Simulation-Ready Compositional 3D Scene Reconstruction from a Single Image

Inhee Lee* **Sangwon Baik*** **Sungjoo Kim**
Hyeonwoo Kim **Hyunsoo Cha** **Hanbyul Joo†**
Seoul National University
*Equal Contribution †Corresponding Author
{ininin0516,bsw1907,masterninja,hwkim408,243stephen,hbjoo}@snu.ac.kr
https://github.com/snuvclab/SimuScene

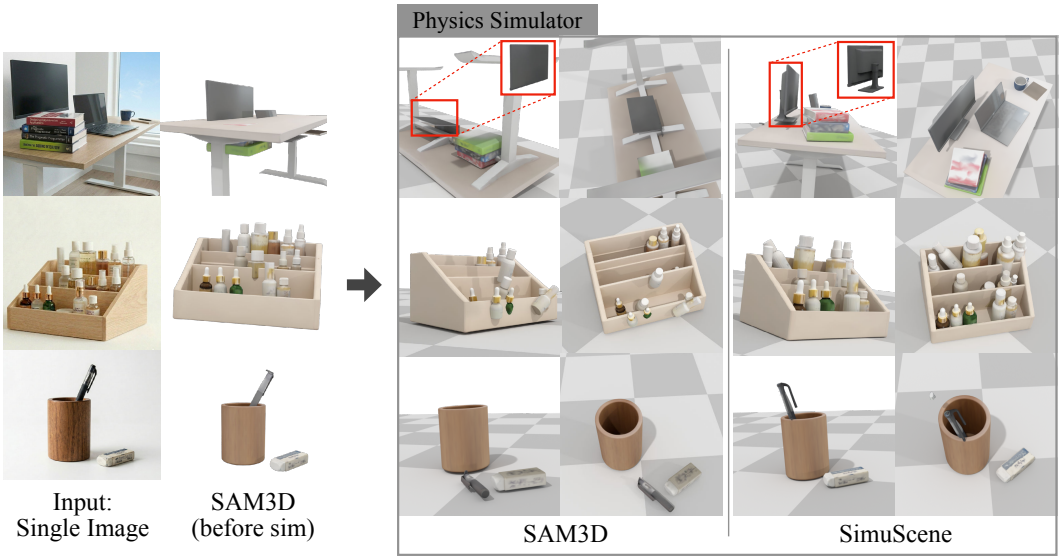


Figure 1: **SimuScene** reconstructs simulation-ready compositional 3D scenes from a single image by using physics simulation to correct object shape and pose. Compared with conventional 3D lifters such as SAM3D [10], our method handles heavy occlusion and produces stable layouts. Red boxes highlight physics-guided shape corrections.

Abstract

Reconstructing interactive, simulation-ready 3D scenes from a single image is a critical bottleneck for robotic manipulation. While recent single-image lifters recover plausible per-object shapes, composing them yields scenes that collapse under physical simulation due to interpenetrating, hovering, or sinking objects. Existing physics-aware compositional reconstruction methods mainly address this as a post-hoc layout correction, leaving the underlying geometric errors unresolved. To address this, we introduce SimuScene, a compositional 3D reconstruction pipeline that puts physics in the loop of shape and layout estimation. Rather than using physics merely for layout cleanup, we utilize the physics engine as a diagnostic measurement tool during the generative process itself. By diagnostically simulating reconstructed objects under gravity, we convert penetration and support failures into quantitative correction signals that drive amodal shape resampling, with gravity-axis stretching as a lightweight correction for minor errors. This physics-informed feedback loop mitigates accumulated reconstruction errors and

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

produces a stable, simulation-ready compositional 3D scene. Extensive experiments demonstrate state-of-the-art performance on physical stability and geometric alignment benchmarks. We further highlight SimuScene's utility by deploying reconstructed environments in humanoid control and robot-arm manipulation tasks.

1 Introduction

A single photograph of a cluttered desk contains, in principle, everything a robot needs to practice on it: the table to push against, the cup to grasp, the book to slide aside. In practice, however, current methods fall short of extracting this. Single-image scene methods either generate the entire view holistically, as novel views or a single inseparable 3D representation [44, 65, 64, 16, 63, 11], or lift each object independently and produce configurations that collapse the moment a physics simulator releases them [24, 35, 10]. Objects sink through tables, hover above shelves, or wedge into one another with overlapping geometry. The difficulty is twofold. A single image is fundamentally ambiguous about occluded geometry and absolute depth, and the physical plausibility that would resolve this ambiguity is not a property any existing dataset readily teaches.

Our goal is to reconstruct a physically plausible, simulation-ready compositional 3D scene from a single image, an output that drops directly into a physics simulator and supports downstream tasks such as reinforcement and imitation learning for robot manipulation, or a VR/AR setup as an interactive 3D scene that responds correctly to contact and gravity. Recent single-image approaches [10] make substantial progress by lifting each detected object into a complete 3D shape. While these results are visually convincing, they are often not physically plausible: once placed in a simulator, objects fall through their supports or explode out of contact, meshes interpenetrate, support relationships break, and objects sink or hover, due to errors in both occlusion-induced shape completion and monocular object pose estimation. Recent efforts [60] begin to incorporate physics into the reconstruction process, yet they apply it only at the layout level, adjusting where each object sits while leaving the underlying shapes untouched. When the geometry itself is incorrect, post-hoc layout adjustment cannot fully recover a plausible configuration. This failure mode also exposes an opportunity: the physics simulation that reveals these artifacts also measures them. For example, the distance an object falls, or the depth at which it interpenetrates a neighbor, gives an important clue to the geometric error that a single view alone cannot supply. Rather than treating these signals as artifacts to clean up at the end, we use them as diagnostic measurements that drive shape correction during reconstruction itself.

We realize this idea in SimuScene, a single-image compositional 3D scene reconstruction pipeline that puts *physics in the loop* of shape and layout estimation. In our pipeline, the physics simulation acts as a diagnostic loop that drives shape and scale correction, not a post-hoc cleanup. The gravity-direction displacement at first contact exposes whether the lifted shape is too short, too tall, or grossly mis-shaped, and feeds directly into the geometry update. We feed this simulator feedback into a resampling stage that recovers more plausible 3D shapes even under heavy occlusion. We further treat the simulator as a source of evidence for shape itself, using the simulated configuration to define the target extent of hidden geometry and thereby reduce the residual uncertainty that occlusion leaves behind. Building on these two roles of physics, SimuScene composes a scene through an iterative, sequential *physics in the loop* procedure that reconstructs one object at a time. Once loaded into a physics simulator, our reconstructions achieve state-of-the-art performance on physical-plausibility and reconstruction-quality metrics. We further demonstrate the resulting object-complete scenes driving downstream applications including humanoid control and robot-arm manipulation tasks.

We make following contributions: (1) **Physics-in-the-loop diagnostic simulation.** A sequential, per-object protocol integrates physical dynamics directly into reconstruction, converting violations (e.g., interpenetration, gravity-induced displacement) into actionable diagnostic signals. (2) **Physics-informed shape correction.** We use simulated displacement to select between two geometry updates. OBB-guided amodal resampling improves shape quality for large displacements, while gravity-axis stretching reduces computation for small ones. (3) **Extensive experimentation.** We provide comprehensive evaluations across diverse datasets and metrics, and show that our reconstructed simulation-ready 3D scenes support downstream robotics applications (i.e. humanoid control policy learning, robot-arm manipulation)

2 Related Work

Multi-Object 3D Reconstruction from Cluttered Images. Recovering a *structured* 3D scene from a single RGB image is ill-posed under occlusion, depth ambiguity, and inter-object interactions, originally tackled by analysis-by-synthesis search over CAD exemplars under structural and physical constraints [22]. End-to-end learning now produces coherent multi-object geometry from a single view via feed-forward regression with non-intersecting outputs [41], multi-instance diffusion with cross-instance attention [24], hierarchical isometric-view amodal completion [13], or promptable scene-level reconstruction [10], while divide-and-conquer pipelines reconstruct each object independently and reassemble them via iterative occlusion removal [2], differentiable optimal-transport alignment [18], or feed-forward decoupling of appearance, rotation, scale, and translation [21]. 3D-Fixer [62] further addresses occluded geometry through coarse-to-fine in-place completion, using observed partial geometry as a spatial anchor without explicit pose alignment. Other related systems decompose scenes from multi-view captures [38, 59], retrieve similar assets instead of reconstructing input-faithful geometry [12], or rebuild scenes as graphics programs through agentic multimodal reasoning [61]. These systems differ from our single-image setting in input modality or reconstruction target. Despite improving geometric fidelity and instance separation, the resulting assets routinely interpenetrate, float, or fail to satisfy support relations, leaving the scenes unsuitable for downstream simulation and embodied use.

Physics-Aware Scene Reconstruction and Simulation-in-the-Loop Refinement. Physics constraints have long shaped reconstruction, from early scene parsers that encode support and collision feasibility in the inference objective [22] to recent physics-aware and differentiable-simulation pipelines that jointly optimize geometry, pose, material, or appearance via physically plausible implicit surfaces [37], MPM-driven compositional Gaussian generation [56], amodal MPM-coupled scene reconstruction [9], or human-scene depth-alignment and contact priors [55], with simulator feedback also serving as a reward-style alignment signal for image-to-3D generators [28]. At the scene level, relation-graph rigid-body solvers [60] and simulator-in-the-loop assembly [54] resolve penetration and floating artifacts in assembled layouts, while text-conditioned generators embed SDF collision avoidance, gravity, or differentiable rigid-body simulation directly in the generation loop to produce intersection-free, statically stable scenes [31, 27, 30].

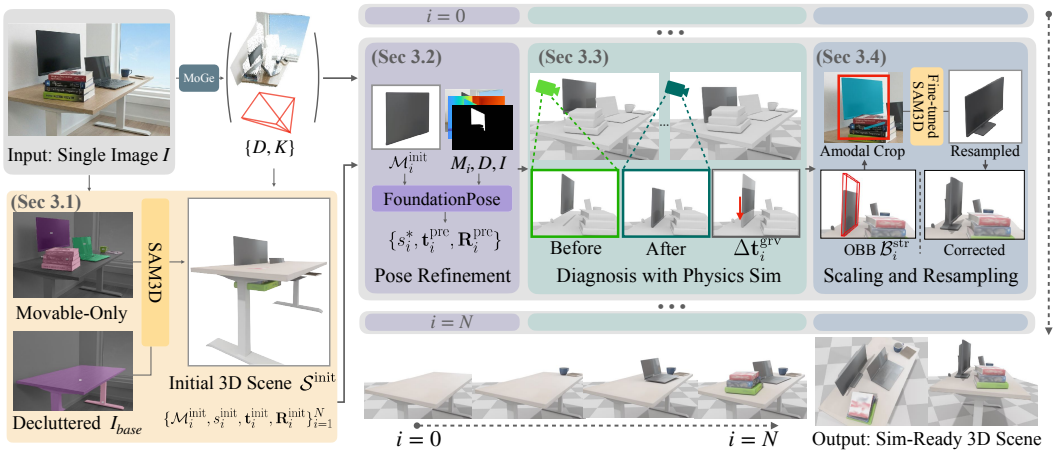
Direct Preference Optimization for Generative Models. Direct Preference Optimization (DPO) replaces the reward modeling and on-policy RL step of RLHF [39] with a simple preference-classification objective derived from a closed-form reparameterization of the Bradley–Terry reward model [43]. Subsequent language-model alignment methods generalize this objective [4], replace pairwise preferences with binary utility feedback [14], integrate preference optimization into supervised fine-tuning [19], or remove the reference model through a length-normalized reward with a target margin [34]. For visual generation, Diffusion-DPO [50] adapts DPO to diffusion models by replacing autoregressive log-likelihood ratios with denoising-loss differences, while related methods formulate denoising as a policy optimization problem [58, 7] or introduce step-wise preference comparisons [29]. At the 3D level, DSO [28] uses rigid-body simulation feedback to fine-tune image-to-3D generators for physical soundness with DPO or DRO objectives. Inspired by DSO and Diffusion-DPO, we fine-tune the SAM3D shape branch with a flow-matching DPO objective over synthetic occlusion-versus-amodal-completion pairs. Unlike DSO, which optimizes standalone object stability, our preference signal targets occlusion-robust shape resampling for objects in cluttered scenes.

3 Method

Given a single RGB image $\boldsymbol{I}$, our goal is to reconstruct a simulation-ready compositional 3D scene $\boldsymbol{\mathcal{S}}$, visually aligned with $\boldsymbol{I}$ and physically consistent under gravity:

$$\boldsymbol{\mathcal{S}} = \{\boldsymbol{\mathcal{M}}_i, s_i, \mathbf{t}_i, \mathbf{R}_i\}_{i=1}^{N}, \tag{1}$$

where $\boldsymbol{\mathcal{M}}_i$ is a 3D canonical mesh for i-th object, $s_i \in \mathbb{R}^+$ is an isotropic scale of $\boldsymbol{\mathcal{M}}_i$, $\mathbf{t}_i \in \mathbb{R}^3$ and $\mathbf{R}_i \in \mathrm{SO}(3)$ are a translation vector and a rotation matrix of $\boldsymbol{\mathcal{M}}_i$, and N is the number of objects in the scene. Monocular ambiguity and occlusion often cause penetration, floating, and toppling. We therefore use diagnostic physics signals to refine object poses and correct geometry through Oriented Bounding Box (OBB)-guided stretching or resampling. Fig. 2 summarizes our pipeline.



(a) MLLM-based initial object lifting (b) Sequential Physics-informed Pose-Shape Refinement

Figure 2: **Pipeline Overview.** (a) From a single image I, foundation priors decompose the scene into object meshes with poses. (b) Each object then passes through pose initialization, diagnostic physics simulation, and shape correction. Completed objects are frozen as colliders, and physics displacements drive the correction. The corrected objects settle into a simulation-ready 3D scene.

3.1 Decomposed Scene Initialization

Because base structures (e.g., tables, shelves) support most objects in cluttered scenes, we reconstruct them separately from other objects. We first remove non-base objects with an MLLM-based image editor [48] and run SAM3D [10] on the decluttered image $\boldsymbol{I}_{\text{base}}$ to obtain fixed base colliders. For the remaining objects, RAM++ [23] and a VLM [1] generate instance labels, SAM3 [8] produces masks $\boldsymbol{M}_i$, and SAM3D, denoted by Φ, lifts each instance:

$$\Phi(\boldsymbol{I}, \boldsymbol{M}_i, \boldsymbol{I}_{\boldsymbol{M}_i}, \boldsymbol{D}) = (\boldsymbol{\mathcal{M}}_i^{\text{init}}, s_i^{\text{init}}, \mathbf{t}_i^{\text{init}}, \mathbf{R}_i^{\text{init}}), \tag{2}$$

where $\boldsymbol{M}_i$ is the segmentation mask of i-th object, $\boldsymbol{I}_{\boldsymbol{M}_i}$ is the masked image crop, and $\boldsymbol{D}$ is single-view depth from MoGe [51]. $\boldsymbol{\mathcal{M}}_i^{\text{init}}$ is in SAM3D's canonical frame, with its AABB centered at the origin and the longest side normalized to 1. Beyond geometry, we extract semantic information that proves critical for physics. The VLM tags each object with one of three pose-DoF labels, *free* (6-DoF), *point-anchored* (3-DoF rotation about a wall point), or *line-anchored* (1-DoF rotation about the anchor line). These tags tell the simulator which constraints to apply to each object. Since the floor and walls are fixed after initial estimation, we omit them from Eq. (1) for simplicity. See Sec. A.1 for additional details on scene decomposition.

3.2 Pose Refinement Before Simulation

SAM3D produces plausible shapes but in its own canonical frame, with poses only approximately consistent with the image. Handing these poses directly to a physics simulator amplifies even small rotational errors into severe penetration and contact artifacts. We therefore refine each object's pose against image evidence before simulation. We obtain depth $\boldsymbol{D}$ and camera intrinsics $\boldsymbol{K}$ from $\boldsymbol{I}$ using MoGe [51], and back-project the pixels inside the segmentation mask $\boldsymbol{M}_i$ to form the object point cloud $\boldsymbol{\mathcal{P}}_i \in \mathbb{R}^{P\times 3}$. We first solve for $(s_i^*, \mathbf{t}_i^*)$ by aligning the canonical mesh $\boldsymbol{\mathcal{M}}_i^{\text{init}}$ with $\boldsymbol{\mathcal{P}}_i$ (see Sec. A.1 for details). Rotation, however, cannot be reliably recovered from point-cloud alignment alone, so we hand the result to FoundationPose [52] for refinement:

$$(\mathbf{t}_i^{\text{pre}}, \mathbf{R}_i^{\text{pre}}) = \text{FoundationPose}\big(\boldsymbol{I}, \boldsymbol{D}, \boldsymbol{K}, \boldsymbol{M}_i, s_i^* \boldsymbol{\mathcal{M}}_i^{\text{init}}, \mathbf{t}_i^*, \mathbf{R}_i^{\text{init}}\big), \tag{3}$$

where $s_i^* \boldsymbol{\mathcal{M}}_i^{\text{init}}$ denotes $\boldsymbol{\mathcal{M}}_i^{\text{init}}$ scaled by s_i^*, and $\mathbf{t}_i^*$ and $\mathbf{R}_i^{\text{init}}$ from the previous alignment are used for the initials. The pre-simulation scene is then composed as:

$$\boldsymbol{\mathcal{S}}^{\text{pre}} = \{\boldsymbol{\mathcal{M}}_i^{\text{init}}, s_i^*, \mathbf{t}_i^{\text{pre}}, \mathbf{R}_i^{\text{pre}}\}_{i=1}^N. \tag{4}$$

3.3 Diagnostic Simulation: Physics as a Probe

The pre-simulation scene $\boldsymbol{\mathcal{S}}^{\text{pre}}$ is the image-faithful configuration we can produce from perception alone, but it is not yet physically valid. It still contains penetrations and floating objects rooted in monocular reconstruction ambiguity, and as shown in Fig. 4, naively dropping everything under gravity produces chaotic dynamics in which mutually penetrating objects push each other into impossible configurations. We use physics simulation to diagnose these residual errors: we first resolve inter-object penetration by displacing the target object along a simulator-informed direction. Then, after releasing the object under gravity, the resulting displacement and contact signals expose unresolved gravity-axis shape and support errors that drive the shape-correction stage (Sec. 3.4).

To keep the diagnostic signal clean and avoid joint-settling deadlocks, we compose the scene with a sequential protocol. Objects are processed in ascending order of the lowest corner of their world-frame bounding box along the gravity direction. For each active object i, only that object is dynamic, while previously placed objects are frozen as static colliders, and the simulation halts at the first frame of contact. This prevents mutually penetrating objects from pushing each other into impossible configurations and avoids unstable trajectories that drift far from the image-aligned pose.

Penetration Resolution. For simplicity, we assume that objects are indexed according to the sequential processing order, so that object i is processed after objects $1, \ldots, i-1$. For each active object i, the simulator generates a penetration-resolving displacement $\Delta\mathbf{t}_i^{\text{pen}}$ to prevent overlap with the previously processed objects $1, \ldots, i-1$. For the first object, the procedure resolves penetration with the floor and any walls. The scene after penetration resolution, denoted by $\boldsymbol{\mathcal{S}}^{\text{pen}}$, is formulated as

$$\boldsymbol{\mathcal{S}}^{\text{pen}} = \{\boldsymbol{\mathcal{M}}_i^{\text{init}}, s_i^*, \mathbf{t}_i^{\text{pre}} + \Delta\mathbf{t}_i^{\text{pen}}, \mathbf{R}_i^{\text{pre}}\}_{i=1}^{N}. \tag{5}$$

Objects whose pose-DoF tag is *point-anchored* or *line-anchored* are anchored at their current positions using the procedure described in Sec. A.3. Accordingly, as discussed in Sec. 3.1, objects tagged as *point-anchored* are restricted to 3-DoF rotation about the anchor point, while objects tagged as *line-anchored* are restricted to 1-DoF rotation about the anchor line.

Gravity-Based Diagnostics. We then sequentially release each active object i under gravity. If the pose-DoF tag of the object is *free*, the object is frozen at the first simulation frame in which it contacts any surface, yielding the gravity-induced displacement $\Delta\mathbf{t}_i^{\text{grv}}$. For wall-anchored objects, i.e., those tagged as *point-anchored* or *line-anchored*, the object is simulated under its corresponding rotational constraint until it settles under gravity, yielding the gravity-induced rotation $\Delta\mathbf{R}_i^{\text{grv}}$. The resulting post-simulation scene $\boldsymbol{\mathcal{S}}^{\text{sim}}$ is formulated as

$$\boldsymbol{\mathcal{S}}^{\text{sim}} = \{\boldsymbol{\mathcal{M}}_i^{\text{init}}, s_i^*, \mathbf{t}_i^{\text{pre}} + \Delta\mathbf{t}_i^{\text{pen}} + \Delta\mathbf{t}_i^{\text{grv}}, \Delta\mathbf{R}_i^{\text{grv}}\mathbf{R}_i^{\text{pre}}\}_{i=1}^{N}. \tag{6}$$

Here, $\Delta\mathbf{R}_i^{\text{grv}}$ is the identity matrix for objects tagged as *free*. For wall-anchored objects, the constrained simulation returns the full pose change $(\Delta\mathbf{t}_i^{\text{grv}}, \Delta\mathbf{R}_i^{\text{grv}})$ induced by rotation about the anchor.

Displacement-Based Shape Correction Criterion. We use the physics-induced translation as a diagnostic signal for shape correction. First, we compute the object's OBB (Oriented Bounding Box) in the pre-simulation scene $\boldsymbol{\mathcal{S}}^{\text{pre}}$:

$$\mathcal{B}_i^{\text{pre}} = \text{OBB}(\mathbf{R}_i^{\text{pre}} \left(s_i^* \boldsymbol{\mathcal{M}}_i^{\text{init}}\right) + \mathbf{t}_i^{\text{pre}}). \tag{7}$$

Let ℓ_i^{pre} be the side length of $\mathcal{B}_i^{\text{pre}}$ along the OBB axis most aligned with the unit gravity direction $\hat{\mathbf{g}}$. We then define the normalized gravity-axis displacement as

$$\rho_i = \frac{\left|\hat{\mathbf{g}}^\top \left(\Delta\mathbf{t}_i^{\text{pen}} + \Delta\mathbf{t}_i^{\text{grv}}\right)\right|}{\ell_i^{\text{pre}}}. \tag{8}$$

ρ_i measures how much each object in the post-simulation scene $\boldsymbol{\mathcal{S}}^{\text{sim}}$ deviates from its counterpart in the pre-simulation scene $\boldsymbol{\mathcal{S}}^{\text{pre}}$, which is the scene most closely aligned with the input image. When ρ_i is small, we regard the deviation as the result of minor errors accumulated through the pipeline and correct the scene with a lightweight adjustment. In contrast, when ρ_i is large, we attribute the deviation to a more fundamental shape-sampling failure, since our pipeline already performs substantial pose refinement. Specifically, if $\rho_i \geq 0.15$, we attribute the error to an unreliable SAM3D shape sample and trigger shape resampling. Otherwise, we correct the object by stretching it along the gravity-aligned OBB axis.

3.4 Physics-Informed Shape Correction

Gravity-axis Stretch. When $\rho_i < 0.15$, we regard the discrepancy between $\boldsymbol{\mathcal{S}}^{\text{pre}}$ and $\boldsymbol{\mathcal{S}}^{\text{sim}}$ as a small accumulated error and correct it with a lightweight stretch rather than resampling, since resampling brings no measurable shape benefit in this regime (Tab. 4). Let $\hat{\mathbf{u}}_i^{\text{grv}}$ be the OBB axis of $\mathcal{B}_i^{\text{pre}}$ most aligned with the gravity direction $\hat{\mathbf{g}}$, and let $\hat{\mathbf{a}}_i^{\text{grv}} = (\mathbf{R}_i^{\text{pre}})^\top \hat{\mathbf{u}}_i^{\text{grv}}$ be the corresponding axis in the canonical mesh frame. We use the signed gravity-axis displacement

$$\eta_i = \frac{\hat{\mathbf{g}}^\top (\Delta \mathbf{t}_i^{\text{pen}} + \Delta \mathbf{t}_i^{\text{grv}})}{\ell_i^{\text{pre}}} \tag{9}$$

to stretch the canonical mesh along $\hat{\mathbf{a}}_i^{\text{grv}}$. The stretched mesh is then renormalized to preserve the canonical-mesh convention, with the normalization absorbed into the object scale. The translation is updated so that the OBB face opposite to gravity remains fixed. This yields the stretched object $(\boldsymbol{\mathcal{M}}_i^{\text{str}}, s_i^{\text{str}}, \mathbf{t}_i^{\text{str}}, \Delta\mathbf{R}_i^{\text{grv}}\mathbf{R}_i^{\text{pre}})$. See Sec. A.4 for details.

Amodal Shape Resampling. When $\rho_i \geq 0.15$, we attribute the error to an unreliable SAM3D shape sample, which typically occurs when severe occlusion causes SAM3D to fail at amodal shape completion. We compute a desired OBB $\mathcal{B}_i^{\text{str}}$ by stretching $\mathcal{B}_i^{\text{pre}}$ along the gravity axis by the simulated displacement, and use it to guide shape resampling. Since the true amodal mask is unavailable, we construct an auxiliary crop mask $\boldsymbol{M}_i'$ by augmenting the modal segmentation mask $\boldsymbol{M}_i$ with the projected lowest part of $\mathcal{B}_i^{\text{str}}$, as shown in Fig. 2. Although $\boldsymbol{M}_i'$ is not a true amodal segmentation mask, it yields an amodal crop image $\boldsymbol{I}_{\boldsymbol{M}_i'}$ that covers the expected hidden extent of the target object. We then run SAM3D with this physics-informed crop:

$$\Phi(\boldsymbol{I}, \boldsymbol{M}_i, \boldsymbol{I}_{\boldsymbol{M}_i'}, \boldsymbol{D}) = (\boldsymbol{\mathcal{M}}_i^{\text{re}}, s_i^{\text{re}}, \tilde{\mathbf{t}}_i^{\text{re}}, \mathbf{R}_i^{\text{re}}). \tag{10}$$

Since the resampled shape has different canonical geometry and scale, the previous scale and rotation are no longer valid, while the desired OBB $\mathcal{B}_i^{\text{str}}$ provides the target object location. We therefore keep the resampled mesh, scale, and rotation from SAM3D, but set the translation to the OBB center, $\mathbf{t}_i^{\text{re}} := \text{center}(\mathcal{B}_i^{\text{str}})$. The resulting resampled object is represented as $(\boldsymbol{\mathcal{M}}_i^{\text{re}}, s_i^{\text{re}}, \mathbf{t}_i^{\text{re}}, \mathbf{R}_i^{\text{re}})$.

SAM3D Fine-tuning for Amodal Resampling. To make SAM3D robust to the resampling input in Eq. (10), we fine-tune it with synthetic preference pairs that contrast occlusion-failed shape latents against edited full-object completion latents.

We use a DPO-style objective [28, 43, 50] that compares the flow-matching [32] losses of preferred and rejected shape latents:

$$\mathcal{L}_{\text{FM-DPO}}(\theta) = \mathbb{E}_i \left[w_i \left(-\log \sigma \left(-\frac{\beta}{2} \left(d_{\theta,i} - d_{\text{ref},i} \right) \right) \right) \right], \tag{11}$$

where w_i is the sample weight, σ is the sigmoid function, β controls the preference strength, $d_{\theta,i}$ is the win–lose flow-matching loss difference of the current model, and $d_{\text{ref},i}$ is the corresponding difference computed by the frozen base SAM3D reference model. This objective encourages the fine-tuned model to assign a lower relative flow-matching loss to the completed amodal latent than to the occlusion-failed latent. To also directly supervise the completed amodal latent, we add a supervised fine-tuning (SFT) term:

$$\mathcal{L}_{\text{total}}(\theta) = \mathcal{L}_{\text{FM-DPO}}(\theta) + \lambda_{\text{SFT}} \mathcal{L}_{\text{SFT}}(\theta). \tag{12}$$

Instead of updating the full SAM3D generator, we train only LoRA [20] adapters inserted into the attention projections of the shape generation branch. The LoRA adapters are trained once offline and kept fixed for all test scenes, so amodal resampling runs feed-forward without gradient-based optimization or per-scene fine-tuning. Details of preference-pair construction, loss definitions, and training weights are provided in Sec. A.4.

3.5 Final Composition

The preceding stages produce corrected object states, but they do not yet define the final simulation-ready layout: the diagnostic simulation in Sec. 3.3 is designed to extract clean correction signals, whereas the final scene must be physically settled under the object's full pose constraints. We first assemble each object from its corrected state. Wall-anchored objects

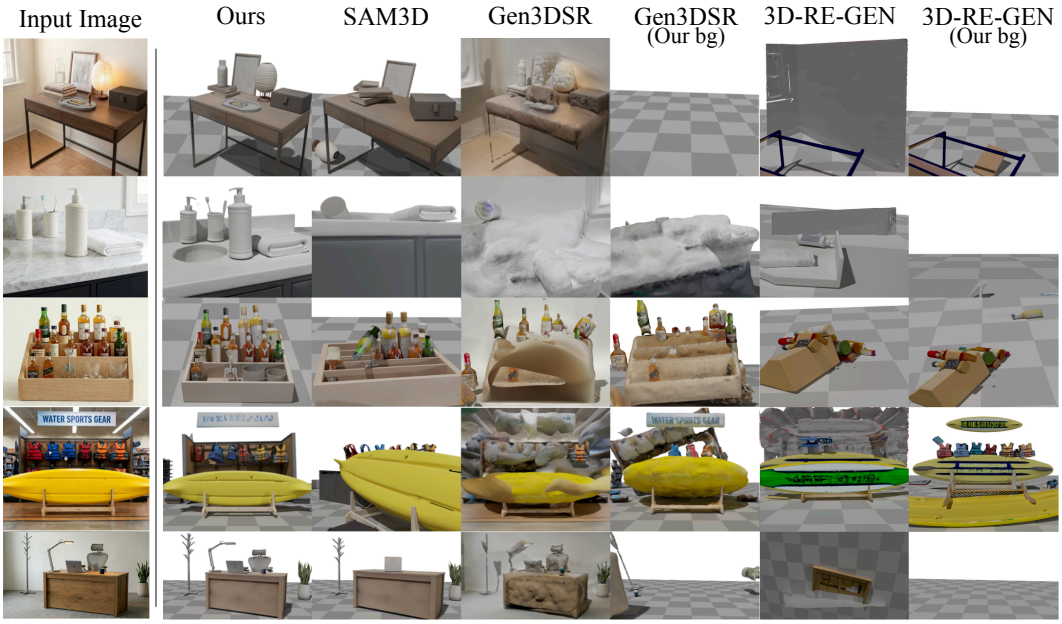


Figure 3: **Qualitative Comparison of GenWild.** All scenes are visualized after gravity-driven physics simulation. Baselines either remain artificially stuck mid-air due to interpenetration or collapse catastrophically, while our method preserves both physical plausibility and input-view alignment in cluttered scenes.

	GraspClutter6D					AriaDigitalTwin					GenWild				
Method	$\text{ABO}_{\text{fh/fo}}\uparrow$	ABO $\uparrow$	$D_{\text{mean}}\downarrow$	$E_{\text{pk}}\downarrow$	Penetr. $\downarrow$	3D $\text{ABO}_{\text{fh/fo}}\uparrow$	3D ABO $\uparrow$	$D_{\text{mean}}\downarrow$	$E_{\text{pk}}\downarrow$	Penetr. $\downarrow$	$\text{ABO}_{\text{fh/fo}}\uparrow$	ABO $\uparrow$	$D_{\text{mean}}\downarrow$	$E_{\text{pk}}\downarrow$	Penetr. $\downarrow$
SAM3D [10]	0.087	0.121	0.638	4.064	24.5%	0.037	0.052	0.721	3.556	16.4%	0.218	0.263	0.555	2.957	16.2%
Gen3DSR [3] (our bg)	0.014	0.036	0.880	5.644	50.3%	0.070	0.095	0.932	5.769	17.2%	0.104	0.144	0.817	5.230	16.9%
Gen3DSR [3]	0.014	0.219	0.267	1.587	75.8%	0.002	0.226	**0.091**	**0.556**	93.3%	0.046	0.477	0.255	1.214	72.7%
3D-RE-GEN [45] (our bg)	0.042	0.071	0.659	4.086	24.9%	0.044	0.060	0.835	4.261	17.2%	0.110	0.154	0.612	3.491	19.7%
3D-RE-GEN [45]	0.056	0.100	0.524	2.790	31.4%	0.025	0.053	0.797	4.162	32.8%	0.104	0.164	0.608	2.790	27.0%
3D-Fixer [62]	0.106	0.122	0.606	3.455	**10.0%**	0.116	0.159	0.720	3.884	20.2%	0.191	0.258	0.568	2.680	20.1%
Ours	**0.248**	**0.267**	**0.250**	**1.092**	10.3%	**0.247**	**0.285**	0.310	1.370	**8.8%**	**0.507**	**0.532**	**0.128**	**0.420**	**2.5%**

Table 1: **Main Quantitative Comparison.** "(our bg)" places each baseline's reconstructed objects in our reconstructed background (floor and walls). $\text{ABO}_{\text{fh/fo}}$ averages the post-simulation ABO over non-penetrating free+hanging (wall-mounted) objects and over free-only objects, measuring the alignment of physically valid objects. Ours generally performs best and leads in $\text{ABO}_{\text{fh/fo}}$ on all three datasets.

retain the initial geometry and use the constrained pose obtained from diagnostic simulation: $(\boldsymbol{\mathcal{M}}_i^{\text{init}}, s_i^*, \mathbf{t}_i^{\text{pre}} + \Delta\mathbf{t}_i^{\text{pen}} + \Delta\mathbf{t}_i^{\text{grv}}, \Delta\mathbf{R}_i^{\text{grv}}\mathbf{R}_i^{\text{pre}})$. Free objects are replaced by either the stretched state $(\boldsymbol{\mathcal{M}}_i^{\text{str}}, s_i^{\text{str}}, \mathbf{t}_i^{\text{str}}, \Delta\mathbf{R}_i^{\text{grv}}\mathbf{R}_i^{\text{pre}})$ or the resampled state $(\boldsymbol{\mathcal{M}}_i^{\text{re}}, s_i^{\text{re}}, \mathbf{t}_i^{\text{re}}, \mathbf{R}_i^{\text{re}})$, according to the correction branch selected in Sec. 3.3. Starting from these assembled states, we rerun sequential penetration resolution to remove residual overlaps introduced by shape correction. Unlike the diagnostic probe, the final simulation does not freeze free objects at first contact. Instead, free objects are simulated as full 6-DoF rigid bodies, while *point-anchored* and *line-anchored* objects follow their corresponding rotational constraints. We let the scene settle under gravity for a fixed duration and take the resulting configuration as the final simulation-ready compositional 3D scene $\boldsymbol{\mathcal{S}}$.

4 Experiments

4.1 Experimental Setup

Implementation Details. For physical simulation and evaluation, we employ MuJoCo [49] and extract collision geometries for each reconstructed object using V-HACD. For SAM3D fine-tuning, we use 13,926 latent preference pairs, with 6,866 occlusion-completion pairs and 7,060 clean preservation pairs. The pair weights w_i in Eq. (11) are set to 1.0 and 0.1, respectively, and we use $\beta = 500$ and $\lambda_{\text{SFT}} = 300$. We freeze the pretrained SAM3D model as θ_{ref} and fine-tune only shape-branch LoRA [20] adapters in the Stage-1 MM-DiT backbone, applying the flow-matching loss only to the

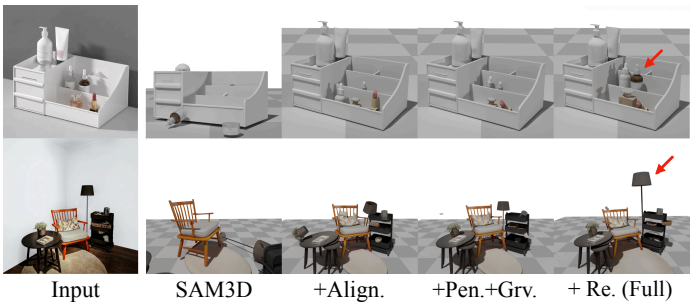


Figure 4: **Qualitative Ablation.** Post-simulation rendering of 3D scene. From SAM3D, we progressively add per-object alignment (Align.), penetration resolution (Pen.), stabilized gravity simulation (Grv.), and shape resampling (Re.). The red arrows indicate that resampling successfully generates plausible, aligned shapes.

Variants	ABO ↑	$\text{ABO}_{\text{fh/fo}}$ ↑	D_{mean} ↓
SAM3D	0.263	0.218	0.555
+Align.	0.409	0.361	0.364
+Pen.	0.403	0.372	0.395
+Grv.	0.440	0.406	0.197
+Re. (Full)	**0.532**	**0.507**	**0.128**

Table 2: **Quantitative Ablation.** Each stage follows Fig. 4.

Metric	Ours	SAM3D	Stretched
Elo ↑	**1083.4**	931.5	908.9
Win Rate (%) ↑	**71.9**	41.3	36.8

Table 3: **Mesh Completion Quality.** VLM-based evaluation [53] on GenWild. Win Rate is the average pairwise win rate against the other candidates.

Stage-1 shape latent. The adapters use rank $r = 64$, scaling $\alpha = 128$, no dropout, and are applied to 120 linear layers across 24 transformer blocks. We train with AdamW [33] using a global batch size of 16 on a single NVIDIA RTX PRO 6000 Blackwell GPU with four gradient accumulation steps, a learning rate of 1×10^{-5} after 300 warmup steps, weight decay 0.01, gradient clipping at 1.0, and bfloat16 mixed precision for 5,000 steps.

Datasets. We evaluate multi-object composition and physical stability on three complementary datasets.

- **GraspClutter6D** [5] provides 94 real cluttered scenes in its YCB-V test split. For each scene, we pick the view with the largest margin between visible objects and the image boundary to avoid truncation, use the provided visible masks as ground truth, and segment unannotated supports such as tables, boxes, and shelves with SAM3.
- **Aria Digital Twin (ADT)** [40] supplies per-instance 3D ground-truth meshes and 6-DoF poses. We sample 40 static scenes without humans or moving objects (Sec. B.3) to evaluate reconstruction, pose, and full-scene stability.
- **GenWild** contains 50 synthetic images of challenging, out-of-distribution everyday layouts generated from text and image prompts [48].

Evaluation Metrics. Visual fidelity alone is insufficient if scenes collapse under physics, while trivial stability (e.g., naively projecting objects onto the floor) destroys the original spatial intent. We therefore compute all metrics *post-simulation*, after objects settle under gravity for 5 s in MuJoCo [49]. Measuring alignment after settling penalizes both physical collapse and distorted layouts, making it more relevant to downstream simulation than alignment before settling.

- **Residual motion** (D_{mean}, E_{pk}). Mean Displacement is the average translation of each object between its reconstructed and settled poses, and Peak Energy is the peak kinetic energy of each object during simulation, averaged over objects. A simulation-ready scene should already be at rest, so residual motion indicates an unstable layout. Peak Energy further captures violent transients, such as overlapping bodies pushed apart at high speed, which net displacement does not reflect.
- **Penetration** (Penetr.). The Penetration Ratio is the fraction of objects with any post-simulation contact deeper than 2 mm. A penetrating object is not directly interactable, since collisions between overlapping bodies produce unnatural behavior.
- **View alignment** (ABO, $\text{ABO}_{\text{fh/fo}}$). Average Best Overlap measures fidelity to the input view after settling by averaging each ground-truth object's highest IoU with any prediction, using 3D bounding boxes for ADT and 2D masks otherwise. A high standard ABO alone does not establish simulation usability, because it also credits penetrating objects that are stuck and unusable in simulation. We thus introduce ABO_{fo} and ABO_{fh}, which assign zero to penetrating objects. ABO_{fo} scores only free objects, which are fully interactable, and ABO_{fh} adds hanging (wall-mounted) objects, which are partially interactable since they can only rotate about their anchors. We use their average $\text{ABO}_{\text{fh/fo}}$, which weights fully interactable objects more heavily since free objects appear in both terms, while penalizing stuck, non-interactable objects.

Detailed formulations are given in Sec. B.1, and appearance metrics (PSNR, LPIPS, and CLIP distance) are reported in Sec. C.6, where our method also performs best.

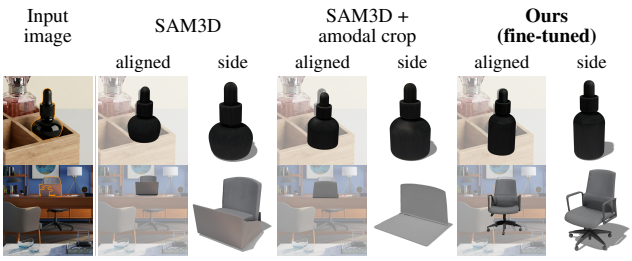


Figure 5: **Effect of SAM3D Fine-tuning.** Under heavy occlusion, our resampling uses an amodal crop covering the hidden extent predicted by the gravity-stretched OBB. Only our fine-tuned model turns this crop into complete shapes aligned with the input view.

τ	#Resampled	ADD-S$_{\text{norm}}$ ↓	Time (s)
0	708	0.0334	318.9
0.15	482	**0.0333**	217.1
∞	0	0.0603	0.0

Table 4: **Resampling Threshold** on ADT. τ=0 resamples every object and τ=∞ only stretches. Our default τ=0.15 is the most economical setting, matching the shape error of τ=0 while cutting resampling calls and time by 32%.

Baselines. We compare against methods capable of input-view aligned compositional reconstruction, namely SAM3D [10], Gen3DSR [3], 3D-RE-GEN [45], and 3D-Fixer [62]. All methods are evaluated with identical MuJoCo parameters and the same kinematic DoF tags to prevent unfair simulation penalties for hanging objects. Because flawed baseline backgrounds (e.g., missing floors) often trigger immediate simulation collapse, we evaluate Gen3DSR and 3D-RE-GEN twice, once with their native backgrounds and once with our wall/floor base mesh replacing those backgrounds (*our bg*), to isolate object reconstruction quality.

4.2 Experimental Results

Quantitative Comparison. As shown in Tab. 1, our method achieves state-of-the-art results across all datasets, yielding the highest ABO$_{\text{fh/fo}}$ on every dataset. While Gen3DSR [3] records a deceptively high standard ABO, its severe penetration reveals physically fused objects, causing its ABO$_{\text{fh/fo}}$ to drop sharply. Furthermore, since baselines often lack physically accurate environments, we evaluate them by supplying our extracted boundary constraints (*our bg*). Interacting with these proper solid boundaries fully exposes their inherent instability. For 3D-RE-GEN, we replace its background mesh with our wall boundary while retaining its native floor parameters. Despite this setup, its objects still fail to settle stably and maintain high energy errors.

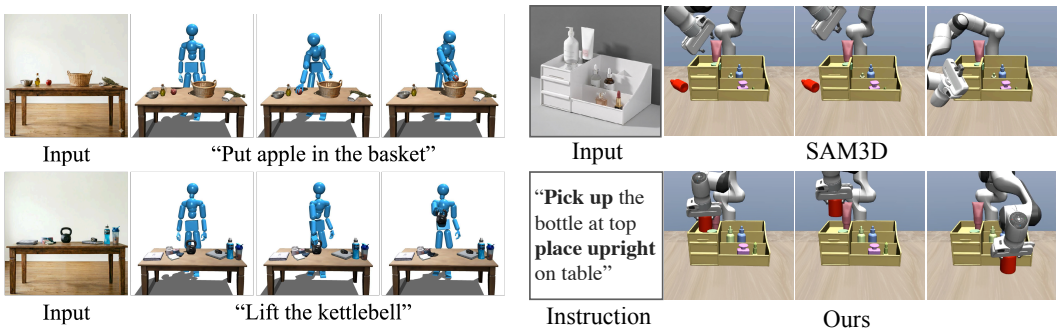
Qualitative Results. Visual comparisons in Fig. 3 confirm these quantitative trends. As baselines frequently fuse objects together or embed them into the background, their high penetration ratios cause severe collisions during simulation, scattering objects out of bounds and resulting in completely missing meshes in their visualizations. In contrast, our pipeline places reconstructed objects in precise, static alignment with the input image, faithfully preserving the original spatial layout. Even heavily occluded items are accurately anchored to their exact locations with complete 3D geometry.

4.3 Ablation Study

We ablate our pipeline on GenWild by progressively adding components (Tab. 2, Fig. 4). Pose alignment (**+Align.**) yields the largest gain by improving visual alignment and resolving deep interpenetrations. Rigid physics (**+Pen.+Grv.**) then stabilizes dynamics, but incorrect geometries persist. Physics-informed shape correction (**+Re.**) resolves these distortions, and amodal resampling replaces flawed meshes, yielding the lowest displacement error and highest alignment scores.

Beyond layout stability, we test whether our fine-tuned amodal resampler completes occluded geometry. For every object that triggers resampling, a VLM-based pairwise comparison [53] judges which rendered mesh is the more plausible completion given the input. Our resampler achieves a substantially higher Elo score and win rate than raw SAM3D and stretching (Tab. 3, Fig. 5).

Correction Threshold. We vary the resampling threshold τ (Sec. 3.3) on ADT and measure shape error with ADD-S$_{\text{norm}}$, the ADD-S error after rigid ICP alignment to the GT mesh, normalized by the GT object diameter (Tab. 4, Sec. C.2). For ρ_i below our default $\tau = 0.15$, stretching and resampling yield nearly identical shape error, while resampling substantially reduces the error for larger ρ_i. Resampling every object ($\tau = 0$) thus brings no measurable shape gain over our two-tier rule but



(a) Physically plausible HOI learning (b) Robotic manipulation test

Figure 6: **Applications.** Two downstream uses of our object-complete, physics-stable reconstructions: (a) physics-based character control [25], (b) cluttered robotic manipulation [6].

requires 47% more resampling calls and time, so gravity-axis stretching serves as an efficiency mechanism. The ablation on GraspClutter6D (Sec. C.3) shows similar trends on real data.

4.4 Applications

Our physically stable 3D reconstruction unlocks a single capability: the scene can be dropped into any simulator as a collection of separately addressable rigid bodies with consistent contact geometry, with no manual scene-authoring step. Fig. 6 presents two downstream systems built on this invariant.

Physics-Based Character Control. We load reconstructed scenes into a physics simulator and train a humanoid control policy following DeVI [25]. As our scene provides physically stable assets, we are able to generate physics-based character motion including dexterous human-object interaction (HOI) in a novel scene by leveraging a video diffusion model as an HOI-aware motion planner.

Robot-arm Manipulation We use our reconstructions as input to a closed-loop VLM agent [6] for text-guided 6D pose prediction, with execution carried out by GraspNet-based grasping [15] and OMPL motion planning [47]. Whereas naive SAM3D leaves occluded geometry incomplete, with objects breaking away from their image-observed positions, our image-aligned reconstructions are object-complete and physically grounded, providing stable grasp targets and collision-free trajectories. The reconstructed scenes therefore serve as a controllable test bed for text-guided robot-arm manipulation.

5 Discussion

We reframe physical simulation from a post-hoc validator into an in-the-loop diagnostic signal that inverts physical violations into geometric corrections, constraining the structural ambiguities of monocular perception. The resulting scenes are directly usable in downstream tasks such as robotic manipulation and reinforcement learning. While our sequential protocol cannot revise early estimates from later evidence, integrating physical dynamics as a generative supervisory signal offers a scalable foundation for reconstructing simulation-ready scenes from single images, with joint scene-level optimization as a natural next step.

Limitations. Our pipeline composes several pretrained modules (segmentation, monocular depth, single-image lifting, pose refinement, and VLM queries), so upstream errors can propagate. Pose initialization is a key bottleneck (Sec. C.4). We assume rigid objects under rigid-body contact, so residuals for articulated or deformable objects may not reflect their true degrees of freedom. The sequential protocol is order-dependent and cannot revise earlier objects from later evidence, and absolute scale remains tied to monocular depth, since gravity-axis residuals reveal support errors rather than metric scale. A more compact formulation that jointly optimizes shape and layout with fewer modules is a natural next step.

Acknowledgments

This work was supported by a research grant from the Youlchon Foundation, Institute of Engineering Research at Seoul National University, the National AI Project of the Ministry of Science and ICT and the AIIS.,NRF grant funded by the Korean government (MSIT) [No. RS-2026-25540546], and IITP grant funded by the Korea government (MSIT) [No. RS-2024-00439854, No. RS-2025-25442338, No. RS-2025-25441838, No. 2022-0-00156, and No. RS-2021-II211343]. H. Joo is the corresponding author.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 4, 20

[2] Rio Aguina-Kang, Kevin James Blackburn-Matzen, Thibault Groueix, Vladimir Kim, and Matheus Gadelha. Seeing through clutter: Structured 3d scene reconstruction via iterative object removal. In *3DV*, 2026. 3

[3] Andreea Ardelean, Mert Özer, and Bernhard Egger. Gen3dsr: Generalizable 3d scene reconstruction via divide and conquer from a single view. In *3DV*, 2025. 7, 9, 15, 22, 24

[4] Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. A general theoretical paradigm to understand learning from human preferences. In *AISTATS*, 2024. 3

[5] Seunghyeok Back, Joosoon Lee, Kangmin Kim, Heeseon Rho, Geonhyup Lee, Raeyoung Kang, Sangbeom Lee, Sangjun Noh, Youngjin Lee, Taeyeop Lee, et al. Graspclutter6d: A large-scale real-world dataset for robust perception and grasping in cluttered scenes. *RA-L*, 2025. 8, 15, 22, 25

[6] Sangwon Baik, Gunhee Kim, Mingi Choi, and Hanbyul Joo. Text-guided 6d object pose rearrangement via closed-loop vlm agents. *arXiv:2604.09781*, 2026. 10, 22

[7] Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. *ICLR*, 2024. 3

[8] 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. *ICLR*, 2026. 4

[9] Boyuan Chen, Hanxiao Jiang, Shaowei Liu, Saurabh Gupta, Yunzhu Li, Hao Zhao, and Shenlong Wang. Physgen3d: Crafting a miniature interactive world from a single image. In *CVPR*, 2025. 3

[10] 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. *CVPR*, 2026. 1, 2, 3, 4, 7, 9, 15, 22, 24

[11] Jaeyoung Chung, Suyoung Lee, Hyeongjin Nam, Jaerin Lee, and Kyoung Mu Lee. Luciddreamer: Domain-free generation of 3d gaussian splatting scenes. *TVCG*, 2025. 2

[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. *CoRL*, 2024. 3

[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. 3

[14] Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Kto: Model alignment as prospect theoretic optimization. *ICML*, 2024. 3

[15] Hao-Shu Fang, Chenxi Wang, Minghao Gou, and Cewu Lu. Graspnet-1billion: A large-scale benchmark for general object grasping. In *CVPR*, 2020. 10, 22

[16] Ruiqi Gao, Aleksander Holynski, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul Srinivasan, Jonathan T Barron, and Ben Poole. Cat3d: Create anything in 3d with multi-view diffusion models. *NeurIPS*, 2024. 2

[17] Google AI Edge. Mediapipe solutions guide. https://developers.google.com/mediapipe, 2025. 22

[18] Haonan Han, Rui Yang, Huan Liao, Jiankai Xing, Zunnan Xu, Xiaoming Yu, Junwei Zha, Xiu Li, and Wanhua Li. Reparo: Compositional 3d assets generation with differentiable 3d layout alignment. In *ICCV*, 2025. 3

[19] Jiwoo Hong, Noah Lee, and James Thorne. Orpo: Monolithic preference optimization without reference model. In *EMNLP*, 2024. 3

[20] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022. 6, 7, 20

[21] Yujia Hu, Songhua Liu, Xingyi Yang, and Xinchao Wang. Flash sculptor: Modular 3d worlds from objects. *arXiv preprint arXiv:2504.06178*, 2025. 3

[22] Siyuan Huang, Siyuan Qi, Yixin Zhu, Yinxue Xiao, Yuanlu Xu, and Song-Chun Zhu. Holistic 3d scene parsing and reconstruction from a single rgb image. In *ECCV*, 2018. 3

[23] Xinyu Huang, Yi-Jie Huang, Youcai Zhang, Weiwei Tian, Rui Feng, Yuejie Zhang, Yanchun Xie, Yaqian Li, and Lei Zhang. Open-set image tagging with multi-grained text supervision. In *MM*, 2025. 4

[24] 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. 2, 3

[25] Hyeonwoo Kim, Jeonghwan Kim, Kyungwon Cho, and Hanbyul Joo. Devi: Physics-based dexterous human-object interaction via synthetic video imitation. In *arXiv:2604.20841*, 2026. 10

[26] Black Forest Labs. FLUX.2: Frontier Visual Intelligence. https://bfl.ai/blog/flux-2, 2025. 19

[27] Qixuan Li, Chao Wang, Zongjin He, and Yan Peng. Phip-g: Physics-guided text-to-3d compositional scene generation. *arXiv preprint arXiv:2502.00708*, 2025. 3

[28] Ruining Li, Chuanxia Zheng, Christian Rupprecht, and Andrea Vedaldi. Dso: Aligning 3d generators with simulation feedback for physical soundness. In *ICCV*, 2025. 3, 6

[29] Zhanhao Liang, Yuhui Yuan, Shuyang Gu, Bohan Chen, Tiankai Hang, Mingxi Cheng, Ji Li, and Liang Zheng. Aesthetic post-training diffusion models from generic preferences with step-by-step preference optimization. In *CVPR*, 2025. 3

[30] Guying Lin, Kemeng Huang, Michael Liu, Ruihan Gao, Hanke Chen, Lyuhao Chen, Beijia Lu, Taku Komura, Yuan Liu, Jun-Yan Zhu, et al. Pat3d: Physics-augmented text-to-3d scene generation. *ICLR*, 2026. 3

[31] 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: A language and vision agentic framework for 3d scene generation. *ICLR*, 2026. 3

[32] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *ICLR*, 2023. 6

[33] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *ICLR*, 2019. 8

[34] Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. *NeurIPS*, 2024. 3

[35] Yanxu Meng, Haoning Wu, Ya Zhang, and Weidi Xie. Scenegen: Single-image 3d scene generation in one feedforward pass. In *3DV*, 2026. 2

[36] Van Nguyen Nguyen, Christian Forster, Sindi Shkodrani, Vincent Lepetit, Bugra Tekin, Cem Keskin, and Tomas Hodan. Gotrack: Generic 6dof object pose refinement and tracking, 2025. 23

[37] Junfeng Ni, Yixin Chen, Bohan Jing, Nan Jiang, Bin Wang, Bo Dai, Puhao Li, Yixin Zhu, Song-Chun Zhu, and Siyuan Huang. Phyrecon: Physically plausible neural scene reconstruction. *NeurIPS*, 2024. 3

[38] Junfeng Ni, Yu Liu, Ruijie Lu, Zirui Zhou, Song-Chun Zhu, Yixin Chen, and Siyuan Huang. Decompositional neural scene reconstruction with generative diffusion prior. *CVPR*, 2025. 3

[39] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *NeurIPS*, 2022. 3

[40] Xiaqing Pan, Nicholas Charron, Yongqian Yang, Scott Peters, Thomas Whelan, Chen Kong, Omkar Parkhi, Richard Newcombe, and Yuheng Carl Ren. Aria digital twin: A new benchmark dataset for egocentric 3d machine perception. In *ICCV*, 2023. 8, 15, 21, 22, 25

[41] Stefan Popov, Pablo Bauszat, and Vittorio Ferrari. Corenet: Coherent 3d scene reconstruction from a single rgb image. In *ECCV*, 2020. 3

[42] Qwen Team. Qwen3.6-35B-A3B: Agentic coding power, now open to all, 2026. FP8 model: https://huggingface.co/Qwen/Qwen3.6-35B-A3B-FP8. 19

[43] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *NeurIPS*, 2023. 3, 6

[44] Xuanchi Ren, Tianchang Shen, Jiahui Huang, Huan Ling, Yifan Lu, Merlin Nimier-David, Thomas Müller, Alexander Keller, Sanja Fidler, and Jun Gao. Gen3c: 3d-informed world-consistent video generation with precise camera control. In *CVPR*, 2025. 2

[45] 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. 7, 9, 15, 22, 24

[46] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025. 15

[47] Ioan A Sucan, Mark Moll, and Lydia E Kavraki. The open motion planning library. *RAM*, 19(4):72–82, 2012. 10, 22

[48] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023. 4, 8, 15, 23

[49] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012. 7, 8

[50] Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *CVPR*, 2024. 3, 6

[51] 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. *NeurIPS*, 2025. 4

[52] Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects. In *CVPR*, 2024. 4, 16, 23

[53] Tong Wu, Guandao Yang, Zhibing Li, Kai Zhang, Ziwei Liu, Leonidas Guibas, Dahua Lin, and Gordon Wetzstein. Gpt-4v(ision) is a human-aligned evaluator for text-to-3d generation. In *CVPR*, 2024. 8, 9

[54] Chong Xia, Kai Zhu, Zizhuo Wang, Fangfu Liu, Zhizheng Zhang, and Yueqi Duan. Simrecon: Simready compositional scene reconstruction from real videos. *CVPR*, 2026. 3

[55] Pradyumna Yalandur Muralidhar, Yuxuan Xue, Xianghui Xie, Margaret Kostyrko, and Gerard Pons-Moll. Physic: Physically plausible 3d human-scene interaction and contact from a single image. In *ACM SIGGRAPH Asia*, 2025. 3

[56] Han Yan, Mingrui Zhang, Yang Li, Chao Ma, and Pan Ji. Phycage: Physically plausible compositional 3d asset generation from a single image. *arXiv preprint arXiv:2411.18548*, 2024. 3

[57] Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *arXiv preprint arXiv:2310.11441*, 2023. 15

[58] Kai Yang, Jian Tao, Jiafei Lyu, Chunjiang Ge, Jiaxin Chen, Weihan Shen, Xiaolong Zhu, and Xiu Li. Using human feedback to fine-tune diffusion models without any reward model. In *CVPR*, 2024. 3

[59] Zesong Yang, Bangbang Yang, Wenqi Dong, Chenxuan Cao, Liyuan Cui, Yuewen Ma, Zhaopeng Cui, and Hujun Bao. Instascene: Towards complete 3d instance decomposition and reconstruction from cluttered scenes. *ICCV*, 2025. 3

[60] Kaixin Yao, Longwen Zhang, Xinhao Yan, Yan Zeng, Qixuan Zhang, Wei Yang, Lan Xu, Jiayuan Gu, and Jingyi Yu. Cast: Component-aligned 3d scene reconstruction from an rgb image. In *SIGGRAPH*, 2025. 2, 3

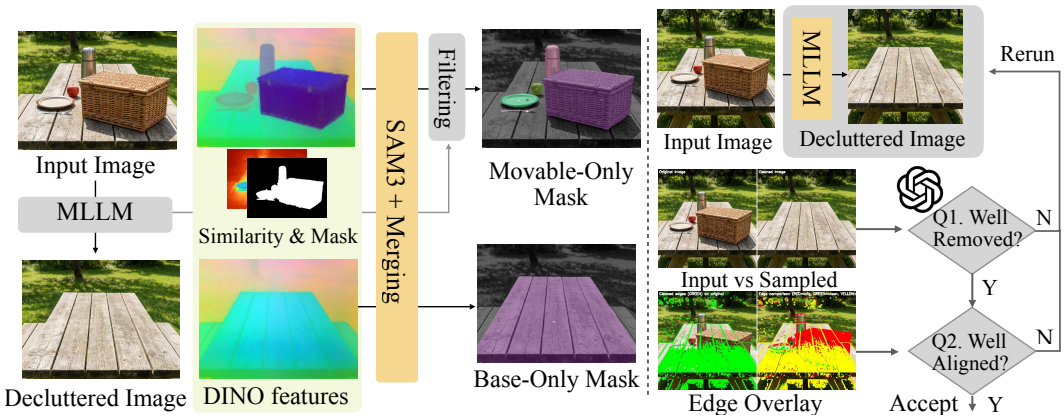
[61] Shaofeng Yin, Jiaxin Ge, Zora Zhiruo Wang, Chenyang Wang, Xiuyu Li, Michael J. Black, Trevor Darrell, Angjoo Kanazawa, and Haiwen Feng. Vision-as-inverse-graphics agent via interleaved multimodal reasoning. *ECCV*, 2026. 3

[62] 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. In *CVPR*, 2026. 3, 7, 9, 15, 22, 24

[63] 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. 2

[64] Wangbo Yu, Jinbo Xing, Li Yuan, Wenbo Hu, Xiaoyu Li, Zhipeng Huang, Xiangjun Gao, Tien-Tsin Wong, Ying Shan, and Yonghong Tian. Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis. *TPAMI*, 2025. 2

[65] Jensen Zhou, Hang Gao, Vikram Voleti, Aaryaman Vasishta, Chun-Han Yao, Mark Boss, Philip Torr, Christian Rupprecht, and Varun Jampani. Stable virtual camera: Generative view synthesis with diffusion models. In *ICCV*, 2025. 2



(a) Distinct Instance Mask Extraction.

(b) Base-Only Image Generation w/ MLLM & VLM filtering

Figure 7: **Distinct Instance Extraction and Decluttered Image I_{base} Generation.** (a) We generate a decluttered image I_{base} via MLLM [48] and compute an object probability map from pixel differences with the original image. Candidate masks in the original image are filtered by their overlap with this map to identify liftable instances, while masks in the decluttered image are selected using Set-of-Mark-based VLM filtering [57] due to the simplified scene layout. (b) We generate a decluttered image by using an MLLM to remove foreground objects. A VLM gate verifies removal completeness and edge alignment with the input, and the generation–verification process is repeated until the acceptance criteria are met.

	GraspClutter6D			AriaDigitalTwin					GenWild		
Method	$\text{IoU}_{\text{fh/fo}}\uparrow$	IoU ↑	Stab. ↑	$\text{IoU}_{\text{fh/fo}}\uparrow$	IoU ↑	3D $\text{IoU}_{\text{fh/fo}}\uparrow$	3D IoU ↑	Stab. ↑	$\text{IoU}_{\text{fh/fo}}\uparrow$	IoU ↑	Stab. ↑
SAM3D [10]	0.055	0.081	0.7%	0.064	0.115	0.027	0.040	0.5%	0.205	0.246	3.2%
Gen3DSR [3] (our bg)	0.007	0.020	0.0%	0.062	0.109	0.064	0.088	0.0%	0.086	0.118	0.4%
Gen3DSR [3]	0.005	0.197	0.0%	0.003	**0.548**	0.001	0.220	0.0%	0.036	0.458	0.0%
3D-RE-GEN [45] (our bg)	0.016	0.032	1.1%	0.041	0.061	0.036	0.051	0.0%	0.090	0.130	5.6%
3D-RE-GEN [45]	0.028	0.054	4.4%	0.025	0.065	0.019	0.042	0.1%	0.081	0.135	6.7%
3D-Fixer [62]	0.088	0.100	3.2%	0.090	0.132	0.110	0.153	0.7%	0.173	0.237	2.4%
Ours	**0.213**	**0.227**	**34.7%**	**0.234**	0.275	**0.239**	**0.277**	**31.5%**	**0.494**	**0.519**	**58.4%**

Table 5: **Supplementary IoU-based quantitative comparison.** 2D IoU on GraspClutter6D [5] and GenWild, and both 2D IoU and 3D IoU on AriaDigitalTwin [40]. All IoUs are measured after the evaluation simulation, and ground-truth objects without a prediction count as zero overlap. Stab. is the ratio of predicted objects that remain free and stable after the evaluation simulation.

A Implementation Details for the Method

This appendix expands the technical details that were deferred from Sec. 3.

A.1 Preprocessing Pipeline

Decluttered image generation. The base structures in Sec. 3.1 are selected per category, so a base structure may consist of one or several object instances. The decluttered image is most beneficial in complex scenes and can be omitted for simple scenes with few objects and little occlusion.

Quality gate for base-only image generation. A VLM checks each base-only image I_{base} produced by the MLLM-based editor for removal completeness and edge alignment. Removal completeness checks whether all foreground objects have been removed. For edge alignment, we overlay edges extracted from I_{base} onto the input image I and ask the VLM whether they align with the base structure in I. This verifies that the base structure is preserved and stays in place. Generation and verification are repeated until both criteria are satisfied.

Cross-path de-duplication via DINOv3. The two extraction paths in Sec. 3.1 can reconstruct the same support structure. SAM3 on I can segment it (e.g., a desk edge) as a placed-object candidate, while SAM3D on I_{base} already reconstructs it as part of the base structure mesh. To detect this, we compare dense DINOv3 [46] feature maps $\mathbf{F}, \mathbf{F}_{\text{base}} \in \mathbb{R}^{H\times W\times C}$ of I and I_{base} through the per-pixel

cosine difference

$$\Delta(\mathbf{p}) = 1 - \frac{\mathbf{F}(\mathbf{p}) \cdot \mathbf{F}_{\text{base}}(\mathbf{p})}{\|\mathbf{F}(\mathbf{p})\| \, \|\mathbf{F}_{\text{base}}(\mathbf{p})\|}. \tag{13}$$

Thresholding Δ at δ_{dino} yields a binary edited-region map $P = \mathbb{1}[\Delta > \delta_{\text{dino}}]$ that marks the pixels of I removed in I_{base}. We retain a placed-object candidate mask M_j only if

$$|M_j \cap P| \,/\, |M_j| > \alpha_{\text{mask}}, \tag{14}$$

so that candidates outside the edited region, which the base path has already lifted, are dropped. Fig. 7 summarizes this pipeline.

Floor and wall plane extraction with Manhattan rectification. We back-project the SAM3 floor and wall masks to 3D with metric depth and fit planes with RANSAC, using residual thresholds of 5 cm for the floor and 1 cm for each wall. We keep up to four walls with the largest coverage whose orientations are mutually parallel or orthogonal. The up direction is given by the cross product of two orthogonal walls, or by the floor normal when no such pair exists, and all wall normals are rectified to be perpendicular to it. After object fitting, we re-estimate the planes with the fitted meshes, which provide a stronger up-direction signal than depth alone, and snap the floor offset to the lowest vertex among all fitted meshes to close the gap between floor-standing objects and the floor. Tables and desks are further refined as described in Sec. A.2.

A.2 Pose Alignment Details

Translation–Scale Initial Alignment. Before applying FoundationPose [52] in Eq. 3, we perform a lightweight translation–scale alignment for each SAM3D object. Although FoundationPose is effective as a local pose refiner, we find that it is sensitive to large errors in the initial scale or object center: when the rendered object is substantially misaligned with the observed mask and depth, the render-and-compare refinement can converge to an incorrect pose or fail to recover. We therefore use the masked depth observation to bring the SAM3D prediction into a reasonable basin before FoundationPose. Given the canonical mesh $\boldsymbol{\mathcal{M}}_i^{\text{init}}$, initial prediction $(\boldsymbol{R}_i^{\text{init}}, \boldsymbol{t}_i^{\text{init}}, s_i^{\text{init}})$, and the target point cloud $\boldsymbol{\mathcal{P}}_i$ back-projected from the masked depth map, we solve only for the isotropic scale and translation $(s_i, \boldsymbol{t}_i)$ while keeping $\boldsymbol{R}_i^{\text{init}}$ fixed. We avoid optimizing rotation in this stage because chamfer-style point-set losses provide weak and often degenerate rotation signals under monocular occlusion, partial visibility, and symmetric or textureless geometry. This step therefore serves as a 4-DoF initialization that corrects scale and centering errors before the rotation-aware refinement.

Since SAM3D meshes are complete while the input depth observes only the visible surface, directly matching all mesh vertices to $\boldsymbol{\mathcal{P}}_i$ biases the alignment toward self-occluded geometry. We therefore rasterize $\boldsymbol{\mathcal{M}}_i^{\text{init}}$ once at the initial pose and keep only the visible vertex subset $\mathcal{V}_i^{\text{vis}}$. The translation–scale alignment is then optimized with a one-sided chamfer objective from the observed point cloud to the visible mesh vertices:

$$\mathcal{L}_{\text{align}}(s_i, \boldsymbol{t}_i) = \frac{1}{|\boldsymbol{\mathcal{P}}_i|} \sum_{\boldsymbol{y} \in \boldsymbol{\mathcal{P}}_i} \min_{\boldsymbol{v} \in \mathcal{V}_i^{\text{vis}}} \left\| \boldsymbol{y} - \left(s_i \boldsymbol{R}_i^{\text{init}} \boldsymbol{v} + \boldsymbol{t}_i \right) \right\|_2 . \tag{15}$$

The asymmetric direction encourages the predicted visible surface to cover the observed depth points, while avoiding penalties from unobserved mesh regions caused by self-occlusion or inter-object occlusion.

However, the one-sided loss alone can favor over-grown meshes, since a large mesh may still cover all observed points. To reduce this failure mode, we run multi-start Adam optimization from several initial scales $s_{i,0}^{(k)} = \rho_k s_i^{\text{init}}$, with $\rho_k \in \{0.5, 0.75, 1.0, 1.25\}$, and apply a weak log-scale prior:

$$\mathcal{L}^{(k)} = \mathcal{L}_{\text{align}}(s_i, \boldsymbol{t}_i) + \lambda \left(\log s_i - \log s_i^{\text{init}} \right)^2 . \tag{16}$$

After convergence, we rank the candidates using a combination of silhouette recall and a weakly weighted bidirectional chamfer score:

$$k^* = \arg\max_k \frac{1}{2} \widetilde{\text{Recall}}^{(k)} + \frac{1}{2} \left(1 - \widetilde{\mathcal{L}}_{\text{bi}}^{(k)} \right), \tag{17}$$

where the recall term penalizes under-coverage of the input mask, and the bidirectional chamfer term discourages excessive scale growth. The selected $(s_i^*, \boldsymbol{t}_i^*)$ is then used as the initialization for FoundationPose, while the rotation remains initialized by $\boldsymbol{R}_i^{\text{init}}$.

Tabletop-Plane Refinement. For tables and desks, even small residual tilt or height errors on the top support surface produce visible floating or penetration artifacts once small objects are placed on top. We therefore refine them after FoundationPose by aligning the mesh top surface to a tabletop plane detected from MoGe depth.

Given the post-refinement mesh and the scene up direction $\mathbf{u}$, we first extract a depth-supported tabletop plane $(\mathbf{n}^d, \mathbf{p}^d)$ by filtering upward-facing pixels within the object mask, applying a height-band constraint to suppress clutter and under-shelf regions, and fitting a plane with RANSAC. We also estimate the mesh top plane $(\mathbf{n}^m, \mathbf{p}^m)$ from canonical top vertices identified along the transformed up direction. The mesh top is then rigidly snapped onto the detected support plane by aligning normals and removing the residual height gap:

$$T_{\text{snap}}(v) = \boldsymbol{R}^{\text{snap}}\, v + (\mathbf{p}^m - \boldsymbol{R}^{\text{snap}}\mathbf{p}^m) + \delta, \tag{18}$$

where $\boldsymbol{R}^{\text{snap}}$ rotates $\mathbf{n}^m$ onto $\mathbf{n}^d$, and $\delta = \big((\mathbf{p}^d - \mathbf{p}^m)\cdot\mathbf{n}^d\big)\,\mathbf{n}^d$ translates the mesh along the detected plane normal to close the residual gap to the support plane.

Starting from the snapped pose, we further optimize four residual degrees of freedom consisting of yaw rotation around the detected plane normal, two in-plane translations, and an isotropic scale. The optimization minimizes a symmetric chamfer between the rendered visible mesh point cloud and the depth-lifted observation, regularized toward the snapped initialization:

$$\min_{\theta, t_x, t_y, \log s} \; \tfrac{1}{2}\big(d_{\hat{P}\to P} + d_{P\to\hat{P}}\big) + \lambda\Big[(\theta/\theta_0)^2 + (t_x/t_0)^2 + (t_y/t_0)^2 + (\log s/\sigma_0)^2\Big]. \tag{19}$$

Refinements with insufficient plane support, non-horizontal detected planes, or floor-plane degeneracies are discarded, and the FoundationPose result is kept.

A.3 Gravity-Based Diagnostics Details

How VLM tags object-specific pose DoF. Indoor scenes routinely contain objects that are not floor-supported, such as picture frames, wall-mounted TVs, and pendant lamps. Without special handling, the physics simulator would drop them to the ground and corrupt the reconstructed layout. To handle these, we compose a two-panel query image from the input image for each object, highlighting the target on the left and isolating it against a desaturated background on the right, and ask a VLM to classify it as standing or hanging. Only objects with their own attachment to a wall or ceiling count as hanging, so an object resting on a wall-mounted shelf remains standing while the shelf is hanging. For hanging objects we additionally elicit a pin count, which the simulator uses to anchor the object with the appropriate number of constraints. The count is one for freely swinging mounts (hooks, pendants) and two for rigid mounts (frames, brackets). The full prompt and qualitative examples are provided in Fig. 8.

Wall-anchoring Mechanism. For each object o flagged as *hanging* by the VLM with pin count $K \in \{1, 2\}$, we attach it to the reconstructed scene wall through K ball-joint anchors. Each anchor pairs a point $\mathbf{p}_o^{(k)}$ on the object's surface with a point $\mathbf{p}_W^{(k)}$ on the wall surface, and the simulator keeps the two points coincident throughout the rollout.

(1) Choosing the object-side pin. We restrict pin candidates to vertices that are simultaneously near the top of the object (along the anti-gravity direction) and on its wall-facing back side, excluding front- or bottom-facing vertices that would otherwise yield an inverted or wall-detached hang. For $K = 1$ (hooks, pendant lamps, and other freely swinging mounts), we take the highest and most laterally centered vertex of this top-back region. For $K = 2$ (picture frames, mounted TVs, and other wall-mounted objects), we take the leftmost and rightmost vertices of this top-back region along the wall-parallel horizontal direction. The object can rotate only about the axis joining the two anchors, so gravity keeps its face flat against the wall.

(2) Pairing each pin with a wall point. Each object-side pin is paired with its closest point on the wall mesh, which is the foot of the perpendicular dropped from the pin onto the wall surface. This

Pose-DoF Extraction Prompt

```
[ROLE]
You are an expert at analyzing how objects are physically supported
in indoor scenes.

[TASK]
Determine whether a specific highlighted object is "standing"
(resting on a surface via gravity) or "hanging" (directly attached
to a wall/ceiling, and would fall if that attachment were removed).
[CRITICAL RULE]
Only classify the object's OWN attachment. Do NOT trace the support
chain further. If an object is simply resting on top of something —
even a wall-mounted shelf — it is "standing". Whether the shelf
itself is hanging is irrelevant.

[INSTRUCTION]
Analyze the highlighted object in the image.

[IMAGE LAYOUT]
- LEFT  : Scene with the target object highlighted
          (colored mask overlay + bounding box + label)
- RIGHT : Same scene with background desaturated to grayscale —
          the full-color region is the target object

[TARGET OBJECT]
- Mesh index  : {idx}
- Description  : "{phrase}"
- Bounding box : [{x1}, {y1}, {x2}, {y2}]

[REASONING PROTOCOL]
Determine this object's physical support type. You MUST reason
through these steps explicitly:

[STEP 1 — IDENTIFY THE OBJECT]
What is this object and where exactly is it in the scene?

[STEP 2 — HOW IS THIS OBJECT ITSELF SUPPORTED?]
Ask: "If I only remove this object's own attachment point, does it fall?"
 - Is it RESTING ON a surface (floor, desk, shelf, another object)?
   Gravity keeps it there with no attachment needed → "standing".
 - Is it DIRECTLY ATTACHED to a wall or ceiling
   (screwed, bolted, nailed, hung on a hook)?
   Removing that attachment causes it to fall → "hanging".
 - STOP HERE. Do NOT ask whether the surface it rests on is itself
   hanging or floor-connected. That is irrelevant.

[STEP 2 — EXAMPLES]
 - Book on a wall-mounted shelf
     → book just RESTS on shelf surface
     → "standing" (the shelf's status is irrelevant)
 - Vase on a floating desk
     → vase RESTS on desk → "standing"
 - Wall-mounted shelf itself
     → shelf is ATTACHED to wall, remove attachment → falls
     → "hanging"
 - Picture frame hung on a nail
     → frame is ATTACHED to wall → "hanging"
 - Coat on a hook
     → coat is ATTACHED to hook → "hanging"
 - TV on a TV stand
     → TV RESTS on stand → "standing"
 - Wall-mounted TV
     → TV is ATTACHED to wall bracket → "hanging"
 - Object leaning against wall but sitting on floor
     → RESTS on floor → "standing"

[STEP 3 — PIN COUNT]  (Only if hanging)
 - 1 pin  : hangs from a single point, could swing/rotate
            (e.g., coat on hook, hanging plant, pendant lamp)
 - 2 pins : rigidly mounted at multiple points, cannot rotate
            (e.g., picture frame, wall-mounted TV, mirror,
             towel rack, shelf brackets)

[OUTPUT FORMAT]
Return ONLY valid JSON (no markdown, no extra text):
{
 "step1_identify"  : "what is this object and where is it?",
 "step2_own_support": "how is THIS object supported?
                       resting on what surface, OR
                       attached to what wall/ceiling point?",
 "support"         : "standing" or "hanging",
 "standing_on"     : "the immediate surface/object it rests on
                      (e.g., 'wall-mounted shelf', 'floor',
                       'wooden desk'). Use null if hanging.",
 "hanging_from"    : "what it is directly mounted/hung on
                      (e.g., 'wall nail', 'ceiling bracket',
                       'wall hook'). Use null if standing.",
 "num_pins"        : 0 for standing, 1 or 2 for hanging
}
```

Example Outputs



A:
- support: hanging
- num_pins: 2
- hanging_from: wall
- standing_on : None

➡ *line-anchored* (1-DoF)



A:
- support: standing
- num_pins: 0
- hanging_from: None
- standing_on : wall-mounted shelf

➡ *free* (6-DoF)

Figure 8: **VLM-based wall-hang detection.** Each object is rendered into a two-panel query image with a highlighted overlay (left) and a color-on-grayscale isolation (right), and a VLM classifies it as standing or hanging under an own-attachment-only rule. Hanging objects are additionally tagged with a pin count (0: free, 1: point-anchored, 2: line-anchored).

preserves the natural standoff distance between the object and the wall rather than collapsing the object onto the wall plane.

(3) Force coupling between the two points. Each pin pair is realized as a MuJoCo ball-joint equality constraint that keeps the two points coincident at every simulation step. The constraint is enforced as a stiff, critically damped spring rather than a hard weld, so contact transients are absorbed without instability.

Physics-mode parameters. The simulator runs at a timestep of $1/480$ s with inelastic, stiff contacts, and the sliding and torsional friction coefficients are 1.0 and 0.95, respectively.

Penetration Resolution. For each newly placed object, we resolve its overlap with the floor, the rectified walls, and every previously placed object while keeping it as the only dynamic body. Floor and wall contacts take priority over contacts with previously placed objects. At each iteration, the object is translated along the depth-weighted average normal of the highest-priority contacts by their maximum penetration depth plus a 7 mm margin. Objects that remain in contact after 200 iterations or whose cumulative displacement exceeds 0.3 m skip the gravity-based diagnostics and shape correction, and are retained in the final scene.

A.4 Shape Correction Details

Gravity-axis Stretch Details. For objects corrected by gravity-axis stretching, we use the signed displacement η_i from Eq. (9). We orient the gravity-aligned OBB axis such that $(\hat{\mathbf{u}}_i^{\mathrm{grv}})^\top \hat{\mathbf{g}} \geq 0$, and define its canonical-frame counterpart as

$$\hat{\mathbf{a}}_i^{\mathrm{grv}} = (\mathbf{R}_i^{\mathrm{pre}})^\top \hat{\mathbf{u}}_i^{\mathrm{grv}}. \tag{20}$$

The stretch factor is

$$\lambda_i = 1 + \eta_i. \tag{21}$$

We first apply anisotropic stretching in the canonical mesh frame:

$$\bar{\boldsymbol{\mathcal{M}}}_i^{\mathrm{str}} = \mathrm{Stretch}_{\hat{\mathbf{a}}_i^{\mathrm{grv}}} \left(\boldsymbol{\mathcal{M}}_i^{\mathrm{init}}; \lambda_i \right), \tag{22}$$

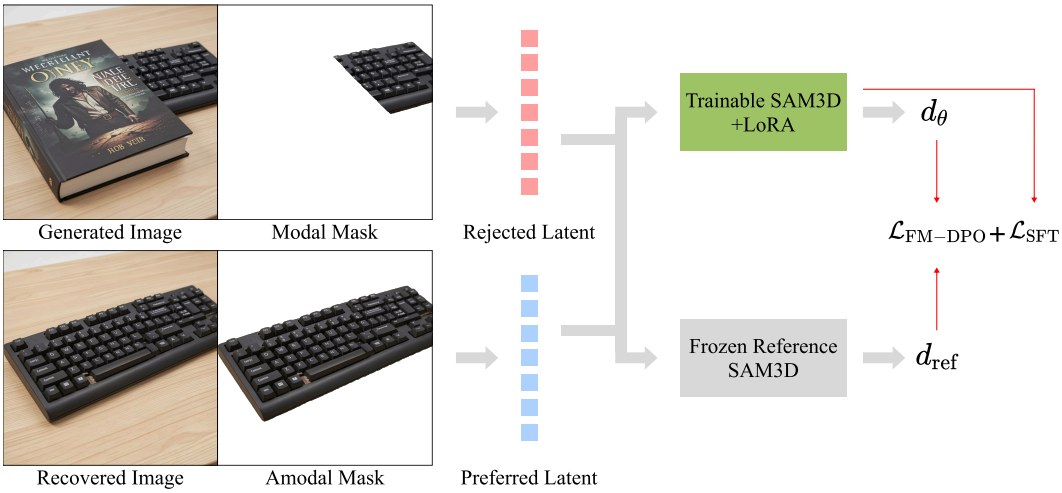


Figure 9: **SAM3D Fine-tuning Pipeline.** We fine-tune SAM3D for amodal shape resampling using synthetic preference pairs that contrast occlusion-failed latents with completed amodal latents. A flow-matching DPO objective with an additional preferred-sample SFT term updates only LoRA adapters while keeping the reference SAM3D frozen, and the resulting model is used at test time for physics-informed resampling.

where $\mathrm{Stretch}_{\hat{\mathbf{a}}}(\boldsymbol{\mathcal{M}}; \lambda)$ anisotropically scales the mesh by factor λ along axis $\hat{\mathbf{a}}$ while leaving the orthogonal directions unchanged. For a vertex $\mathbf{v}$ of a canonical mesh centered at the origin, this operation is

$$\mathbf{v}' = \lambda_i \left(\mathbf{v}^\top \hat{\mathbf{a}}_i^{\text{grv}}\right) \hat{\mathbf{a}}_i^{\text{grv}} + \left(\mathbf{I} - \hat{\mathbf{a}}_i^{\text{grv}} (\hat{\mathbf{a}}_i^{\text{grv}})^\top\right) \mathbf{v}. \tag{23}$$

Let γ_i be the longest side length of the AABB of $\bar{\boldsymbol{\mathcal{M}}}_i^{\text{str}}$. To preserve the canonical-mesh convention, we renormalize the stretched mesh and absorb the normalization factor into the object scale:

$$\boldsymbol{\mathcal{M}}_i^{\text{str}} = \frac{1}{\gamma_i} \bar{\boldsymbol{\mathcal{M}}}_i^{\text{str}}, \qquad s_i^{\text{str}} = \gamma_i s_i^*. \tag{24}$$

Finally, we update the translation so that the OBB face opposite to gravity remains fixed:

$$\mathbf{t}_i^{\text{str}} = \mathbf{t}_i^{\text{pre}} + \frac{1}{2} \eta_i \ell_i^{\text{pre}} \hat{\mathbf{u}}_i^{\text{grv}}. \tag{25}$$

Thus, stretching changes the object geometry, scale, and translation, while preserving the corrected rotation from diagnostic simulation.

SAM3D Fine-tuning Details. We construct two types of preference pairs for fine-tuning SAM3D under the resampling input in Eq. (10). First, for occlusion-completion pairs, we synthesize paired-object scenes using a text-to-image generator [26], where one object substantially occludes the target object. An image-to-image editor [26] then removes the occluding object and completes the full shape of the target object. From the original occluded image, we extract the modal mask $\boldsymbol{M}_i$, while from the edited image, we extract a full-object mask $\boldsymbol{M}_i'$ that serves as an amodal crop mask. We run SAM3D on both images to obtain shape latents: the latent from the occluded input is treated as the rejected sample x_i^-, and the latent from the edited full-object input is treated as the preferred sample x_i^+. During fine-tuning, the condition $c_i = (\boldsymbol{I}, \boldsymbol{M}_i, \boldsymbol{I}_{\boldsymbol{M}_i'}, \boldsymbol{D})$ follows the same input setting as Eq. (10).

We additionally construct clean preservation pairs to avoid degrading SAM3D's original input behavior. For these pairs, we synthesize scenes containing an unoccluded object, obtain its segmentation mask, and run SAM3D to produce a clean latent as x_i^+. We then corrupt the mask by dropping part of the segmentation and run SAM3D again, using the resulting latent as x_i^-. Thus, occlusion-completion pairs teach amodal completion under the new crop setting, while clean preservation pairs regularize the model to retain its original modal-mask reconstruction ability. Before training, we filter preferred shape samples using a VLM [42] given the original image and six rendered views of each preferred

mesh. We retain only samples that pass checks for a single target, correct identity, complete major parts, and consistency with the visible shape.

We fine-tune SAM3D using the combined objective in Eq. (12). Below, we detail its two components: $\mathcal{L}_{\text{FM-DPO}}$ and $\mathcal{L}_{\text{SFT}}$. Let $\ell_\theta^{\text{shape}}(x, c, t, \epsilon)$ denote the shape-branch flow-matching loss of SAM3D for latent x under condition c, timestep t, and noise ϵ. For each preference pair, we compute the win–lose loss difference of the current model as

$$d_{\theta,i} = \ell_\theta^{\text{shape}}(x_i^+, c_i, t_i, \epsilon_i^+) - \ell_\theta^{\text{shape}}(x_i^-, c_i, t_i, \epsilon_i^-), \tag{26}$$

and the corresponding difference of the frozen reference model as

$$d_{\text{ref},i} = \ell_{\theta_{\text{ref}}}^{\text{shape}}(x_i^+, c_i, t_i, \epsilon_i^+) - \ell_{\theta_{\text{ref}}}^{\text{shape}}(x_i^-, c_i, t_i, \epsilon_i^-), \tag{27}$$

where θ_{ref} is the frozen base SAM3D model. The current model and the reference model are evaluated on the same condition, timestep, and noise for each preference pair. The FM-DPO term optimizes preference improvement relative to the frozen reference. The additional SFT term in Eq. (12) is defined as

$$\mathcal{L}_{\text{SFT}}(\theta) = \mathbb{E}_i \left[w_i \, \ell_\theta^{\text{shape}}(x_i^+, c_i, t_i, \epsilon_i^+) \right]. \tag{28}$$

This term directly minimizes the preferred-sample flow-matching loss using the same preferred latent, condition, timestep, and noise as the FM-DPO term.

The sample weight w_i is defined by the pair type:

$$w_i = \begin{cases} 1.0, & \text{if } i \text{ is an occlusion-completion pair,} \\ 0.1, & \text{if } i \text{ is a clean preservation pair.} \end{cases} \tag{29}$$

We apply these pair weights to both the FM-DPO and SFT terms, assigning a larger weight to occlusion-completion pairs because they directly target the amodal resampling failure mode, while clean preservation pairs act as a weaker regularizer.

We do not fine-tune the full SAM3D generator. Instead, we attach LoRA [20] adapters only to the attention projections of the shape generation branch and update these adapters during training. This limits adaptation to the shape latent pathway while preserving the base model's original reconstruction behavior.

A.5 Sequential Composition Details

Ordering strategy. Dynamic objects are ordered by the height of their world-frame bounding-box bottom along the up direction, and the lowest object is placed first.

Wall-attached object handling. Hanging objects identified by the VLM [1] (Sec. A.3) remain held by their K anchors during penetration resolution, so a single-pin object retains 3-DoF rotation about its anchor point and a two-pin object retains 1-DoF rotation about the line joining its anchors. Without these anchors, hanging objects released as fully dynamic bodies tend to rotate or fall away from the wall under penetration-resolution forces, since their visible silhouette underdetermines the wall-normal contact geometry.

Final-pass full settling. Once sequential penetration resolution has placed every object into a collision-consistent assembled scene, we run a single final settling simulation over the whole scene. Unlike the diagnostic probe, which keeps only one object dynamic and freezes it at first contact, here all *free* objects are simultaneously dynamic, simulated as full 6-DoF rigid bodies and allowed to settle jointly under gravity for a fixed duration T_{settle}, while *point-anchored* and *line-anchored* objects stay pinned at their anchors and move only by rotation about the anchor point (3-DoF) or anchor line (1-DoF), respectively. This pass corrects minor residual motion left by sequential placement and yields a globally consistent layout. Objects whose displacement in this pass exceeds a divergence bound are reverted to their pre-settle pose.

B Experimental Details

This section provides details of the quantitative, qualitative, and application experiments.

B.1 Metrics

Physical plausibility metrics. We evaluate reconstruction quality at *post-sim*, the configuration obtained after a $5\,\mathrm{s}$ MuJoCo gravity-settle simulation, and we report both mask-side and shape-side metrics. For mask-side metrics, we render all meshes jointly and compare their corresponding visible rendered mask against input segmentation. We then report (i) **2D IoU**, the mean IoU between each ground-truth mask and its identity-paired prediction, and (ii) **2D ABO**,

$$\mathrm{ABO}_{\mathrm{2D}} \;=\; \frac{1}{N_{\mathrm{gt}}} \sum_{i=1}^{N_{\mathrm{gt}}} \max_{j=1,\dots,M_{\mathrm{pred}}} \mathrm{IoU}(g_i, p_j)\,,$$

with the denominator fixed at the ground-truth count so that missing or filtered-out predictions are penalized as misses rather than hidden. Shape-side metrics are computed against the ADT object library. Each ground-truth mesh is placed in the scene frame using its provided pose, transformed into the input camera frame, and sampled with $10\mathrm{k}$ surface points.

We report **3D IoU** as the mean axis-aligned-bounding-box (AABB) IoU over matched (GT, pred) pairs and **3D ABO**,

$$\mathrm{ABO}_{\mathrm{3D}} \;=\; \frac{1}{N_{\mathrm{gt}}} \sum_{i=1}^{N_{\mathrm{gt}}} \max_{j} \mathrm{IoU}_{\mathrm{3D}}\big(B_{g_i}, B_{p_j}\big)\,,$$

again with denominator N_{gt}. We compute all AABBs in the shared input camera frame.

Free/hanging variants (fh/fo). We compute each metric under two settings, fh and fo, and report their average, e.g. $\mathrm{ABO}_{\mathrm{fh/fo}} = \frac{1}{2}(\mathrm{ABO}_{\mathrm{fh}} + \mathrm{ABO}_{\mathrm{fo}})$ (likewise for IoU). Both settings keep the denominator N_{gt} but take the maximum only over predictions that are valid after the evaluation simulation: fh excludes *penetrating* predictions (defined below), and fo additionally excludes *hanging* ones, so a ground-truth object without a valid prediction scores zero.

Penetration criterion and Stability. An object is **penetrating** if any of its post-simulation contacts has a penetration depth above $\varepsilon = 2\,\mathrm{mm}$. We report *Stab.* as the fraction of all simulated predicted objects that end the evaluation simulation free and stable. Let $\mathcal{O}$ denote the predicted objects entered into the evaluation simulation, and $\mathcal{O}_{\mathrm{free}} \subseteq \mathcal{O}$ those that are neither hanging nor penetrating at the end of the simulation. An object is considered stable if its final displacement $\Delta(o)$ from the initial pose is below $5\,\mathrm{cm}$.

$$\mathrm{Stab.} = \frac{|\{o \in \mathcal{O}_{\mathrm{free}} : \Delta(o) < 0.05\,\mathrm{m}\}|}{|\mathcal{O}|} \times 100.$$

B.2 Mesh Quality Evaluation Details

For each instance that triggers resampling, we render every method's mesh from the input camera and arrange the input image with its visible-mask overlay and two candidate renderings in a horizontal triptych. A VLM judge then selects the more plausible full-object completion of the partially observed occluded instance. Each method pair is queried in both orders to counter left/right position bias.

Given the resulting win counts A_{ij} (number of times method i was preferred over j, with ties contributing $+1$ to both A_{ij} and A_{ji}), we fit Elo ratings r_i, initialized to 1000, by minimizing the Bradley–Terry negative log-likelihood with the standard Elo scale $c = 400$,

$$\mathcal{L}(r) \;=\; \sum_{i \neq j} A_{ij}\, \log\!\big(1 + 10^{(r_j - r_i)/c}\big), \quad P(i \succ j) \;=\; \frac{1}{1 + 10^{(r_j - r_i)/c}}. \tag{30}$$

B.3 Aria Digital Twin Preprocessing Details

We build a scenario-balanced evaluation set of 40 frames drawn from 24 Aria Digital Twin (ADT) [40] sequences, selected with five sequences each from the *clean*, *decoration*, *meal*, and *work* scenarios and four from *recognition*. We exclude the multi-user sequences, where third-party hands appear in frame, and the skeleton-annotated releases, whose motion-capture skeleton pixels contaminate the ground-truth segmentation. Within each retained sequence, we pick up to two frames maximally

separated in time that satisfy five conditions. (i) GT depth, segmentation, and 3D bounding boxes are available. (ii) The head-mounted camera is near-stationary, with linear and angular velocities below $0.6\,\mathrm{m/s}$ and $0.6\,\mathrm{rad/s}$. (iii) Every movable object is at rest, moving less than $10\,\mathrm{mm}$ within $\pm 5\,\mathrm{s}$, so that the GT 3D state matches the RGB frame. (iv) At least ten non-structural objects are visible, with at least 60% of their pixels inside the central $60\% \times 60\%$ region. (v) MediaPipe [17] detects no hands in the frame or its 90°-rotated version. Each accepted frame is then undistorted to a pinhole camera, cropped to the largest valid square, and rotated 90° clockwise so that gravity points downward.

Ground-truth per-object masks are taken from the ADT instance segmentation, excluding architectural elements (floors, walls, doors, and ceilings), instances without a GT 3D bounding box, and instances smaller than 100 pixels or 0.1% of the image. Built-in furniture and large appliances (e.g., beds, kitchen islands, and refrigerators) are kept as physical objects, yielding on average 26 objects per scene, which are passed to SAM3D to generate the input meshes.

B.4 Application Details

Robot-arm Manipulation. We adopt the closed-loop VLM agent of VLMPose [6] as our text-guided 6D goal-pose predictor, using GPT-5.2 with the authors' default configuration. The predicted goal pose is passed to GraspNet [15] for grasp proposals on the target object's mesh-sampled point cloud, and OMPL [47] plans a rectangular trajectory that lifts, translates, and lowers the object with a Franka arm to reduce inter-object collisions.

C Additional Experimental Results

C.1 Additional Quantitative Comparisons

We benchmark Ours against SAM3D [10], Gen3DSR [3], 3D-RE-GEN [45], and 3D-Fixer [62] on GraspClutter6D [5], Aria Digital Twin (ADT) [40], and GenWild, reporting 2D IoU (fh/fo and identity-paired) and Stab., plus 3D IoU on ADT. For Gen3DSR and 3D-RE-GEN we also evaluate a variant that swaps in our background reconstruction ("our bg") to isolate the foreground predictor from the layout.

Tab. 5 shows that Ours achieves the highest $\mathrm{IoU}_{\mathrm{fh/fo}}$ and Stab. on all three datasets and the highest 3D $\mathrm{IoU}_{\mathrm{fh/fo}}$ and 3D IoU on ADT, with Stab. margins of 30.3 to 51.7 percentage points over the strongest baseline. On identity-paired IoU, Ours leads on GraspClutter6D and GenWild, while Gen3DSR scores higher on ADT (0.548 vs. 0.275) because most of its objects interpenetrate (93.3% in Tab. 1) and stay fused near their initial poses, which preserves their projected masks. The Stab. margin reflects the role of gravity-based diagnostics and shape correction in producing layouts that physically settle, rather than merely projecting plausibly into the input view (see also Tab. 2 and Tab. 7).

C.2 Correction Threshold Analysis

We compare gravity-axis stretching and amodal shape resampling on 708 ADT objects to examine the correction threshold $\tau = 0.15$. The objects are partitioned into four disjoint ranges of the normalized gravity-axis displacement ρ_i defined in Eq. (8). Since ADT provides ground-truth (GT) object meshes, we evaluate shape quality directly by aligning each reconstructed mesh to its GT mesh with ICP and normalizing ADD-S by the GT object diameter. As shown in Tab. 6, the two branches yield nearly identical error for $\rho_i < 0.15$ (0.0277 vs. 0.0281), whereas resampling reduces the error by 15.3%, 13.3%, and 62.8% in the three ranges with $\rho_i \geq 0.15$. These results support stretching below $\tau = 0.15$ and resampling at or above it on this evaluation set. The resulting trade-off between shape quality and resampling time is summarized in Tab. 4.

C.3 Progressive Ablation on Real Data

We repeat the progressive ablation of Tab. 2 on GraspClutter6D and ADT, adding one stage at a time to the SAM3D baseline. As shown in Tab. 7, alignment and penetration resolution each increase ABO and $\mathrm{ABO}_{\mathrm{fh/fo}}$ on both datasets, and gravity correction reduces D_{mean} from 0.342 to 0.234 on GraspClutter6D and from 0.586 to 0.357 on ADT. As on GenWild, the full model achieves the highest ABO and $\mathrm{ABO}_{\mathrm{fh/fo}}$ on both datasets, and it also attains the lowest D_{mean} on ADT. Two

ρ_i range	#Objects	ADD-S$_{\text{norm}}$ ↓		Change
		Stretch	Resample	
$\rho_i < 0.15$	226	**0.0277**	0.0281	$+1.4\%$
$0.15 \leq \rho_i < 0.30$	108	0.0321	**0.0272**	-15.3%
$0.30 \leq \rho_i < 1.0$	120	0.0353	**0.0306**	-13.3%
$\rho_i \geq 1.0$	254	0.1131	**0.0421**	-62.8%
All evaluated objects	708	0.0603	**0.0334**	-44.6%

Table 6: **Shape quality of stretching and resampling across ρ_i ranges on ADT.** ADD-S$_{\text{norm}}$ is ADD-S after rigid ICP alignment to the GT mesh, normalized by the GT object diameter. Change is the relative error of resampling with respect to stretching, so negative values indicate lower error with resampling. Bold marks the lower error in each row.

trade-offs appear. On ADT, gravity correction slightly lowers ABO from 0.216 to 0.211, and on GraspClutter6D, shape correction raises D_{mean} from 0.234 to 0.250.

Variants	GraspClutter6D			ADT		
	ABO ↑	ABO$_{\text{fh/fo}}$ ↑	D_{mean} ↓	ABO ↑	ABO$_{\text{fh/fo}}$ ↑	D_{mean} ↓
SAM3D	0.121	0.087	0.638	0.147	0.092	0.721
+Align.	0.186	0.150	0.494	0.189	0.138	0.630
+Pen.	0.240	0.223	0.342	0.216	0.165	0.586
+Grv.	0.258	0.242	**0.234**	0.211	0.165	0.357
+Re. (Full)	**0.267**	**0.248**	0.250	**0.299**	**0.256**	**0.311**

Table 7: **Progressive ablation on real data.** Each row adds one stage to the row above, following Tab. 2. Align., Pen., Grv., and Re. denote alignment, penetration resolution, gravity correction, and shape correction with resampling.

C.4 Upstream Module Ablations

We examine the sensitivity of our pipeline to two upstream modules on GenWild while keeping the downstream pipeline unchanged. First, we replace FoundationPose [52] with GoTrack [36] for object pose initialization. Second, we reconstruct the base structures from the original image instead of the image decluttered with Gemini [48]. As shown in Tab. 8, replacing FoundationPose with GoTrack lowers ABO$_{\text{fh/fo}}$ from 0.507 to 0.260 and raises D_{mean} from 0.128 to 0.337, E_{pk} from 0.42 to 1.65, and Penetr. from 2.5% to 5.6%. Removing decluttering lowers ABO$_{\text{fh/fo}}$ from 0.507 to 0.444 and raises D_{mean} from 0.128 to 0.181. Of the two modules, pose initialization has the larger impact in these experiments, which makes it the more critical upstream bottleneck. These ablations are not an exhaustive oracle replacement of every pretrained module, but they show how changes in these two components affect the final reconstruction.

Variant	ABO$_{\text{fh/fo}}$ ↑	ABO ↑	D_{mean} ↓	E_{pk} ↓	Penetr. ↓
Pose initialization					
FoundationPose (Ours)	**0.507**	**0.532**	**0.128**	**0.42**	**2.5%**
GoTrack	0.260	0.290	0.337	1.65	5.6%
Base decluttering					
w/ Gemini (Ours)	**0.507**	**0.532**	**0.128**	–	–
w/o Gemini	0.444	0.473	0.181	–	–

Table 8: **Upstream module ablations on GenWild.** Each block changes one upstream module and keeps the rest of the pipeline fixed. Bold marks the best value in each column within each block. Dashes denote metrics not reported for the decluttering ablation.

C.5 Robustness to Pose Perturbation

To evaluate robustness to inaccurate pose initialization, we perturb the initial rotation of each object on GenWild before the subsequent correction stages. The perturbation rotates the object about a randomly sampled axis by an angle drawn from a Gaussian distribution with mean 45° and standard deviation 15°. As shown in Tab. 9, stretching alone improves all three metrics over no shape correction, which corresponds to the +Grv. setting of Tab. 2. Adding resampling improves them further, raising $\text{ABO}_{\text{fh/fo}}$ from 0.3107 to 0.3320 and lowering D_{mean} from 0.4771 to 0.3905. All perturbed variants still have lower ABO and $\text{ABO}_{\text{fh/fo}}$ and higher D_{mean} than the unperturbed full model in Tab. 2 (0.532, 0.507, and 0.128). Shape correction therefore partially mitigates inaccurate pose initialization rather than fully recovering from it.

Correction	$\text{ABO}_{\text{fh/fo}} \uparrow$	ABO $\uparrow$	$D_{\text{mean}} \downarrow$
No shape correction	0.2512	0.2699	0.5318
Stretch only	0.3107	0.3304	0.4771
Stretch + Resample (Ours)	**0.3320**	**0.3576**	**0.3905**

Table 9: **Recovery under rotation perturbation on GenWild.** The initial rotation of each object is perturbed about a random axis by an angle drawn from a Gaussian distribution with mean 45° and standard deviation 15°. Bold marks the best value in each column.

C.6 Appearance Metrics

ABO measures mask overlap with the input view and therefore does not depend on the textures of the generated objects, which are not the focus of our work. For completeness, we also report the appearance metrics PSNR, LPIPS, and CLIP distance on GraspClutter6D and GenWild, which are strongly affected by these generated textures. As shown in Tab. 10, Ours achieves the best value on all three metrics on both datasets.

	GraspClutter6D			GenWild		
Method	PSNR $\uparrow$	LPIPS $\downarrow$	CLIP $\downarrow$	PSNR $\uparrow$	LPIPS $\downarrow$	CLIP $\downarrow$
SAM3D [10]	11.09	0.381	0.272	11.59	0.405	0.220
Gen3DSR [3] (our bg)	10.97	0.387	0.357	10.34	0.449	0.344
Gen3DSR [3]	10.91	0.379	0.357	9.88	0.444	0.395
3D-RE-GEN [45] (our bg)	10.98	0.381	0.358	10.73	0.426	0.357
3D-RE-GEN [45]	11.08	0.378	0.371	9.88	0.438	0.423
3D-Fixer [62]	10.97	0.367	0.294	10.85	0.411	0.288
Ours	**13.65**	**0.306**	**0.207**	**14.41**	**0.314**	**0.138**

Table 10: **Appearance metrics on GraspClutter6D and GenWild.** CLIP denotes CLIP distance. Columns marked $\uparrow$ ($\downarrow$) indicate higher (lower) is better.

C.7 Runtime Analysis

We measure the per-stage runtime of our pipeline on GenWild and report it in Tab. 11. GPU-based stages run on a single NVIDIA RTX PRO 6000 Blackwell GPU, the same GPU used for training, and the MLLM and VLM stages report end-to-end external API latency. The full pipeline takes 347.7 s per scene, about 2.13$\times$ the 163.5 s of preprocessing plus SAM3D lifting. The largest additional costs come from resampling (52.4 s), decluttered image generation (46.9 s), and FoundationPose pose alignment (38.9 s), while plane extraction and V-HACD decomposition add little overhead.

#	Stage	Time (s)	Relative
1	Preprocessing + SAM3D lifting (baseline)	163.5 ± 27.3	100%
	Preprocessing (depth + masks)	4.7 ± 7.6	2.9%
	SAM3D lifting (incl. base)	158.8 ± 26.2	97.1%
2	Decluttered image generation (MLLM)	46.9 ± 7.0	28.7%
3	Pose-DoF estimation (VLM)	18.1 ± 10.4	11.1%
4	Plane extraction	2.4 ± 1.9	1.5%
5	V-HACD decomposition	5.7 ± 3.3	3.5%
6	FoundationPose pose alignment	38.9 ± 4.6	23.8%
7	Physics simulation + setup	19.8 ± 3.6	12.1%
8	Resampling	52.4 ± 13.6	32.0%
	Total	347.7 ± 45	212.7%

Table 11: **Per-stage runtime on GenWild.** Time is mean ± std per scene. Relative runtime is normalized to preprocessing plus SAM3D lifting. MLLM and VLM stages report external API latency.

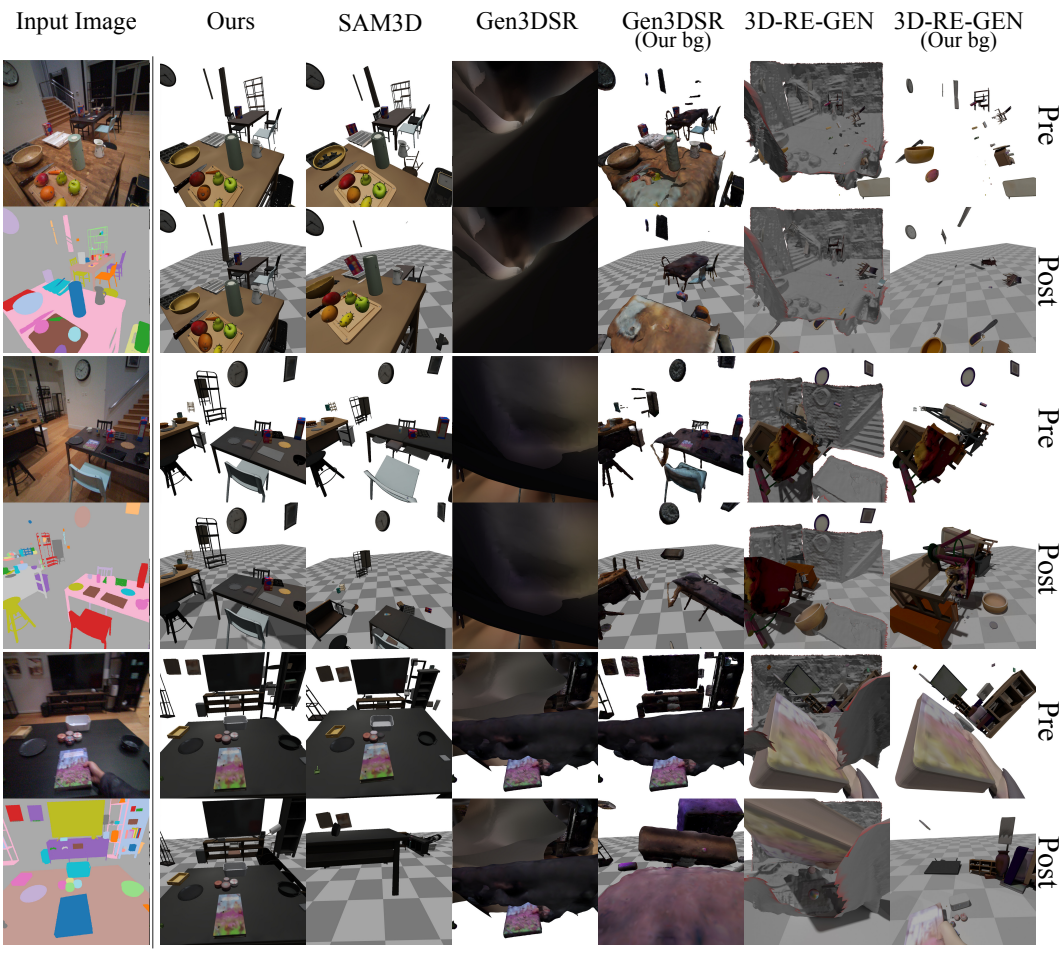


Figure 10: **Qualitative Comparison of ADT.** Additional qualitative comparisons on the ADT dataset.

C.8 Additional Qualitative Comparisons

We supplement the qualitative comparisons in the main paper with further results on the Aria Digital Twin (ADT) [40] and GraspClutter6D [5] datasets. Fig. 10 extends the ADT comparisons, while Fig. 11 extends those on GraspClutter6D.

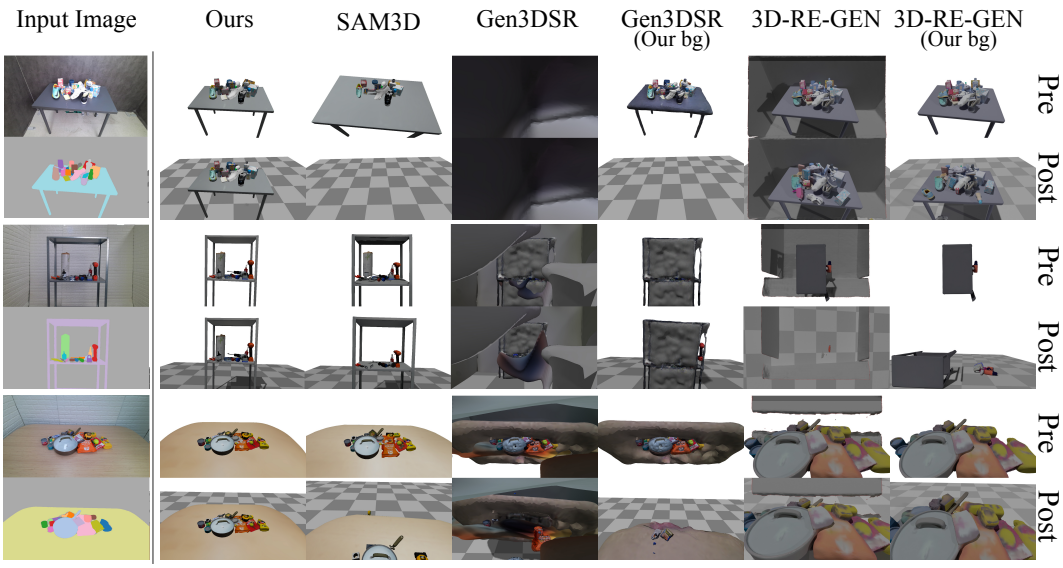


Figure 11: **Qualitative Comparison of GraspClutter6D.** Additional qualitative comparisons on the GraspClutter6D dataset.

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: The contribution of our paper is clearly summarized in the last paragraph of the introduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discuss the limitations of our method in the Limitations paragraph of Sec. 5.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

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: The paper does not include theoretical results or formal proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

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: We have tried to include as many details as possible for reproduction in the main paper and supplementary materials. However, since text information alone may be insufficient, we plan to release the code.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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: [No]

Justification: The code will be released at https://github.com/snuvclab/SimuScene.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

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: We have tried to provide all possible information regarding training and test details in the paper and supplementary materials.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

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: We do not report error bars or confidence intervals, as we did not repeat our main experiments multiple times.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

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: Sec. 4.1 specifies the training settings on a single NVIDIA RTX PRO 6000 Blackwell GPU, and Supp. Sec. C.7 reports per-stage runtime (mean $\pm$ std) on the same GPU, including the latency of external MLLM and VLM APIs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

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: Our research adheres to the NeurIPS Code of Ethics. We ensured respect for privacy and dignity, did not involve human subjects or sensitive data, and considered potential societal impacts and harms. No ethical concerns arose during the development or evaluation of our method.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [No]

    Justification: Since we are solving the problem of restoring a simulation-ready 3D scene from a single RGB image, we believe there are no factors that will cause negative societal impacts.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

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: We believe that our dataset and model do not pose a high risk of misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

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: We have properly cited all the papers we referenced.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [No]

    Justification: We used our own image dataset in the experiment but did not submit it. However, an explanation for this exists in the paper.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

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: [N/A]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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: [N/A]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

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: MLLMs and VLMs are core components of our method, used for decluttered image generation, removal and alignment checks, pose-DoF tagging, and instance labeling. A VLM also serves as the mesh quality judge in our evaluation and as the goal-pose predictor in the robotics application, as described in Secs. 3 and 4 and the supplementary material.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.