with bootstrapped diffusion prior. In *ICLR*, 2024.

[54] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. In *ICLR*, 2024.

[55] Junshu Tang, Tengfei Wang, Bo Zhang, Ting Zhang, Ran Yi, Lizhuang Ma, and Dong Chen. Make-it-3d: High-fidelity 3d creation from a single image with diffusion prior. In *ICCV*, 2023.

[56] Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, Karsten Kreis, et al. Lion: Latent point diffusion models for 3d shape generation. *NeurIPS*, 2022.

[57] Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A Yeh, and Greg Shakhnarovich. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In *CVPR*, 2023.

[58] Junjie Wang, Jiemin Fang, Xiaopeng Zhang, Lingxi Xie, and Qi Tian. Gaussianeditor: Editing 3d gaussians delicately with text instructions. In *CVPR*, 2024.

[59] Shaoxiang Wang, Shihong Zhang, Christen Millerdurai, Rüdiger Westermann, Didier Stricker, and Alain Pagani. Inpaint360gs: Efficient object-aware 3d inpainting via gaussian splatting for 360deg scenes. In *WACV*, 2026.

[60] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. In *NeurIPS*, 2023.

[61] Ziqian Wang, Yonghao He, Licheng Yang, Wei Zou, Hongxuan Ma, Liu Liu, Wei Sui, Yuxin Guo, and Hu Su. Tabletopgen: Instance-level interactive 3d tabletop scene generation from text or single image. *arXiv preprint arXiv:2512.01204*, 2025.

[62] World Labs. Marble. `https://www.worldlabs.ai/blog/marble-world-model`, 2025. Accessed: 2026-03-07.

[63] Guanjun Wu, Jiemin Fang, Chen Yang, Sikuang Li, Taoran Yi, Jia Lu, Zanwei Zhou, Jiazhong Cen, Lingxi Xie, Xiaopeng Zhang, et al. Unilat3d: Geometry-appearance unified latents for single-stage 3d generation. *arXiv preprint arXiv:2509.25079*, 2025.

[64] Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Jingxi Xu, Philip Torr, Xun Cao, and Yao Yao. Direct3d: Scalable image-to-3d generation via 3d latent diffusion transformer. In *NeurIPS*, 2024.

[65] Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Yikang Yang, Yajie Bao, Jiachen Qian, Siyu Zhu, Xun Cao, Philip Torr, et al. Direct3d-s2: Gigascale 3d generation made easy with spatial sparse attention. *arXiv preprint arXiv:2505.17412*, 2025.

[66] Hongchi Xia, Xuan Li, Zhaoshuo Li, Qianli Ma, Jiashu Xu, Ming-Yu Liu, Yin Cui, Tsung-Yi Lin, Wei-Chiu Ma, Shenlong Wang, Shuran Song, and Fangyin Wei. Sage: Scalable agentic 3d scene generation for embodied ai. In *CVPR*, 2026.

[67] Bojun Xiong, Si-Tong Wei, Xin-Yang Zheng, Yan-Pei Cao, Zhouhui Lian, and Peng-Shuai Wang. Octfusion: Octree-based diffusion models for 3d shape generation. *Computer Graphics Forum*, 2025.

[68] Dejia Xu, Yifan Jiang, Peihao Wang, Zhiwen Fan, Yi Wang, and Zhangyang Wang. Neurallift-360: Lifting an in-the-wild 2d photo to a 3d object with 360deg views. In *CVPR*, 2023.

[69] 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.

[70] Haitao Yang, Yuan Dong, Hanwen Jiang, Dejia Xu, Georgios Pavlakos, and Qixing Huang. Atlas gaussians diffusion for 3d generation. In *ICLR*, 2025.

[71] Sidi Yang, Tianhe Wu, Shuwei Shi, Shanshan Lao, Yuan Gong, Mingdeng Cao, Jiahao Wang, and Yujiu Yang. Maniqa: Multi-dimension attention network for no-reference image quality assessment. In *CVPR*, 2022.

[72] Yandan Yang, Baoxiong Jia, Shujie Zhang, and Siyuan Huang. Sceneweaver: All-in-one 3d scene synthesis with an extensible and self-reflective agent. In *NeurIPS*, 2025.

[73] 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. *TOG*, 2025.

[74] 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. *arXiv preprint arXiv:2503.22236*, 2025.

[75] Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3d scenes. In *ECCV*, 2024.

[76] Taoran Yi, Jiemin Fang, Junjie Wang, Guanjun Wu, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Qi Tian, and Xinggang Wang. Gaussiandreamer: Fast generation from text to 3d gaussians by bridging 2d and 3d diffusion models. In *CVPR*, 2024.

[77] Haiyang Ying, Yixuan Yin, Jinzhi Zhang, Fan Wang, Tao Yu, Ruqi Huang, and Lu Fang. Omniseg3d: Omniversal 3d segmentation via hierarchical contrastive learning. In *CVPR*, 2024.

[78] Hong-Xing Yu, Haoyi Duan, Charles Herrmann, William T Freeman, and Jiajun Wu. Wonderworld: Interactive 3d scene generation from a single image. In *CVPR*, 2025.

[79] Biao Zhang, Jiapeng Tang, Matthias Niessner, and Peter Wonka. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *ACM TOG*, 2023.

[80] 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 TOG*, 2024.

[81] Jixin Zhao, Shangchen Zhou, Zhouxia Wang, Peiqing Yang, and Chen Change Loy. Objectclear: Complete object removal via object-effect attention. *arXiv preprint arXiv:2505.22636*, 2025.

[82] Junwei Zhou, Xueting Li, Lu Qi, and Ming-Hsuan Yang. Layout-your-3d: Controllable and precise 3d generation with 2d blueprint. In *International Conference on Learning Representations*, volume 2025, pages 2673–2695, 2025.

[83] Shijie Zhou, Haoran Chang, Sicheng Jiang, Zhiwen Fan, Zehao Zhu, Dejia Xu, Pradyumna Chari, Suya You, Zhangyang Wang, and Achuta Kadambi. Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields. In *CVPR*, 2024.

[84] Zanwei Zhou, Taoran Yi, Jiemin Fang, Chen Yang, Lingxi Xie, Xinggang Wang, Wei Shen, and Qi Tian. Few-step flow for 3d generation via marginal-data transport distillation. In *AAAI*, 2026.

# A Overview

This supplementary material provides additional details and analyses to complement the main manuscript. Specifically, Section B presents the detailed design of the agent utilized in Sections 4.2 and 4.3. Furthermore, we elaborate on the design and implementation of the user study in Section C. Section D then showcases additional experimental results of our proposed reconstruction method.

# B Agent Design Details

This section details the implementation of our object- and view-selection agent. The agent automatically identifies portable objects from multi-view observations, extracts object masks via text-guided segmentation, and provides the resulting masks and inpainted videos for downstream reconstruction. The agent comprises three core components: a vision module, a memory module, and an execution module.

## B.1 Overview

Given visual observations $\mathcal{O} = \{I_v\}_{v=1}^{V}$ or a video $\mathcal{V}$, the agent first parses the scene into a structured object inventory. It selects portable objects, generates text prompts for segmentation, invokes SAM3 to extract candidate masks across rendered views, and employs a Vision-Language Model (VLM) to score and select the optimal view for each object. Following mask aggregation and duplicate removal, the agent performs video inpainting to visually erase the selected objects. Formally, the agent outputs a set of object-level representations:

$$\mathcal{Y} = \{(q_i, v_i^\star, M_i, \mathcal{V}_i^{\mathrm{rm}})\}_{i=1}^{N}, \tag{9}$$

where $q_i$ is the text prompt of the $i$-th selected object, $v_i^\star$ is its best view, $M_i$ is the aggregated mask, and $\mathcal{V}_i^{\mathrm{rm}}$ is the inpainted background video. These outputs isolate portable objects from the static scene, facilitating the downstream modeling of interactable 3D environments.

## B.2 Vision Module

The vision module converts raw visual inputs into structured semantic descriptions via the Qwen3.6-Plus API. The VLM is prompted to enumerate all visible objects and classify them along two orthogonal axes: mobility (`portable` vs. `fixed`) and semantic recognizability (`precise`, `subtle`, or `unrecognizable`). This process produces a comprehensive, deduplicated object inventory that systematically accounts for items with ambiguous boundaries or semantics.

## B.3 Memory Module

The memory module maintains the state of the parsed object inventory. For the $i$-th discovered object, it stores a state dictionary:

$$\mathcal{M}_i = \{\texttt{name}_i, \texttt{category}_i, \texttt{count}_i, \texttt{recognizability}_i\}. \tag{10}$$

Serving as the agent's central representation, the memory module supplies the execution module with exact object-level prompts and tracks processing states to guarantee logical consistency across the pipeline.

## B.4 Execution Module

The execution module handles segmentation, view selection, and video inpainting. For each object categorized as `portable` in memory, its `name` serves as a text prompt $q_i$ for SAM3 to generate candidate masks across all available views.

To resolve viewpoint variations, occlusions, and scale inconsistencies, we propose a VLM-based view-scoring mechanism. We prompt Qwen3.6-Plus to evaluate each candidate crop, returning an integer score from 0 to 100 based on the completeness, clarity, and centeredness of the target object. Given the score $s_{i,v}$ for object $i$ in view $v$, the optimal view is selected as:

$$v_i^\star = \arg\max_{v} s_{i,v}. \tag{11}$$

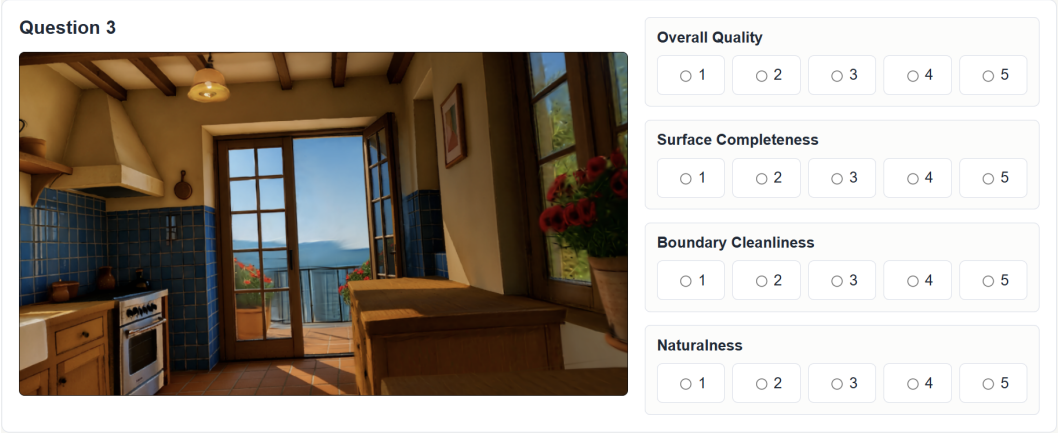


Figure 8: Screenshot of the web-based user-study interface. Participants first read the bilingual evaluation criteria and MOS-style score definitions, then rate one anonymized image at a time.

Candidate masks for the same object are subsequently aggregated. After filtering out duplicate masks to prevent redundant selection, the final mask set guides the video inpainting process, yielding a clean background video.

### B.5 Agent Workflow and Interface to Reconstruction

The complete agent pipeline is summarized in Alg. 1. The final outputs—best-view masks and inpainted background videos—provide explicit object-level supervision. By cleanly separating portable objects from the static environment prior to reconstruction, our approach significantly reduces the entanglement between object geometry, appearance, and the background, directly supporting the downstream decomposition of interactive 3D scenes.

**Algorithm 1** Object-Selection Agent Workflow

**Require:** Input video $\mathcal{V}$ or rendered views $\mathcal{O}$; SAM3 threshold $\tau$
**Ensure:** Object masks $M$, best views $v^\star$, and inpainted videos $\mathcal{V}^{\mathrm{rm}}$

Parse $\mathcal{V}$ or $\mathcal{O}$ into a structured object inventory via the vision module
Initialize memory module $\mathcal{M}$ with object properties (`name`, `category`, etc.)
**for** each `portable` object $i$ in $\mathcal{M}$ **do**
    Generate candidate masks across views using SAM3 with prompt $q_i = \texttt{name}_i$
    Score candidate crops via VLM to evaluate target completeness and clarity
    Select the best view $v_i^\star = \arg\max_v s_{i,v}$
    Aggregate candidate masks corresponding to object $i$
**end for**
Filter duplicate object masks
Generate inpainted video $\mathcal{V}^{\mathrm{rm}}$ using the aggregated masks
**return** $\{(q_i, v_i^\star, M_i, \mathcal{V}_i^{\mathrm{rm}})\}_{i=1}^{N}$

## C User Study and Auxiliary Agent Evaluation

### C.1 Human User Study

We conducted a blind user study to evaluate the perceptual quality of the rendered results produced by Marble and our method. The study was implemented as a web-based questionnaire. Each questionnaire contained 20 single-image questions, consisting of 10 whole-scene renderings and 10 object-level renderings. We distributed 20 questionnaires in total.

We evaluate all the scene in the Marble dataset. For each question, the participant saw only one anonymized rendering. The questionnaire did not reveal the scene name, object identity, frame index, or method label. These metadata were stored only on the server for later aggregation.

To reduce recognition bias and direct pairwise comparison effects, each questionnaire was generated using randomized sampling under two constraints. First, within a single questionnaire, each scene was assigned to only one method, either Marble or ours. Therefore, a participant never saw both methods for the same scene in the same questionnaire. Second, whole-scene images sampled from the same scene were required to be temporally separated by at least 10 frames. For object-level questions, each object identity appeared at most once in a questionnaire and was shown using only one method.

Each image was rated along four perceptual criteria:

- **Overall Quality.** The participant judged the result based on the first overall visual impression, considering whether the image appeared clear, stable, plausible, and whether artifacts affected the viewing experience.
- **Surface Completeness.** The participant assessed whether the visible geometric surfaces were continuous and complete. For scene images, this criterion focused on obvious holes, discontinuities, or missing regions. For object images, it focused on missing, broken, or implausibly incomplete object surfaces.
- **Boundary Cleanliness.** The participant assessed whether boundaries between structures were clear and free from visible contamination. For scene images, this criterion focused on boundaries between objects and between objects and the background. For object images, it focused on whether the object's silhouette and outer boundary were clean and stable.
- **Naturalness.** The participant assessed whether the result was consistent with real visual experience. For scene images, this criterion focused on whether reconstructed or inserted regions were coherent with the surrounding scene in appearance, scale, and style. For object images, it focused on whether the object's shape and appearance resembled a normal, naturally occurring object.

All criteria were scored using a five-point scale following the Mean Opinion Score (MOS) protocol commonly used in subjective perceptual quality assessment. A score of 5 indicated excellent quality with almost no visible distortion; 4 indicated good quality with only minor distortion; 3 indicated fair quality with noticeable but still acceptable distortion; 2 indicated poor quality with severe distortion; and 1 indicated bad quality that was difficult to accept. The questionnaire instructions and score definitions were provided in both Chinese and English, and participants could choose either language before starting the study.

Because object-level renderings may occupy only a small region of the image canvas, the interface provided a local zoom and panning function for object questions. This function allowed participants to inspect object boundaries and local surface details more reliably. The zoom function did not reveal any hidden metadata and was not used for scene-level questions.

For each submitted questionnaire, the backend recorded the anonymous question identifier, hidden image type, hidden method label, hidden scene/object identifier, and the four criterion scores. After collection, we computed average scores and sample counts for each scene-method pair, each object-method pair, all scene images grouped by method, and all object images grouped by method. Aggregation was performed separately for each criterion and for the mean score across the four criteria.

### C.2 Auxiliary Agent Evaluation

In addition to the human user study, we used **GPT-5.5** for an auxiliary assessment of scene and object quality. The evaluation followed the same four criteria and five-point scale as the human study: Overall Quality, Surface Completeness, Boundary Cleanliness, and Naturalness. Human ratings remained the primary measure of subjective quality.

We evaluated 50 matched scene pairs and 50 matched object pairs from the same six scenes in the Marble dataset. Each scene pair contained renderings of the original and converted scenes from the same viewpoint, while each object pair contained renderings of the corresponding original and regenerated object. This pairing allowed us to compare scores before and after conversion at both levels.

GPT-5.5 evaluated each image individually, returning four scores and brief justifications based on visible evidence, such as missing surfaces, blurred boundaries, detached fragments, or inconsistent appearance. We aggregated these scores separately from the human ratings.

The exact prompt used for the agent evaluation was:

> You are evaluating rendered 3D reconstruction images for a paper's supplementary qualitative audit. For each provided image, assign four independent MOS-style scores from 1 to 5. Use the following scale: 5 = excellent, almost no visible distortion; 4 = good, minor distortion that does not affect the overall viewing experience; 3 = fair, visible distortion but still acceptable; 2 = poor, severe distortion and degraded viewing experience; 1 = bad, very low quality and difficult to accept.
>
> Evaluate the following four criteria independently:
>
> 1. Overall Quality: judge the first overall visual impression. Consider whether the rendering appears clear, stable, visually plausible, and whether artifacts affect the overall viewing experience. This criterion is holistic and should not be limited to one specific artifact type.
>
> 2. Surface Completeness: evaluate whether visible geometric surfaces are continuous and complete. For scene images, focus on holes, discontinuities, missing regions, or broken scene structures. For object images, focus on whether the object surface is missing, broken, collapsed, or implausibly incomplete.
>
> 3. Boundary Cleanliness: evaluate whether structural boundaries are clean and well separated. For scene images, focus on boundaries between objects and between objects and the background, including bleeding, halos, smearing, or boundary contamination. For object images, focus on the object's silhouette and outer contour, including edge blur, spillover, detached fragments, or unstable outlines.
>
> 4. Naturalness: evaluate whether the result is consistent with real visual experience. For scene images, focus on whether reconstructed or inserted regions are coherent with the surrounding scene in appearance, scale, lighting, and style. For object images, focus on whether the object's shape, material appearance, and proportions resemble a normal, naturally plausible object.
>
> Do not infer or reveal method identity from sample identifiers. Base every score only on visible image evidence. For each image, output the category, sample identifier, four numeric scores, and a concise rationale that explicitly mentions the main evidence for the assigned scores.

# D More results

To further demonstrate the robustness and generalization of our pipeline, we include additional qualitative results in the supplementary material. Figures 9 and 10 show extended visualizations on the datasets, illustrating the full pipeline progression from the input scene to segmentation, removal, inpainting, and final assembly.

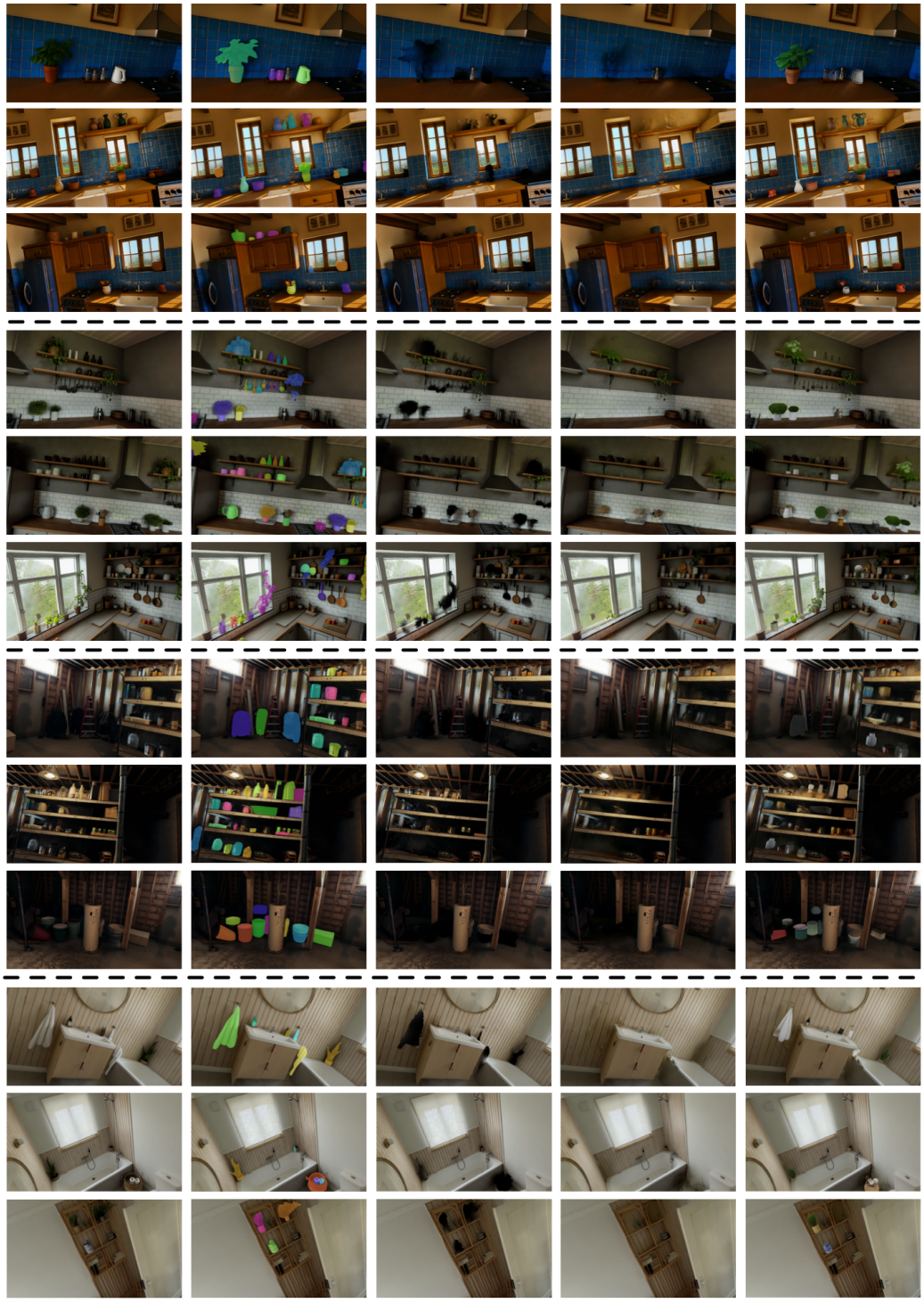


Figure 9: Additional pipeline results. For each scene, we show from left to right: the original Marble-generated 3DGS scene with detected object masks, object removal, background inpainting, and final assembly with generated objects placed back.

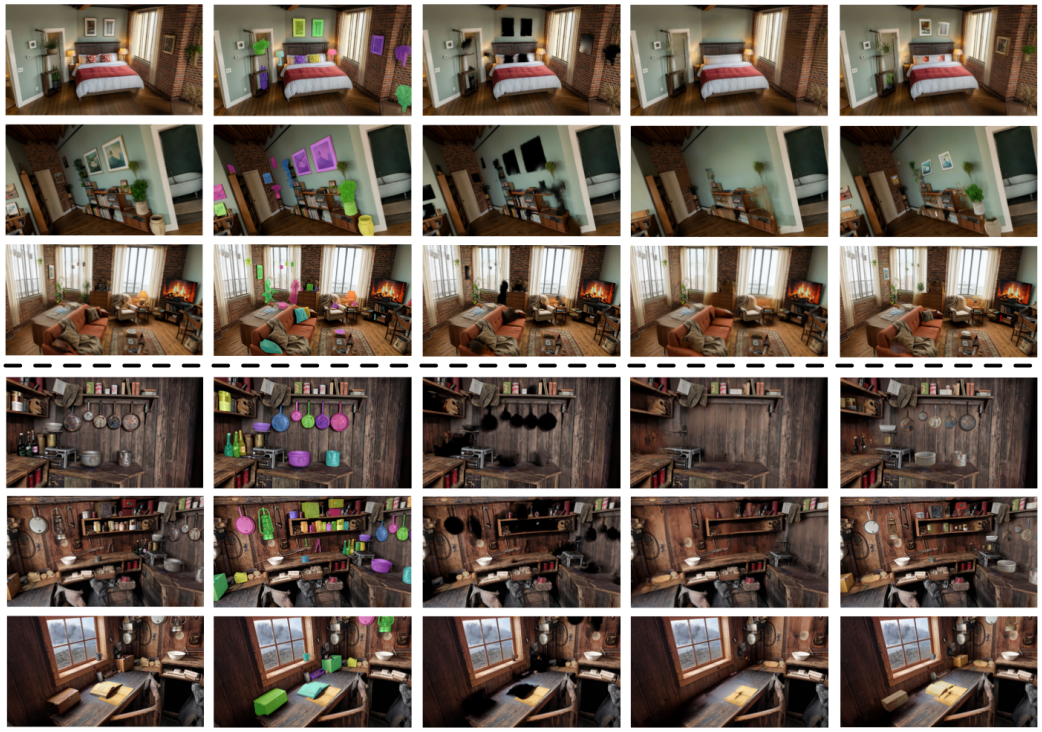


Figure 10: Additional pipeline results, which contains highly cluttered and occluded scenes. Despite the increased complexity, our method successfully decomposes objects, repairs the background, and reassembles high-quality assets.

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Abstract and introduction state the core contribution: converting monolithic 3DGS scenes into interactive, decomposable environments, supported by experiments.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in the Conclusion section.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: This paper presents a new paradigm and does not contain theoretical results.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Implementation details, hyperparameters, and evaluation protocols are provided. Code will be released upon acceptance.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code and data are provided and will be public upon acceptance, with detailed instructions provided in the paper and supplementary material.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Implementation Details section provides hardware, runtime, optimization settings, and evaluation protocols.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Results are reported as qualitative visualizations and quantitative metrics on individual scenes without repeated runs, so error bars are not applicable.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

Justification: Single NVIDIA RTX 3090 GPU and runtime are reported.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research involves no privacy violations or malicious use cases.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Positive impacts (embodied AI, content creation) and potential misuse are discussed.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release new models or datasets with high misuse risk.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All existing assets are properly cited with their original papers.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not introduce new assets for release. Dataset consists of scenes generated by Marble and HY-World.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Justification: The MOS study instructions are provided in the supplementary material.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [No]

    Justification: The study involved minimal risk (viewing rendered images). Formal IRB approval was not obtained as this was a low-stakes internal evaluation; participants were informed of the task and provided consent.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: LLMs are used as core components of our pipeline: Qwen3.6-Plus serves as the vision-language agent for object discovery and viewpoint selection, and GPT-5.5 is used for automated evaluation.