CODIMRECON: AGENTIC RECONSTRUCTION OF SIM-READY 3D SCENES WITH DEFORMABLE CURVES, SURFACES, AND VOLUMES

Anonymous authors
Paper under double-blind review

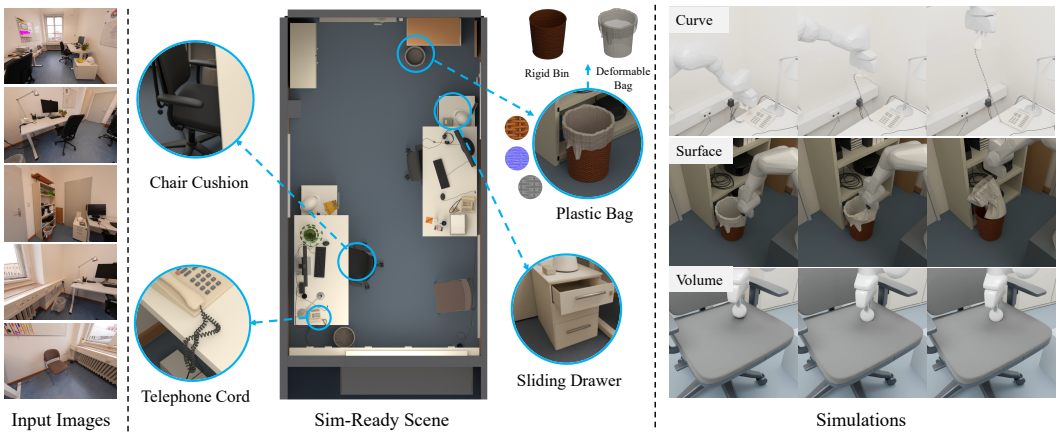


Figure 1: **CoDimRecon reconstructs a simulation-ready scene from multi-view RGB images.** Left: input views of an office. Center: top-down reconstruction with an articulated drawer and deformable examples—a coiled telephone cord (curve), plastic bag (surface), and chair cushion (volume). The bin and the bag are separate objects, rigid and deformable, respectively. Right: robot interactions with the reconstructed cord, plastic bag, and cushion.

ABSTRACT

Reconstructing simulation-ready 3D scenes from real-world observations enables robotics, gaming, and immersive applications, yet existing methods largely assume rigid objects. This leaves an important gap for deformables, whose simulation-ready geometry depends on dimensionality—curves, surfaces, or volumes—and whose behavior may require models beyond elasticity. We present CoDimRecon, an agentic framework that reconstructs editable scenes containing rigid, articulated, and deformable objects from multi-view RGB observations. Scene-level geometric priors ground scale and layout, while object-level generated meshes guide the agent toward detailed, compact geometry; articulated rigid objects are decomposed into movable parts with explicit joints. For deformables, category-wise agent sessions reconstruct curves as centerlines with radii, surfaces as manifold shells with thickness, and volumes as watertight solids for volumetric meshing. Reusable simulator skills initialize compatible physical models and parameters, while agent-guided behavioral tests expose mismatches and trigger targeted revisions of motion, geometry, numerics, or material modeling. On evaluated Replica and ScanNet++ scenes, CoDimRecon achieves competitive compositional reconstruction accuracy while additionally producing deformable assets for rod, shell, and solid simulation. We further demonstrate robot interactions across all three representations, including a controlled paper-folding case in which behavioral testing motivates plastic bending.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

Reconstructing simulation-ready indoor scenes from captured observations turns real environments into reusable digital worlds. Such scenes support applications ranging from robot learning (Nasiriany et al., 2024) and computer games (Li et al., 2024) to immersive content creation (Luo et al., 2025). Given a captured video or multi-view RGB images, we aim to recover a compositional 3D scene in which the room structure is explicit and each persistent object has its own geometry, pose, material, and physical model, with explicit articulation for movable rigid parts when applicable.

Existing compositional scene reconstruction methods are either optimization-based or zero-shot. Optimization-based approaches (Wu et al., 2023; Ni et al., 2024; Xia et al., 2025; Ni et al., 2025; Yang et al., 2025; Liu et al., 2024) optimize decomposed representations from multi-view images and masks, while zero-shot approaches compose pretrained reconstruction modules (Xia et al., 2026a; Dong et al., 2026) or directly infer object-centric scenes (Siddiqui et al., 2026; Wu et al., 2026; Xia et al., 2026b). However, these methods largely treat scene objects as rigid bodies, leaving deformable reconstruction and simulation underexplored.

Thin structures make this gap especially important. In physics-based simulation (Li et al., 2026b), cables, cloth, paper, and other slender bodies are often not discretized as ordinary 3D solids: resolving a very small thickness volumetrically can require fine through-thickness resolution and, with standard low-order formulations, can suffer locking as structures become thinner (Bischoff et al., 2004). Instead, decades of simulation research have developed codimensional models that represent rods as curves and shells as surfaces embedded in 3D (Grinspun et al., 2003; Bergou et al., 2008). Dimensional reduction avoids explicitly meshing thickness, but introduces additional bending and, for rods, twisting mechanics. It therefore changes the reconstruction target itself: a cable needs a continuous centerline and radius, a sheet a manifold midsurface and thickness, and a volumetric soft body a watertight solid suitable for volumetric meshing. Irreversible deformation adds another practical gap. Everyday interactions such as folding paper cannot be represented by elasticity alone; they require a plastic model that retains deformation after unloading. Existing compositional reconstruction pipelines with physical modeling largely focus on rigid-body physics and have not been designed around either codimensional geometry or such irreversible behavior.

Recent multimodal agents provide a natural interface between visual observations, 3D authoring tools, and physical simulators (OpenAI, 2026). We therefore investigate agentic reconstruction of simulation-ready scenes containing rigid objects, including articulated ones, alongside deformable objects. Direct agentic reconstruction from images alone can misjudge scale and layout or replace visible detail with coarse approximations. Deformables add a further challenge: their representation, physical model, and parameters cannot be determined from static geometry alone, and whether they support the intended interaction often becomes evident only when exercised in simulation.

To address these challenges, we introduce **CoDimRecon**, an agentic framework for reconstructing simulation-ready scenes with rigid objects and deformable curves, surfaces, and volumes. Scene-level geometric priors ground scale and layout, while generated object meshes provide detailed shape references for rebuilding compact, editable geometry. For articulated rigid objects, the agent separates movable parts and assigns joints; for deformables, CoDimRecon reconstructs each geometric dimension separately: curves are completed into centerlines with radii, surfaces are repaired into manifold shells with thickness, and volumes are closed for tetrahedral meshing. Reusable simulator skills then initialize compatible physical models and parameters.

Finally, CoDimRecon uses agent-guided behavioral tests to check whether each deformable asset supports its intended interaction. The agent plans preset diagnostic manipulations, checks each rollout against physical acceptance criteria and expected behavior, and attributes failures to motion, geometry, numerics, or the material model. Material models are revised only when a behavioral test exposes a mismatch; for example, paper that springs back after folding triggers plastic hinge bending. This closes the loop from visual reconstruction to deformable assets represented and tested in the form required by downstream simulation.

Our contributions can be summarized as follows:

- We present CoDimRecon, an agentic framework for reconstructing editable, simulation-ready indoor scenes from RGB observations, using complementary geometric and generative references to improve scene layout and object geometry.

- We extend simulation-ready reconstruction to deformables across geometric dimensions, producing solver-compatible curves, surfaces, and volumes for rod, shell, and solid simulation and initializing their physical models from reusable simulator skills.
- We close the reconstruction-to-simulation loop with agent-guided behavioral tests that trigger targeted revisions of motion, geometry, numerics, or material modeling. Paper folding provides a controlled elastic/plastic example, while the framework remains competitive on compositional reconstruction metrics across the evaluated Replica and ScanNet++ scenes.

2 Related Work

Multi-View Compositional Scene Reconstruction. This task reconstructs a scene as individually represented objects and their spatial arrangement from captured images or video. Existing approaches optimize object-level signed distance fields (Wu et al., 2023; Ni et al., 2024; 2025), incorporate generative priors for partial-observation completion (Yang et al., 2025; Xia et al., 2025; Siddiqui et al., 2026; Wu et al., 2026; Xia et al., 2026b), retrieve reusable CAD assets (Huang et al., 2025; Yu et al., 2025), or assemble instances from pretrained 3D generators (Xia et al., 2026a; Dong et al., 2026). Concurrently, several agentic pipelines have emerged for captured-scene authoring (Qin et al., 2026; Huang et al., 2026; Chen et al., 2026). As concurrent works, they are not directly comparable as empirical baselines: Lucida (Qin et al., 2026) and Lumera (Chen et al., 2026) rely on task-specific fine-tuning or RL policies for layout parsing and pose refinement, while LiteReality-Agent (Huang et al., 2026) requires privileged RGB-D scans and structural RoomPlan layouts rather than multi-view RGB alone.

Method	Input	Geometric Guidance	Generative Prior	Auto Instance Discovery	Training-Free	Agentic	Deformable Simulation
HoloScene	RGB, Mask, Cam	✓	✓	✗	✗	✗	✗
SimRecon	RGB	✓	✓	✓	✓	✗	✗
ReplicateAnyScene	RGB	✓	✓	✓	✓	✗	✗
VIGA	Single-view RGB	✗	✓	✓	✓	✓	✗
Lucida	RGB	✓	✓	✓	✗	✓	✗
Lumera	Single-view RGB	✓	✓	✓	✗	✓	✗
LiteReality-Agent	Posed RGBD	✓	✓	✓	✓	✓	✗
GPT-6 Astra	RGB	✗	✗	✓	✓	✓	✗
Our Method	RGB	✓	✓	✓	✓	✓	✓

Table 1: **Comparison of compositional scene-reconstruction methods.** Cam: camera parameters; RGBD: RGB images with depth.

Tab. 1 compares these methods with ours. Among the listed methods, none reconstructs deformable curves, surfaces, and volumes for physical simulation or models plastic behavior. CoDimRecon additionally supports articulated rigid parts, uses generated meshes as references rather than final assets, and verifies reconstructed deformables through simulated robot interaction.

Deformable and Codimensional Simulation. Physics-based simulation has developed mature models for rods, shells, volumetric solids, contact across mixed dimensions, and inelastic materials, but these methods generally assume solver-ready geometry and material models as input. Appendix A reviews this literature in more detail; CoDimRecon targets the complementary problem of reconstructing such assets from visual observations.

Single-Image Compositional Scene Reconstruction. Gen3DSR (Ardelean et al., 2025), SceneMaker (Shi et al., 2026), and TabletopGen (Wang et al., 2026b) recover compositional scenes from a single image. VIGA (Yin et al., 2026) reconstructs editable scene programs through a code–render–inspect loop from a single view; because it takes one image rather than multi-view observations of a specific room, its setting differs from ours and thus we do not evaluate against it. ϕ-Scene (Li et al., 2026a), REST3D (Ma et al., 2026), and SimuScene (Lee et al., 2026) refine object layout through rigid-body physics. Our method instead uses multi-view geometric evidence and extends simulation-ready reconstruction to representation-specific deformable curves, surfaces, and volumes, with material models revised when required by the target behavior.

Text-Driven Scene Synthesis. SceneSmith (Pfaff et al., 2026) and SAGE (Xia et al., 2026c) generate 3D environments from language or task specifications, while MUSE (Xu et al., 2026b) supports incremental construction and local editing through explicit requirements and verification. PAT3D (Lin et al., 2026) uses differentiable rigid-body simulation to refine text-generated scene layouts, and GIF (Xu et al., 2026a) targets functional object compositions with geometric and physical guidance. GS-Agent (Zhang et al., 2026) integrates a physics engine in an agentic loop to tune material parameters for text-driven 4D world generation. These methods synthesize scenes to satisfy user specifications; our task is to reconstruct the geometry, layout, and deformable objects of a particular observed environment, with deformable assets verified through simulated robot contact.

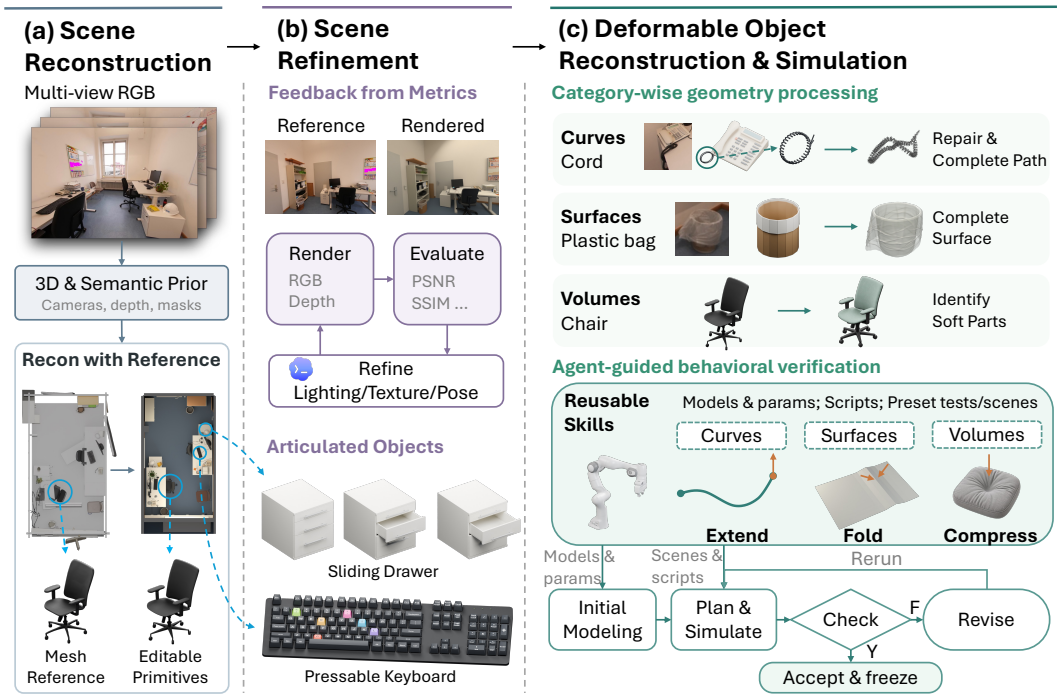


Figure 2: **Overview.** (a) Geometric context and generated meshes guide editable primitive-based scene reconstruction. (b) Render–evaluate–refine improves appearance and pose; articulated rigid objects receive joints, and rigid bodies are settled under gravity. (c) Separate sessions reconstruct solver-compatible curves, surfaces, and volumes; reusable skills initialize physical models, and agent-guided robot tests diagnose motion, geometry, or numerical issues before material changes.

3 METHOD

As shown in Fig. 2, CoDimRecon takes multi-view RGB observations $\mathcal{I} = \{I_v\}_{v=1}^{V}$ and proceeds in three stages. The agent first authors an editable scene using geometric context and generated meshes as references (Sec. 3.1), then refines appearance, articulation, and rigid-body stability (Sec. 3.2). Finally, it reconstructs deformable curves, surfaces, and volumes and tests their behavior through simulated robot interaction (Sec. 3.3).

3.1 RECONSTRUCTION WITH GEOMETRIC AND GENERATIVE REFERENCES

In preliminary image-to-Blender trials, direct agentic reconstruction showed three recurring limitations: errors in scene scale and layout, coarse approximations of visible shape details, and limited use of external 3D tools even when available. We therefore make scene-level geometric context and object-level generated meshes explicit inputs to a fixed reconstruction workflow.

Geometric Context. VGGT-Omega (Wang et al., 2026a) provides camera intrinsics, camera-to-world poses, depth maps D_v, and back-projected point maps. The agent queries this shared metric context when estimating object dimensions and poses and when comparing renders with the observations, grounding object-level fitting in the scene layout.

Obtaining Mesh References. Before agent authoring, we construct an object-level reference scene. CropFormer (Lu et al., 2023) extracts entity masks per frame; we back-project and cluster them across views by 3D overlap following MaskClustering (Yan et al., 2024), with InstaScene's under-segmentation filter (Yang et al., 2025). For each 3D track, SAM3D (SAM-3D-Team et al., 2025) generates a mesh from the most informative view, which is registered to the scene following ReplicateAnyScene (Dong et al., 2026). The agent retrieves the reference nearest to a target object's point cloud. Small missing objects are recovered with REST3D (Ma et al., 2026) within their supporting-

object regions. These meshes serve only as structural references; Appendices B.1 and B.3 give details, and Appendix D.1 compares this design with 2D mask propagation.

Workflow and Primitive Fitting. The agent loads the reference scene into Blender, removes redundant or erroneous objects, and refines the room envelope using the geometric context. For furniture and regular rigid objects, the generated mesh remains a shape reference rather than the final asset: the agent rebuilds the object from Blender primitives shaped with bevel, subdivision, and lattice modifiers. This yields compact, editable geometry, lets the agent correct implausible reference regions against the observations, and exposes parts for articulation. Deformable objects use the representations of Sec. 3.3.

3.2 Scene Refinement

Rendering Enhancement with Metric Evaluation. Agent-authored scenes often differ from the observations in lighting and appearance, so the agent runs a render–evaluate–refine loop with quadratic color alignment (Zhang et al., 2025) as a diagnostic. Using iteratively reweighted least squares, it fits a per-channel mapping $c' = a\,c^2 + b\,c + d$ from rendered to reference pixel values. A large improvement after alignment suggests a photometric mismatch in lighting or materials; little improvement directs attention to geometry or pose. The agent edits the corresponding scene components and reevaluates the input views. We keep these edits explicit rather than baking a 3DGS appearance onto the mesh, which can entangle illumination with albedo and become inconsistent under simulation lighting.

Articulated Object Reconstruction. We provide a joint-modeling guide covering revolute, prismatic, screw, cylindrical, universal, and spherical joints. The agent separates independently movable rigid parts and assigns explicit joints, including fine components such as keyboard keys and telephone buttons when applicable. This decomposition also exposes geometry that might otherwise be collapsed into a coarse textured proxy.

Rigid-Body Stabilization. Small pose errors can leave objects floating or interpenetrating and destabilize downstream simulation. We therefore import the scene into MuJoCo (Todorov et al., 2012), treat all objects as rigid at this stage, settle them under gravity, and write the resulting poses back as the canonical placements. Deformable simulation is handled separately in Sec. 3.3.

3.3 Category-wise Deformable Object Reconstruction and Simulation

Among the compositional reconstruction methods in Tab. 1, none reconstructs deformable curves, surfaces, and volumes for physical simulation. We therefore reconstruct deformables in representation-specific forms—curves, surfaces, and volumes—and configure them for downstream simulation via agent-guided behavioral verification.

Deformable Object Reconstruction. Simulation requires geometry compatible with its discretization. The required repairs depend on geometric dimension: curves such as cables need path completion and radius estimation; surfaces such as clothing and paper need hole patching and fragment merging into a manifold shell with thickness; and volumes such as cushions need watertight closure for volumetric meshing. We preserve this dimensionality in simulation, using rod, shell, and solid discretizations for curves, surfaces, and volumes, respectively. We process the three categories in separate agent sessions; a cable ablation shows that combining all instructions degrades curve reconstruction (Fig. 7).

Physical Modeling from Simulator Examples. Reconstructed geometry does not determine the material and contact properties needed for simulation. Our backend uses IPC-family contact (Li et al., 2020), while reusable skills package available constitutive models, runnable example scenes, scripts, and candidate initial parameters. The geometric representation determines the discretization—rod for curves, shell for surfaces, and solid finite elements for volumes—while the agent selects an asset-appropriate material model and adapts the closest reference configuration. Thus the telephone cord uses a discrete elastic rod, paper uses a shell model with plastic hinges, and the cushions use solid finite elements. Tab. 2 summarizes the resulting parameters, and Appendix B.4 gives the discretizations.

Object	Model	ρ ($\mathrm{kg/m^3}$)	E (MPa)	ν	r/h (mm)	κ_Y ($\mathrm{m^{-1}}$)	δ (mm)	μ
Telephone cord	Discrete elastic rod	1600	40	0.35	1.3	–	0.15	0.60
Paper	StVK membrane, plastic hinges	761.9	2250	0.15	0.105	25	0.50	0.40
Chair cushion	StVK–Hencky solid	100	0.03	0.20	–	–	0.50	0.40
Beanbag	Stable Neo-Hookean solid	120	0.02	0.30	–	–	0.50	0.40

Table 2: **Deformable simulation parameters.** ρ: density; E: Young's modulus (paper: membrane and bending; cord: stretching, bending, and twisting); ν: Poisson's ratio; r/h: cord radius or shell thickness; κ_Y: plastic-hinge yield curvature; δ: contact activation distance; μ: friction coefficient. Dashes denote inapplicable entries. The cord's stress-free natural shape is the reconstructed coil, so the rod keeps its coiled shape without load. Values are effective simulation parameters, not measured material properties; paper plasticity and the cushion's StVK–Hencky model were selected through behavioral testing. Appendix B.4 gives discretization, boundary conditions, and tasks.

Rest Shape. Reconstructed deformable geometry is not necessarily in equilibrium under gravity, so each asset is settled before interaction. Curves and surfaces are imported directly; if settling produces excessive deformation, the agent revises the geometry and repeats the process. For high-polygon volumes, we instead build a closed low-poly proxy, bind the detailed mesh to it, tetrahedralize the proxy, and settle that representation. The resulting rest shape is written back to Blender while preserving the simulation parameters and editable asset.

Agent-Guided Behavioral Verification. Simulator examples provide only an initialization, so we use behavioral tests to check whether each asset supports its intended interaction. The agent plans preset diagnostic manipulations rather than requiring real-robot rollouts for calibration (Zhang et al., 2025). The robot lifts a telephone handset to extend the cord, presses and releases a chair cushion at three locations, and folds paper that should retain a crease after release. Each task has physical validity checks, such as bounds on element stretch and inversion, together with task-level acceptance criteria. The agent specifies grasp and approach poses, gripper actuation, and arm motion; the simulator executes the task offline, and the agent reviews the rendered rollout and diagnostic audit.

When a rollout fails, the agent diagnoses the failure in a fixed order: commanded motion, tool, and contact location; geometry and boundary conditions; then numerical settings. The material model changes only if the required behavior still cannot be reproduced, after which it remains fixed for that asset. For paper, springback after release triggers plastic hinge bending; replaying the same trajectory with plasticity disabled isolates its effect on crease retention (Fig. 5, Tab. 4). Because no measured real dynamics are available, accepted values are effective properties under the assumed model, not recovered material parameters. Appendix B.4 lists the tasks and acceptance criteria, and Tab. 6 records each revision.

4 Experiments

4.1 Setup

Datasets. We evaluate three Replica (Straub et al., 2019) scenes and three ScanNet++ (Yeshwanth et al., 2023) scenes from the HoloScene (Xia et al., 2025) release, covering varied indoor layouts, object density, and lighting.

Baselines. 1) HoloScene (Xia et al., 2025) is an optimization-based method for simulation-ready 3D worlds. We run its official code with the same instance masks, VGGT-Omega cameras, and depth used by our pipeline. **2) ReplicateAnyScene** (Dong et al., 2026) is a zero-shot compositional pipeline; we use its default VLM + SAM3 segmentation and reimplement the unreleased pose-alignment and relation-reasoning modules. **3) GPT-6 Astra** (OpenAI, 2026) is an agent-only baseline that directly authors each scene in Blender from the RGB frames with reasoning effort `xhigh`. SimRecon (Xia et al., 2026a) appears in the capability comparison (Tab. 1) but not the quantitative benchmark because its object-completion and FoundationPose-based object-pose refinement modules were not publicly available at evaluation time. Appendix D details the baseline implementations, and Appendix E gives the GPT-6 Astra prompt.

Datasets	Method	Geometry			Rendering			Latent Similarity		Rigid Stability	
		CD↓	F1↑	NC↑	PSNR↑	SSIM↑	LPIPS↓	Layout↑	Motion↑	Stable (Ground) % ↑	Stable (All) % ↑
Replica	HoloScene	6.79	53.48	78.76	21.22	0.6881	0.4171	94.46	95.71	77.77	49.41
	ReplicateAnyScene	41.88	18.74	61.82	11.17	0.5529	0.6461	76.84	80.88	96.97	69.47
	GPT-6 Astra	11.79	52.91	73.12	14.07	0.5671	0.5440	97.24	95.57	92.27	87.97
	Our Method	6.63	52.24	79.76	15.06	0.5695	0.4817	95.14	96.90	100.00	97.54
ScanNet++	HoloScene	27.15	28.49	65.24	18.79	0.7134	0.3959	80.20	81.73	55.40	42.03
	ReplicateAnyScene	50.73	19.08	54.69	8.94	0.5483	0.6710	92.87	74.87	78.66	82.72
	GPT-6 Astra	22.73	35.90	70.14	14.22	0.5579	0.5356	94.47	96.31	96.30	96.31
	Our Method	21.66	38.28	74.91	16.68	0.6440	0.3728	96.30	97.04	97.05	96.31

Table 3: **Compositional scene-reconstruction results.** CoDimRecon produces the most physically stable reconstructions, with the lowest CD and highest NC on both datasets, while keeping input-view rendering second only to HoloScene in PSNR and SSIM. Dark and light blue mark the best and second-best results; ties share the same color.

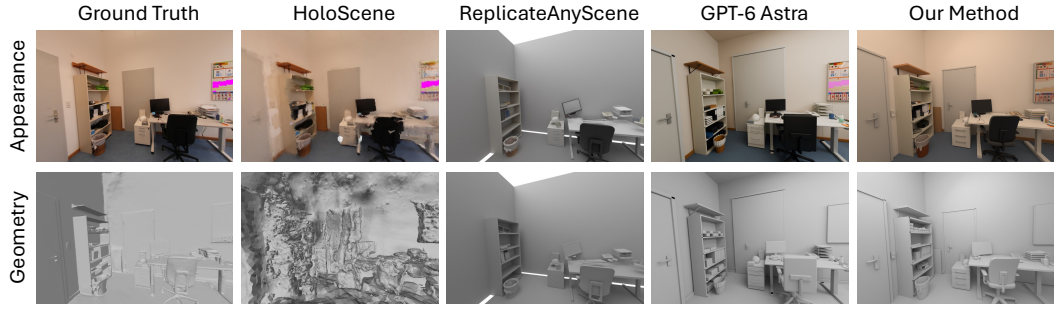


Figure 3: **Qualitative compositional scene-reconstruction comparison.** We compare rendered appearance (top) and geometry (bottom) on ScanNet++ scene `67d702f2e8`. Our method reconstructs complete, compact object geometry at faithful scale, from the rear door and wall shelf to the bookcase contents and swivel chair. By contrast, HoloScene's surfaces are fragmented, ReplicateAnyScene omits the doors and poster, and GPT-6 Astra enlarges the rear door. Our rendering is clean, without the blur and artifacts around the chair and bookcase in HoloScene's rendering.

Metrics. Geometry is evaluated with Chamfer distance (CD, cm), F1@5 cm, and normal consistency (NC). Rendering uses PSNR, SSIM, and LPIPS on the same input views used for reconstruction, so these scores measure observation fidelity rather than novel-view synthesis. Latent Similarity (Tang et al., 2026) compares matched source and rendered clips in a frozen V-JEPA 2.1 encoder (Mur-Labadia et al., 2026), with Layout and Motion as feature-space similarity measures. As a rigid-body stability proxy, we release each dynamic object individually in MuJoCo while all others remain fixed and report the fraction that stay in place: Stable(Ground) for floor-contact objects and Stable(All) for all dynamic objects. Appendix C gives the full protocols.

4.2 Results

Comparison with Baselines. Tab. 3 reports scene-level results. Among the evaluated methods, ours achieves the lowest CD, the highest NC, and the best rigid-body stability on both datasets, while keeping input-view rendering second only to HoloScene in PSNR and SSIM. On Replica, however, the geometry margins over HoloScene are small, and HoloScene and GPT-6 Astra lead on F1. GPT-6 Astra shares our agent backbone but reconstructs directly from RGB; Sec. 4.3 studies the effect of our reference inputs. The higher PSNR and SSIM of HoloScene come with visibly fragmented geometry around the shelf and desk (Fig. 3). PSNR and SSIM compare pixels, so they penalize the lighting and material differences that remain in our explicitly authored scenes (Sec. 3.2). Layout and Motion instead compare source and rendered clips in the feature space of a frozen video encoder. In this space, our renders are closer to the source than HoloScene's on both datasets. This suggests that our lower PSNR stems mainly from appearance differences rather than missing or misplaced scene content. Because these feature-space scores also depend on rendering style, we use them as complementary evidence (Appendix C.3).



(a) Telephone cord.



(b) Paper.



(c) Cushion.

Figure 4: **Deformable simulation examples.** Robot interactions with a reconstructed telephone cord (curve), paper (surface), and chair cushion (volume), shown left to right.

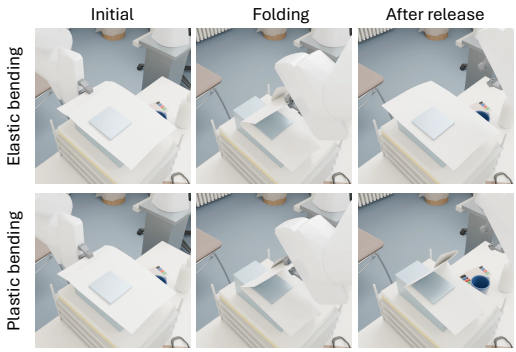


Figure 5: **Paper-folding behavioral test.** Under the same robot trajectory, elastic bending (top) springs back after release, while plastic bending (bottom) retains the fold. The elastic case is a controlled rerun with plasticity disabled.

Bending	Fold angle (°)			Retained (%)	Yielded
	Peak	+1 s	End		
Elastic	121.66	4.48	2.67	3.7	0
Plastic	124.24	114.97	114.15	92.5	372

Table 4: **Paper-folding comparison.** +1 s: one second after opening; Retained: +1 s angle divided by peak angle; Yielded: hinges whose natural angle changes by more than 0.1 rad. See Appendix B.4 for settings and angle definition.

Deformable Object Simulation Results. Among the evaluated reconstruction baselines, none outputs deformable assets for physical simulation (Tab. 1), so we report capability demonstrations rather than a cross-method quantitative comparison. We simulate reconstructed curve, surface, and volume assets with an inserted robot and replay the trajectories in Blender for rendering. The telephone cord extends as the handset is raised (Fig. 4a), the paper retains its crease after release (Fig. 4b), and the cushion indents under three presses and recovers (Fig. 4c). To make the indentation visible, the cushion is rendered with modified surface material and lighting; the simulation is unaffected. Demos of the reconstructed scenes and deformable objects can be viewed as a web page from the supplementary material (Appendix F).

Behavioral Testing and Revision: Paper. The first folding rollout passed the physical-validity checks but sprang back after release, exposing a behavioral mismatch. The agent therefore enabled plastic hinge bending with yield curvature $25\,\mathrm{m}^{-1}$ and no hardening. To isolate this model change, we replay the same $140°$ gripper trajectory with plasticity enabled or disabled while holding the remaining parameters fixed (Fig. 5); the elastic control is a subsequent rerun rather than the original trial. One second after opening, the fold angle is $4.48°$ with elastic bending and $114.97°$ with plastic bending (Tab. 4), while both rollouts remain physically valid. This controlled case illustrates how behavioral testing can expose a material-model mismatch not detected by numerical validity checks. Tab. 6 records the revisions across assets.

The rest-shape settling study on a reconstructed beanbag is reported in Appendix B.4 (Fig. 9).

4.3 Ablation Study

Reference Priors. We remove the geometric and generative references one at a time and measure geometry on ScanNet++ scene `67d702f2e8` (Tab. 5, top); Fig. 6 shows representative visual differences. Without geometric references, CD increases by 64.9%, the largest increase among the geometry variants, and NC decreases by 8.2%, whereas F1 changes little. These changes are consistent with the scale and layout errors visible in Fig. 6, such as oversized cabinets. Without generative references, F1 and NC show the largest drops among the geometry variants, decreasing by 13.0%

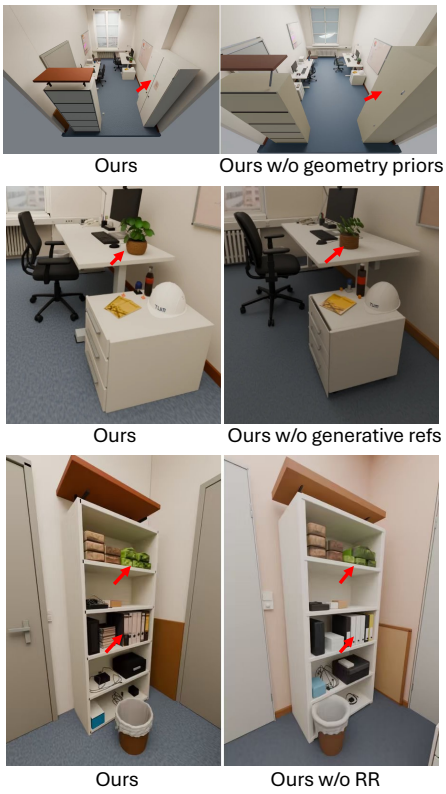


Figure 6: **Qualitative component ablation.** Each row compares the full method with one component removed; red arrows mark affected regions.

Geometry Variant	CD↓	F1↑	NC↑
Full	7.24	44.98	75.93
w/o geometric refs	11.94	44.53	69.70
w/o generative refs	9.86	39.12	68.14
w/o RR	7.36	43.72	75.41
Rendering Variant	PSNR↑	SSIM↑	LPIPS↓
Full	16.68	0.6440	0.3728
w/o RR	15.82	0.6343	0.3938

Table 5: **Component ablation.** Geometry on ScanNet++ scene `67d702f2e8`; rendering averaged over the evaluated ScanNet++ scenes. RR: rendering refinement.



Figure 7: **Separate-session ablation on cables.** Combined deformable-category instructions (left) produce coarser cable geometry than a dedicated curve session (right).

and 10.3% relative to the full method, consistent with coarser object shapes when generated meshes are unavailable as references. Because this geometry ablation covers a single scene, we read these differences as indicating the role of each reference rather than as benchmark-level gains.

Rendering Refinement. As shown in the bottom of Tab. 5, we evaluate the render–evaluate–refine loop of Sec. 3.2 over the evaluated ScanNet++ scenes. Without rendering refinement (RR), PSNR and SSIM decrease by 5.2% and 1.5%, and LPIPS increases by 5.6% relative to the full method. On the geometry-ablation scene, the same variant increases CD by only 1.7%, so in this ablation the loop mainly improves appearance rather than geometry.

Deformable-Category Instructions. In Sec. 3.3, we reconstruct curves, surfaces, and volumes in separate agent sessions. To test this choice, we compare a single session that receives the instructions for all three categories with a dedicated curve session. As shown in Fig. 7, the combined session produces coarser and less plausible cable geometry than the dedicated session.

5 Conclusion

We presented CoDimRecon, an agentic framework for reconstructing editable, simulation-ready scenes with rigid objects and deformable curves, surfaces, and volumes. Geometric and generative references guide scene authoring, while representation-specific reconstruction produces solver-compatible assets for rod, shell, and solid simulation. Agent-guided behavioral tests then expose mismatches and trigger targeted revisions; paper folding provides a controlled example in which an elastic model passes numerical-validity checks yet requires plastic bending to retain the fold. Across the evaluated Replica and ScanNet++ scenes, CoDimRecon remains competitive on compositional reconstruction metrics while extending the output to simulation-ready deformables.

6 AI USE STATEMENT

Our method uses GPT-6 Astra as the agent backbone to execute the reconstruction workflow, category-wise deformable procedures, and simulation feedback loops designed in this work. It is also used as a baseline for direct scene reconstruction (Appendix E). The authors reviewed agent outputs and reported results for correctness; this review is not treated as an algorithmic component of the method. Additionally, generative AI tools were used to assist with editing portions of the manuscript text and software code. We have reviewed all AI-assisted output and take responsibility for the final content of this work.

7 ETHICS STATEMENT

Our experiments use the publicly released Replica and ScanNet++ datasets. Reconstructing real interiors can expose private spaces, so the method should be applied only to captures made with the consent of the occupants.

8 REPRODUCIBILITY STATEMENT

Our experiments use the publicly released Replica and ScanNet++ datasets. Appendix B expands the reconstruction stage and the deformable simulation settings, and Tab. 2 lists the material parameters. Appendix C specifies the evaluation protocol, including alignment, point sampling, and rendering with fixed random seeds. Appendix D describes the baseline reimplementations, and Appendix E gives the GPT-6 Astra baseline prompt. We will release the code for both reconstruction and simulation.

REFERENCES

Andreea Ardelean, Mert Özer, and Bernhard Egger. Generalizable 3D scene reconstruction via divide and conquer from a single view. In *International Conference on 3D Vision (3DV)*, 2025. URL https://andreeadogaru.github.io/Gen3DSR/.

David Baraff and Andrew Witkin. Large steps in cloth simulation. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 43–54, 1998. doi: 10.1145/280814.280821.

Miklós Bergou, Max Wardetzky, Stephen Robinson, Basile Audoly, and Eitan Grinspun. Discrete elastic rods. *ACM Transactions on Graphics*, 27(3):63:1–63:12, 2008. doi: 10.1145/1399504.1360662.

Manfred Bischoff, Kai-Uwe Bletzinger, Wolfgang A. Wall, and Ekkehard Ramm. Models and finite elements for thin-walled structures. In Erwin Stein, René de Borst, and Thomas J. R. Hughes (eds.), *Encyclopedia of Computational Mechanics*, pp. 59–137. Wiley, 2004. doi: 10.1002/0470091355.ecm026.

Robert Bridson, Ronald Fedkiw, and John Anderson. Robust treatment of collisions, contact and friction for cloth animation. *ACM Transactions on Graphics*, 21(3):594–603, 2002. doi: 10.1145/566654.566623.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. Sam 3: Segment anything with concepts, 2025. URL https://arxiv.org/abs/2511.16719.

Junhao Chen, Xinghao Chen, Henghaofan Zhang, Zihao Qiao, Saining Zhang, Yongzhi Li, Ruqi Huang, Sisi Li, Yimin Sheng, Jianyi Zhu, and Hao Zhao. Engine-native editable 3d world reconstruction with objects and lighting, 2026. URL https://arxiv.org/abs/2607.20889. Preprint.

Mingyu Dong, Chong Xia, Mingyuan Jia, Weichen Lyu, Long Xu, Zheng Zhu, and Yueqi Duan. Replicateanyscene: Zero-shot video-to-3d composition via textual-visual-spatial alignment, 2026. URL https://arxiv.org/abs/2604.10789.

Eitan Grinspun, Anil N. Hirani, Mathieu Desbrun, and Peter Schröder. Discrete shells. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 62–67. The Eurographics Association, 2003. doi: 10.2312/SCA03/062-067.

Zhening Huang, Xiaoyang Wu, Fangcheng Zhong, Hengshuang Zhao, Matthias Nießner, and Joan Lasenby. Litereality: Graphics-ready 3d scene reconstruction from rgb-d scans, 2025. URL https://arxiv.org/abs/2507.02861.

Zhening Huang, Yueyan Li, Johnathan Chiu, Xiaoyang Lyu, Matt Zhou, Yuxin Yao, Joan Lasenby, and Shangzhe Wu. LiteReality-Agent: An agentic system for interactable 3d indoor scene reconstruction. Blog post, 2026. URL https://github.com/LiteReality/LiteReality-Agent.

Inhee Lee, Sangwon Baik, Sungjoo Kim, Hyeonwoo Kim, Hyunsoo Cha, and Hanbyul Joo. SimuScene: Simulation-ready compositional 3D scene reconstruction from a single image. *arXiv preprint arXiv:2606.03994*, 2026. URL https://arxiv.org/abs/2606.03994.

Vincent Leroy, Yohann Cabon, and Jerome Revaud. Grounding image matching in 3d with mast3r, 2024.

Haodong Li, Lulu Shao, Haolin Lu, Yu Fu, Yen-Ru Chen, Seemandhar Jain, and Manmohan Chandraker. ϕ-Scene: Physically grounded image-to-3D scene reconstruction. *arXiv preprint arXiv:2606.21596*, 2026a. URL https://arxiv.org/abs/2606.21596.

Minchen Li, Zachary Ferguson, Teseo Schneider, Timothy Langlois, Denis Zorin, Daniele Panozzo, Chenfanfu Jiang, and Danny M. Kaufman. Incremental potential contact: Intersection- and inversion-free large-deformation dynamics. *ACM Transactions on Graphics*, 39(4), jul 2020. doi: 10.1145/3386569.3392425. URL https://doi.org/10.1145/3386569.3392425.

Minchen Li, Danny M. Kaufman, and Chenfanfu Jiang. Codimensional incremental potential contact. *ACM Transactions on Graphics*, 40(4), 2021. doi: 10.1145/3450626.3459767.

Minchen Li, Chenfanfu Jiang, Zhaofeng Luo, Wenxin Du, Chang Yu, Žiga Kovačič, and Tianyi Xie. *Physics-Based Simulation*. March 2026b. doi: 10.5281/zenodo.20597655. URL https://doi.org/10.5281/zenodo.20597655. Open-source online book. Live version available at https://phys-sim-book.github.io/.

Xiaoyu Li, Qi Zhang, Di Kang, Weihao Cheng, Yiming Gao, Jingbo Zhang, Zhihao Liang, Jing Liao, Yan-Pei Cao, and Ying Shan. Advances in 3d generation: A survey. *arXiv preprint arXiv:2401.17807*, 2024.

Xuan Li, Minchen Li, and Chenfanfu Jiang. Energetically consistent inelasticity for optimization time integration. *ACM Transactions on Graphics*, 41(4), 2022. doi: 10.1145/3528223.3530072.

Guying Lin, Kemeng Huang, Michael Liu, Ruihan Gao, Hanke Chen, Lyuhao Chen, Beijia Lu, Taku Komura, Yuan Liu, Jun-Yan Zhu, and Minchen Li. PAT3D: Physics-augmented text-to-3D scene generation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=iIRxFkeCuY.

Liu Liu, Xinjie Wang, Jiaxiong Qiu, Tianwei Lin, Xiaolin Zhou, and Zhizhong Su. Gaussian object carver: Object-compositional gaussian splatting with surfaces completion. *arXiv preprint arXiv:2412.02075*, 2024.

Qi Lu, Jason Kuen, Shen Tiancheng, Gu Jiuxiang, Guo Weidong, Jia Jiaya, Lin Zhe, and Yang Ming-Hsuan. High-quality entity segmentation. In *ICCV*, 2023.

Zhaofeng Luo, Zhitong Cui, Shijian Luo, Mengyu Chu, and Minchen Li. Vr-doh: hands-on 3d modeling in virtual reality. *ACM Transactions on Graphics (TOG)*, 44(4):1–12, 2025.

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

Yuqi Meng, Yihao Shi, Kemeng Huang, Zixuan Lu, Ning Guo, Taku Komura, Yin Yang, and Minchen Li. Efficient b-spline finite elements for cloth simulation. *ACM Transactions on Graphics*, 45(4), 2026. doi: 10.1145/3811278.

Lorenzo Mur-Labadia, Matthew Muckley, Amir Bar, Mido Assran, Koustuv Sinha, Mike Rabbat, Yann LeCun, Nicolas Ballas, and Adrien Bardes. V-JEPA 2.1: Unlocking dense features in video self-supervised learning, 2026. URL https://arxiv.org/abs/2603.14482.

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. RoboCasa: Large-scale simulation of household tasks for generalist robots. In *Proceedings of Robotics: Science and Systems*, 2024. doi: 10.15607/RSS.2024.XX.050. URL https://www.roboticsproceedings.org/rss20/p050.html.

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. *arXiv preprint arXiv:2404.16666*, 2024.

Junfeng Ni, Yu Liu, Ruijie Lu, Zirui Zhou, Song-Chun Zhu, Yixin Chen, and Siyuan Huang. Decompositional neural scene reconstruction with generative diffusion prior. *arXiv preprint arXiv:2503.14830*, 2025.

OpenAI. GPT-6 Astra: A new generation of intelligence, 2026. URL https://openai.com/index/gpt-6-astra/. Accessed: 2026-09-11.

Nicholas Pfaff, Thomas Cohn, Sergey Zakharov, Rick Cory, and Russ Tedrake. SceneSmith: Agentic generation of simulation-ready indoor scenes. *arXiv preprint arXiv:2602.09153*, 2026. URL https://arxiv.org/abs/2602.09153.

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

SAM-3D-Team, Xingyu Chen, Fu-Jen Chu, Pierre Gleize, Kevin J Liang, Alexander Sax, Hao Tang, Weiyao Wang, Michelle Guo, Thibaut Hardin, Xiang Li, Aohan Lin, Jiawei Liu, Ziqi Ma, Anushka Sagar, Bowen Song, Xiaodong Wang, Jianing Yang, Bowen Zhang, Piotr Dollár, Georgia Gkioxari, Matt Feiszli, and Jitendra Malik. Sam 3d: 3dfy anything in images. *arXiv preprint arXiv:2511.16624*, 2025. URL https://arxiv.org/abs/2511.16624.

Yukai Shi, Weiyu Li, Zihao Wang, Hongyang Li, Xingyu Chen, Ping Tan, and Lei Zhang. SceneMaker: Open-set 3D scene generation with decoupled de-occlusion and pose estimation model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27146–27156, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Shi_SceneMaker_Open-set_3D_Scene_Generation_with_Decoupled_De-occlusion_and_Pose_CVPR_2026_paper.html.

Yawar Siddiqui, Duncan Frost, Samir Aroudj, Armen Avetisyan, Henry Howard-Jenkins, Daniel DeTone, Pierre Moulon, Qirui Wu, Zhengqin Li, Julian Straub, Richard Newcombe, and Jakob Engel. Shaper: Robust conditional 3d shape generation from casual captures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Alexey Stomakhin, Craig Schroeder, Lawrence Chai, Joseph Teran, and Andrew Selle. A material point method for snow simulation. *ACM Transactions on Graphics*, 32(4), 2013. doi: 10.1145/2461912.2461948.

Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

Yolo Y. Tang, Daiki Shimada, Jiayue Meng, Jing Bi, Pinxin Liu, Yicheng Wang, Yunzhong Xiao, Zhangyun Tan, Zeliang Zhang, Chao Huang, Susan Liang, Qianxiang Shen, Luchuan Song, Ali Vosoughi, Mingqian Feng, Melika Filvantorkaman, and Chenliang Xu. BVB: Benchmarking agentic video understanding via programmatic reconstruction in blender. *arXiv preprint arXiv:2609.15478*, 2026. URL https://arxiv.org/abs/2609.15478.

Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, 2012. doi: 10.1109/IROS.2012.6386109.

Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380, 1991. doi: 10.1109/34.88573.

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. Vggt-ω, 2026a. URL https://arxiv.org/abs/2605.15195.

Ziqian Wang, Yonghao He, Licheng Yang, Wei Zou, Hongxuan Ma, Liu Liu, Wei Sui, Yuxin Guo, and Hu Su. TabletopGen: Tabletop scene generation and interactive simulation for robotic manipulation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2026b. URL https://arxiv.org/abs/2512.01204.

Xinyue Wei, Minghua Liu, Zhan Ling, and Hao Su. Approximate convex decomposition for 3D meshes with collision-aware concavity and tree search. *ACM Transactions on Graphics*, 41(4), 2022. doi: 10.1145/3528223.3530103.

Qianyi Wu, Kaisiyuan Wang, Kejie Li, Jianmin Zheng, and Jianfei Cai. Objectsdf++: Improved object-compositional neural implicit surfaces. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 21764–21774, 2023.

Qirui Wu, Yawar Siddiqui, Duncan Frost, Samir Aroudj, Armen Avetisyan, Richard Newcombe, Angel X. Chang, Jakob Engel, and Henry Howard-Jenkins. Jrm: Joint reconstruction model for multiple objects without alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Chong Xia, Kai Zhu, Zizhuo Wang, Fangfu Liu, Zhizheng Zhang, and Yueqi Duan. Simrecon: Simready compositional scene reconstruction from real videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 42452–42463, June 2026a.

Hongchi Xia, Chih-Hao Lin, Hao-Yu Hsu, Quentin Leboutet, Katelyn Gao, Michael Paulitsch, Benjamin Ummenhofer, and Shenlong Wang. Holoscene: Simulation-ready interactive 3d worlds from a single video. *Advances in Neural Information Processing Systems*, 38:32501–32524, 2025.

Hongchi Xia, Tianhang Cheng, Wei-Chiu Ma, and Shenlong Wang. FIRE3D: Feed-forward interactive 3d scene reconstruction within a minute. *arXiv preprint arXiv:2609.08848*, 2026b. URL https://arxiv.org/pdf/2609.08848.

Hongchi Xia, Xuan Li, Zhaoshuo Li, Qianli Ma, Jiashu Xu, Ming-Yu Liu, Yin Cui, Tsung-Yi Lin, Wei-Chiu Ma, Shenlong Wang, Shuran Song, and Fangyin Wei. SAGE: Scalable agentic 3D scene generation for embodied AI. *arXiv preprint arXiv:2602.10116*, 2026c. URL https://arxiv.org/abs/2602.10116.

Long Xu, Zhiqi Zhang, Mi Yan, Shengliang Deng, Chong Xia, Mingyu Dong, Jiayi Chen, Jiangran Lyu, Fei Gao, Zhizheng Zhang, et al. Gif: Agentic generation of interactive and functional object compositions for robot learning. *arXiv preprint arXiv:2609.05927*, 2026a.

Ruijie Xu, Xinnan Zhu, Jiayu Ying, Daoguo Dong, Yuzhou Ji, and Xin Tan. MUSE: Agentic 3D scene authoring via memory-grounded incremental requirement satisfaction. *arXiv preprint arXiv:2606.14168*, 2026b. URL https://arxiv.org/abs/2606.14168.

Mi Yan, Jiazhao Zhang, Yan Zhu, and He Wang. Maskclustering: View consensus based mask graph clustering for open-vocabulary 3d instance segmentation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28274–28284. IEEE, 2024.

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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7771–7781, 2025.

Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12–22. IEEE, 2023.

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. *arXiv preprint arXiv:2601.11109*, 2026. URL https://arxiv.org/abs/2601.11109.

Huangyue Yu, Baoxiong Jia, Yixin Chen, Yandan Yang, Puhao Li, Rongpeng Su, Jiaxin Li, Qing Li, Wei Liang, Zhu Song-Chun, Tengyu Liu, and Siyuan Huang. Metascenes: Towards automated replica creation for real-world 3d scans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Hongxin Zhang, Chunru Lin, Junyan Li, Zhou Xian, Tsun-Hsuan Wang, and Chuang Gan. GS-Agent: Creating 4d physical worlds with generative simulation. *arXiv preprint arXiv:2607.21522*, 2026.

Kaifeng Zhang, Shuo Sha, Hanxiao Jiang, Matthew Loper, Hyunjong Song, Guangyan Cai, Zhuo Xu, Xiaochen Hu, Changxi Zheng, and Yunzhu Li. Real-to-sim robot policy evaluation with gaussian splatting simulation of soft-body interactions. *arXiv preprint arXiv:2511.04665*, 2025.

A Additional Related Work: Deformable and Codimensional Simulation

Thin structures and codimensional mechanics. Physics-based graphics has long treated thin structures with reduced-dimensional models rather than resolving their full thickness volumetrically; see Li et al. (2026b) for a broader modern overview. Classical cloth simulation represents fabric as a triangulated surface with in-plane and bending response (Baraff & Witkin, 1998), while discrete-shell models formulate bending directly on triangle meshes and can represent sharp folds and changes in rest curvature (Grinspun et al., 2003). Discrete elastic rods similarly reduce slender bodies to centerlines with bending and twisting energies (Bergou et al., 2008). These representations avoid the high through-thickness resolution and locking issues that can arise when very thin structures are treated as ordinary low-order 3D solids (Bischoff et al., 2004). Recent work continues to improve thin-structure discretization, for example with smooth B-spline finite elements for cloth (Meng et al., 2026). These methods make clear that the geometry required by a simulator depends on the object's effective dimension: rods need centerlines and radii, shells need midsurfaces and thicknesses, and solids need volumetric domains.

Contact across dimensions. Thin structures also require robust contact handling. Classical cloth work developed collision and friction treatment for surface meshes (Bridson et al., 2002); IPC later introduced intersection- and inversion-free variational contact (Li et al., 2020), and C-IPC extended it to mixed-dimensional particles, rods, shells, and solids (Li et al., 2021). Our backend uses this class of contact methods, while our contribution is the reconstruction of solver-compatible assets rather than a new contact formulation.

Elastoplastic and inelastic behavior. Many real objects do not return to their original rest state after manipulation. Plasticity has therefore been modeled across several simulation paradigms. Discrete shells can encode permanent folds by changing rest dihedral angles (Grinspun et al., 2003); elastoplastic constitutive laws have also been central to volumetric and particle methods, for example in snow simulation (Stomakhin et al., 2013). Energetically Consistent Inelasticity formulates finite-strain elastoplasticity and viscoelasticity for optimization-based FEM and MPM time integration (Li et al., 2022). These works provide increasingly general tools for irreversible deformation; CoDimRecon uses this modeling capacity when a behavioral test requires it, as in the paper-folding example where elastic bending cannot retain a crease.

Relation to simulation-ready reconstruction. The works above start from a prescribed rest geometry, discretization, constitutive model, and usually material parameters. In contrast, existing simulation-ready scene reconstruction has largely focused on rigid geometry and rigid-body physics. CoDimRecon bridges these areas by reconstructing representation-specific geometry for deformable curves, surfaces, and volumes, initializing compatible physical models from simulator examples, and revising them when behavioral tests expose a mismatch. Because static RGB observations do not identify true dynamic material parameters, the resulting values are treated as effective simulation parameters rather than measured material properties.

B METHOD DETAILS

Fig. 8 expands the reconstruction stage of Sec. 3.1.

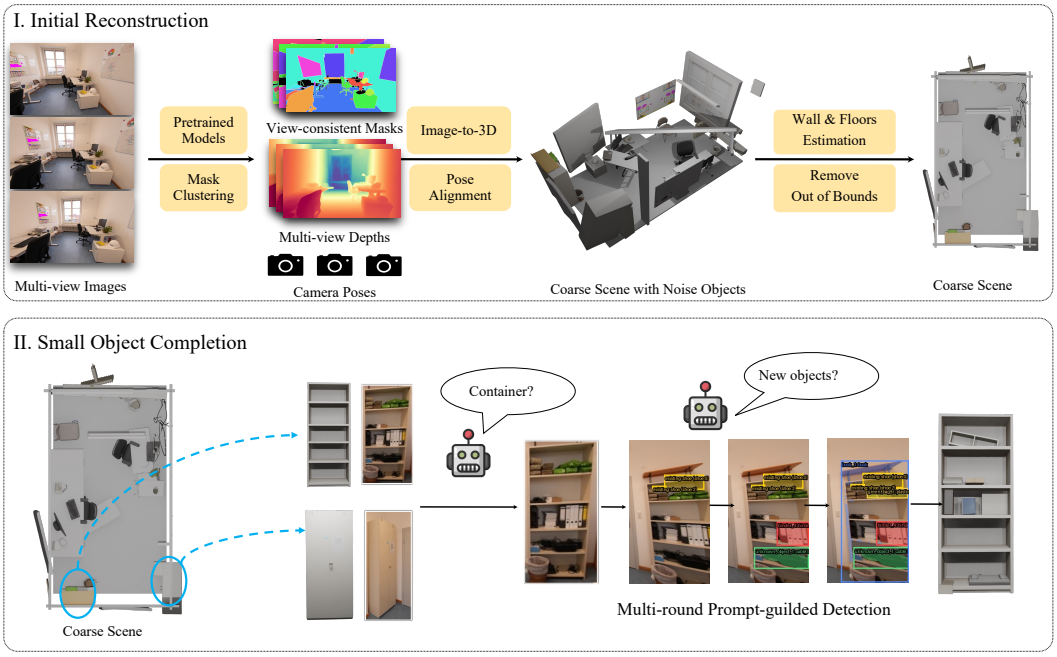


Figure 8: **Initial reconstruction with geometric and generative references.** VGGT-Omega supplies cameras, depth, and point maps, while CropFormer masks are clustered into 3D tracks. SAM3D generates and aligns a reference mesh from a selected view of each track; the agent then rebuilds the object with editable Blender primitives.

B.1 GEOMETRY REFERENCE DETAILS

Our reference generation and alignment follow Dong et al. (2026), with the details below.

Generation. For object track o, let $\mathcal{K}_o$ contain the views with valid mask M_v^o. Rather than choose the largest 2D mask, which can favor a close-up showing only part of the object, we select the view with the greatest lifted surface area. For each $v \in \mathcal{K}_o$, we lift valid mask pixels with the VGGT-Omega point map, form a local triangular surface, and compute

$$A_v^o = \sum_{\tau \in \mathcal{F}(M_v^o)} \mathrm{Area}(\tau), \qquad v^\star = \arg\max_{v \in \mathcal{K}_o} A_v^o, \tag{1}$$

where $\mathcal{F}(M_v^o)$ denotes valid triangles induced by the lifted mask. This criterion favors views exposing more 3D surface and is less sensitive to camera distance than raw mask area.

We crop the RGB image, mask, and point map around $M_{v^\star}^o$ and condition SAM3D (SAM-3D-Team et al., 2025). The selected camera maps the generated canonical mesh back to scene coordinates,

yielding an initial similarity transform $\Theta^{(0)} = \{s^{(0)}, \mathbf{R}^{(0)}, \mathbf{t}^{(0)}\}$. We use this single-view estimate only to initialize multi-view alignment.

Pose Alignment. Around the selected view, we form $\mathcal{V}_o = \{v^\star - r, \ldots, v^\star + r\}$ using only frames with a valid tracked mask. At iteration i, the mesh under $\Theta^{(i-1)}$ is rendered into each $v \in \mathcal{V}_o$, producing RGB $\widehat{I}_v^{(i)}$, depth $\widehat{D}_v^{(i)}$, and mask $\widehat{M}_v^{(i)}$. MASt3R (Leroy et al., 2024) provides dense correspondences between the masked observation and rendering,

$$\mathcal{C}_v^{(i)} = \{(\mathbf{p}_j, \mathbf{q}_j)\}_j = \Phi\Big(I_v \odot M_v^o, \widehat{I}_v^{(i)}\Big), \tag{2}$$

where $\mathbf{p}_j$ and $\mathbf{q}_j$ are matched pixels in the observed and rendered images.

Let $\Pi_v^{-1}(\mathbf{p}, d)$ back-project pixel $\mathbf{p}$ with depth d using the VGGT-Omega intrinsics and camera-to-world pose. Each 2D match yields a 3D pair,

$$\mathbf{x}_j = \Pi_v^{-1}(\mathbf{p}_j, D_v(\mathbf{p}_j)), \qquad \mathbf{y}_j^{(i)} = \Pi_v^{-1}\Big(\mathbf{q}_j, \widehat{D}_v^{(i)}(\mathbf{q}_j)\Big). \tag{3}$$

Aggregating pairs over $\mathcal{V}_o$, we solve the incremental scale, rotation, and translation with Umeyama alignment (Umeyama, 1991):

$$\Delta\Theta^{(i)} = \arg\min_{\Delta s, \Delta\mathbf{R}, \Delta\mathbf{t}} \sum_{(\mathbf{x},\mathbf{y})\in\mathcal{P}^{(i)}} \|\mathbf{x} - (\Delta s\, \Delta\mathbf{R}\mathbf{y} + \Delta\mathbf{t})\|_2^2, \qquad \Theta^{(i)} = \Delta\Theta^{(i)} \circ \Theta^{(i-1)}. \tag{4}$$

We repeat render–match–align for K iterations and retain the transform with the highest mean rendered-to-tracked mask IoU:

$$i^\star = \arg\max_{i\in\{1,\ldots,K\}} \frac{1}{|\mathcal{V}_o|} \sum_{v\in\mathcal{V}_o} \mathrm{IoU}\Big(\widehat{M}_v^{(i)}, M_v^o\Big), \qquad \Theta^\star = \Theta^{(i^\star)}. \tag{5}$$

B.2 Primitive Representation Compactness

We quantify the compaction of Sec. 3.1 on one office chair. The SAM3D reference has 178,172 vertices and 356,084 triangles; the agent reconstruction uses 43 primitives (22 cuboids, 18 cylinders, and 3 frustums), 2,192 vertices, and 1,183 polygonal faces. This is an $81\times$ vertex and $301\times$ face reduction, and the authoring file drops from 22.9 MiB to 610 KiB after removing the reference mesh. For geometric agreement, 20,000 area-weighted reference samples have one-sided fitted-surface distances of 0.59% of object height on average and 1.53% at the 95th percentile; silhouette IoU is 81.1–93.7% over five orthographic views. This single-object measurement is against the generated reference, not the real chair.

B.3 Small Object Completion

Scene-level segmentation can omit small objects inside containers. In a completion pass, the agent identifies reconstructed containers, inspects the corresponding image regions, and obtains masks for visible contents. REST3D (Ma et al., 2026) then generates each missing object with SAM3D, while the container provides context for placing it back into the scene. This is part of reference-scene construction (Sec. 3.1), not a separately evaluated component.

B.4 Deformable Simulation Settings

These settings accompany Tab. 2. All scenes use gravity $\mathbf{g} = (0, 0, -9.81)\ \mathrm{m/s^2}$; the telephone cord, chair cushion, and beanbag use $dt = 0.01$ s. For scenes with many deformables, rest-shape settling processes bodies in order of decreasing volume until its runtime budget is reached.

Trial scene construction. Because IPC requires penetration-free input, each robot trial uses only the geometry needed for the interaction: (1) an inserted fixed-base Franka with seven revolute arm joints and two prismatic finger joints, modeled with affine body dynamics; (2) the target deformable and attached rigid bodies, such as the telephone handset and base; and (3) the remaining scene merged into a fixed collision environment.

Telephone cord. The cord uses QIPC's native discrete elastic rod with twist (Bergou et al., 2008), initialized from the reconstructed centerline and radius. The reconstructed coil is the stress-free

natural shape; Bishop-frame directors initialize the rod, and twist evolves dynamically. Stretching, bending, and twisting moduli are all $40\,\mathrm{MPa}$. Together with the density and $1.3\,\mathrm{mm}$ radius in Tab. 2, this gives a cord mass of about $14\,\mathrm{g}$. These values come from an existing simulator phone example as a visual-plausibility prior and are not calibrated to the real cord.

The reconstructed centerline contains $3{,}200$ nodes over $1.694\,\mathrm{m}$ and produces $15{,}755$ overlapping non-adjacent segment-capsule pairs, with up to $2.48\,\mathrm{mm}$ overlap. To satisfy IPC's penetration-free input requirement, we fit a 576-segment polyline whose segments all exceed the rod diameter by at least $1\,\mu\mathrm{m}$. Segment lengths are 2.60–$3.49\,\mathrm{mm}$ (mean $2.89\,\mathrm{mm}$), total discrete length is $1.664\,\mathrm{m}$, and the maximum deviation from the reconstructed curve is $0.422\,\mathrm{mm}$.

At each endpoint, the outermost three edges form a strain-relief lead attached to the handset or base: position and tangent are fixed in the body frame, while axial twist remains free. The handset and base are dynamic affine bodies of 0.2 and $0.55\,\mathrm{kg}$. Contact uses friction 0.6, activation distance $0.15\,\mathrm{mm}$, resistance 10^8, and global continuous collision detection. Each step permits up to 100 Newton iterations (velocity tolerance $2 \times 10^{-3}\,\mathrm{m/s}$), $4{,}000$ PCG iterations (tolerance rate 10^{-6}), and 24 line-search iterations. We add no artificial damping or velocity modification.

Each $14\,\mathrm{s}$ rollout stores $1{,}401$ states. The phone assembly starts $10\,\mathrm{mm}$ above the scene and settles for $8\,\mathrm{s}$. The gripper approaches from 8.6 to $9.6\,\mathrm{s}$, closes by $10.3\,\mathrm{s}$, lifts the handset through finger contact from 10.7 to $12.8\,\mathrm{s}$, and holds until $14\,\mathrm{s}$; independent finger PD controllers are limited to $20\,\mathrm{N}$. A run passes if all segment stretches remain in $[0.9, 1.1]$; settling drift and maximum nodal speed over 7.5–$8\,\mathrm{s}$ stay below $1\,\mathrm{mm}$ and $2\,\mathrm{mm/s}$; the handset rises and remains at least $120\,\mathrm{mm}$ above its start, with hold slip below $5\,\mathrm{mm}$ and finger–surface gap below $1\,\mathrm{mm}$; rigid-body affine strain stays below 10^{-3}; and robot joint residuals stay below $0.1\,\mathrm{mm}$.

Both repeated runs pass without a failed state. Settling drift and speed are $0.773\,\mathrm{mm}$ and $1.6\,\mathrm{mm/s}$; peak handset lifts are 197.8 and $197.7\,\mathrm{mm}$. During lift, the minimum non-adjacent cord gap is $22.0\,\mu\mathrm{m}$ in both runs, the minimum cord–environment gaps are 0.126 and $0.110\,\mathrm{mm}$, and endpoint error remains below $1.4 \times 10^{-14}\,\mathrm{m}$. During hold, maximum finger-relative slip is 1.79 and $1.77\,\mathrm{mm}$ and the maximum finger–surface gap is $0.143\,\mathrm{mm}$. No step exceeds 26 Newton iterations. Segment stretch remains within $[0.9985, 1.0463]$ and $[0.9983, 1.0465]$.

The maximum stretch of about 1.046 appears in the first step and remains nearly constant thereafter; lifting does not increase it. Resampling leaves 127 next-but-one capsule pairs within the contact activation distance, the closest at $3.5\,\mu\mathrm{m}$. At the first step, the barrier separates these pairs and lengthens 88 short segments from 2.60–$2.71\,\mathrm{mm}$ to 2.72–$2.74\,\mathrm{mm}$, just below $2r + \delta = 2.75\,\mathrm{mm}$. Across the two repeated plans, peak lift, final handset height, settling drift, and settling speed remain within the declared repeat tolerances of $3\,\mathrm{mm}$, $3\,\mathrm{mm}$, $0.2\,\mathrm{mm}$, and $0.5\,\mathrm{mm/s}$; the peak lifts differ by $0.085\,\mathrm{mm}$. The cord trajectories themselves differ by $0.665\,\mathrm{mm}$ RMS and up to $9.55\,\mathrm{mm}$ locally. The settled shape is written back to Blender as the rest shape (Sec. 3.3); the natural shape remains the reconstructed coil, so the stored rest shape is an equilibrium under gravity rather than a stress-free configuration.

Chair cushion. The seat and back cushions use StVK–Hencky solids for large compression; no viscous foam model is used. Their underside and rear surfaces are bonded to a fixed frame, with contact elsewhere. A $70\,\mathrm{mm}$ rounded tool performs two tasks under one material model: a $1\,\mathrm{s}$ gravity release and a $10\,\mathrm{s}$ three-point press with $25\,\mathrm{mm}$ commanded indentation. Both require principal stretches in $[0.1, 3.0]$, minimum element Jacobian 0.04, and at most $3\,\mathrm{mm}$ displacement under gravity and after recovery; the press also requires 18–$45\,\mathrm{mm}$ peak indentation. Final indentations are 22.9, 22.6, and $22.9\,\mathrm{mm}$.

Beanbag settling. The beanbag uses stable Neo-Hookean solid finite elements with Tab. 2's parameters. It is released under gravity without pinned vertices, added supports, or artificial damping; the surrounding scene is fixed and contact resistance is 10^5. QIPC (v0.0.1.dev895) runs 600 steps over $6\,\mathrm{s}$ with $dt = 0.01\,\mathrm{s}$ and stores all 601 states. Each step allows up to 100 Newton iterations (velocity tolerance $0.005\,\mathrm{m/s}$), $1{,}000$ linear-solver iterations, and 20 line-search iterations. Fig. 9 shows convergence. Relative to equilibrium, the initial geometry differs by at most 1.1% of bounding-box diagonal D and 0.29% on average. Maximum vertex distance stays below $10^{-3}D$ after $1.15\,\mathrm{s}$ and $10^{-4}D$ after $2.26\,\mathrm{s}$. Finite-difference kinetic energy peaks at $0.26\,\mathrm{J}$ and falls to $8.5 \times 10^{-11}\,\mathrm{J}$ by $6\,\mathrm{s}$. No tetrahedron inverts: minimum element Jacobian remains above 0.91 and volume changes by less than 0.75%.

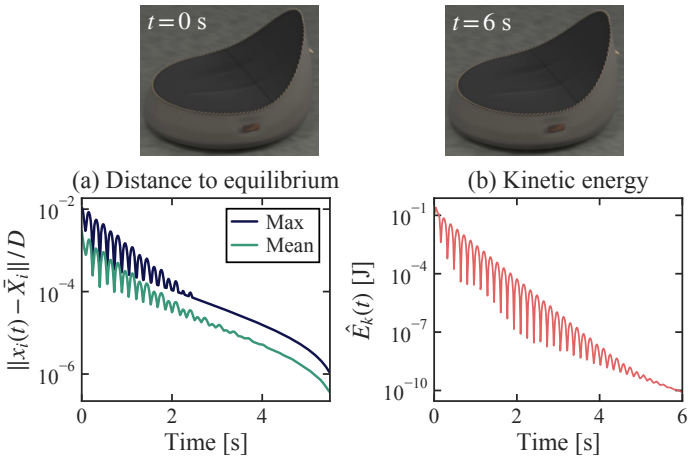


Figure 9: **Rest-shape settling.** Reconstructed beanbag at $t = 0$ and 6 s. (a) Vertex distance to equilibrium, normalized by bounding-box diagonal $D = 1.75$ m; equilibrium is averaged over $t \in [5.5, 6]$ s. (b) Finite-difference kinetic energy. Maximum distance remains below $10^{-4}D$ after 2.3 s.

Paper. Paper uses an isotropic St. Venant–Kirchhoff membrane with hinge bending: density $761.9\ \mathrm{kg/m^3}$ (areal density $80\ \mathrm{g/m^2}$), Young's modulus $2.25\ \mathrm{GPa}$ for membrane and bending, Poisson's ratio 0.15, thickness $0.105\ \mathrm{mm}$, contact activation distance $0.5\ \mathrm{mm}$, and friction 0.4. QIPC's plastic variant uses yield curvature $25\ \mathrm{m^{-1}}$ with no hardening, updating a yielded hinge's natural angle irreversibly. The elastic control disables bending plasticity and keeps initial natural angles; membrane plasticity is disabled in both.

The gripper executes a $140°$ motion about a prescribed fold axis and pivot; this is the gripper motion, not a paper constraint. Both rollouts store $1{,}051$ states at 0.01 s intervals, with opening from 7.2 to 7.7 s. Fold angle is measured between area-weighted normals of two fixed material regions, with $0°$ denoting parallel normals; the initial $5.35°$ offset is not subtracted. Both rollouts initialize without detected intersections and pass strain and joint checks. The plastic run changes the natural angle of 372 hinges by more than 0.1 rad; the elastic run changes none. The elastic control was rerun after enabling plastic bending to isolate that model change, rather than using the original failed elastic trial. Fig. 5 and Tab. 4 report the release behavior and fold angles.

Plastic bag. The plastic bag in Fig. 1 is a discrete thin shell with St. Venant–Kirchhoff in-plane elasticity and hinge bending: density $920\ \mathrm{kg/m^3}$, Young's modulus $60\ \mathrm{MPa}$ for membrane and bending, Poisson's ratio 0.45, thickness $0.03\ \mathrm{mm}$, and friction 0.45. The bag deforms freely, without pinned or bound vertices, and the gripper holds it through contact alone. The trash bin is a separate fixed affine body with density $700\ \mathrm{kg/m^3}$ and rigidity penalty 10^9; the Franka uses rigidity penalty 10^8. These values are simulation settings and are not calibrated to the real bag.

Behavioral-test revisions. Tab. 6 summarizes revisions triggered while testing the three assets in Fig. 4.

C Evaluation Details

C.1 Geometry and Rendering

Alignment. Our method, HoloScene, and ReplicateAnyScene share VGGT-Omega cameras. We match predicted and ground-truth cameras by frame, initialize $\mathbf{x}_{\mathrm{GT}} = s\mathbf{R}\mathbf{x}_{\mathrm{pred}} + \mathbf{t}$ from camera centers with Umeyama alignment (Umeyama, 1991), compute camera RMSE and rotation error, and refine the transform with scene-level ICP. GPT-6 Astra authors each scene in an independent coordinate frame with estimated scale and hand-placed cameras. We therefore register it by a global Z-up search over yaw in $5°$ increments, with and without mirroring; initialize scale from horizontal room extents and translation from floor height and horizontal center; retain the candidate with the

Asset	Observed failure	Attributed to	Revision	Material changed
Telephone cord	No collision-free path in the first two plans	Motion	Re-planned trajectory	No
Cushion, indenter study	Element inversion and excess strain under large compression	Material model	Hencky logarithmic-strain elasticity	Yes
Cushion, press study	Element inversion and excess strain during the press	Motion	Revised press motion	No
Chair cushion	Back fragments fell under gravity; the solve did not converge	Geometry, boundary	Volume re-classified; full seat underside bonded	No
Chair cushion	Excess stretch and shallow indentation at the third point	Tool	70 mm rounded tool	No
Chair cushion	Excess stretch at the third point	Contact location	Third point moved to the cushion front	No
Paper	Crease not kept after release	Material model	Plastic hinge bending ($\kappa_Y = 25\,\mathrm{m}^{-1}$)	Yes
Paper	Plan check failed for the first two plans	Motion	Re-planned trajectory	No

Table 6: **Revisions during agent-guided behavioral testing.** Rows are chronological within each asset. Earlier cushion studies supply the Hencky model and $E = 30\,\mathrm{kPa}$ used in the final cushion; setup failures, repeats, interrupted runs, and rendering are omitted.

highest F-score at $10\ \mathrm{cm}$; and run coarse-to-fine ICP with scale at 0.3, 0.15, 0.08, and $0.05\ \mathrm{m}$. Each transform is scene-level and applied to the full predicted mesh; no per-object alignment is used. Ground-truth meshes and cameras come from the HoloScene release.

Geometry. We sample $500{,}000$ surface-area-weighted points from predicted mesh P and ground-truth mesh Q (seed 0). Chamfer distance is the unsquared symmetric mean nearest-neighbor distance,

$$\mathrm{CD} = \frac{1}{2}\Big(\frac{1}{|P|}\sum_{\mathbf{p}\in P}\min_{\mathbf{q}\in Q}\|\mathbf{p}-\mathbf{q}\|_2 + \frac{1}{|Q|}\sum_{\mathbf{q}\in Q}\min_{\mathbf{p}\in P}\|\mathbf{q}-\mathbf{p}\|_2\Big), \tag{6}$$

and is reported in centimeters. Precision (Prec) and recall (Rec) are the fractions of predicted and ground-truth points within $5\ \mathrm{cm}$ of the other set, with $\mathrm{F1} = 2\,\mathrm{Prec}\cdot\mathrm{Rec}/(\mathrm{Prec}+\mathrm{Rec})$ reported in percent. Normal consistency (NC) is the mean absolute cosine similarity between the surface normal at each sampled point and that at its nearest neighbor in the other set, averaged over both directions and reported in percent. Evaluation includes walls, floors, ceilings, and other background surfaces: all render-enabled Blender surfaces are used for the prediction and the full mesh for ground truth, without foreground filtering or visibility culling.

Rendering. We render every input frame: 811, 371, and 731 views for the three Replica scenes, and 186, 380, and 528 views for ScanNet++ scenes `67d702f2e8`, `7831862f02`, and `acd69a1746`. Frames are rendered at the input resolution (512×512 for Replica and 876×584 for ScanNet++) with Cycles (24 samples, seed 0), using the corresponding cameras in Blender coordinates. PSNR, SSIM, and LPIPS (AlexNet features) are computed on full unmasked images and averaged equally over frames. Because every method reconstructs from these same views, these metrics measure observation fidelity rather than held-out novel-view synthesis. Layout and Motion use 64 uniformly sampled render/reference frame pairs (Appendix C.3).

C.2 RIGID-BODY STABILITY

We use a MuJoCo rigid-body drop test as a physical-stability proxy. For evaluation only, GPT-6 Astra assigns each reconstructed object to the same semantic categories for every method: walls, floors, and ceilings are *background*; wall- or ceiling-mounted fixtures (e.g., paintings, lights, doors, windows, and attached blinds) are *static*; and freestanding objects (e.g., furniture, books, plants, cushions, rugs, and bedding) are *dynamic*. Background and static geometry remain fixed, while each dynamic object is tested individually under gravity.

Background. A standard convex decomposition would fill an enclosing room shell. We instead cluster triangles by face-normal direction (grouping angle $< 20^\circ$), bisect each cluster until deviation from a fitted plane is below $3\ \mathrm{cm}$, and extrude each patch into a $5\ \mathrm{cm}$-thick convex slab, preserving the hollow interior.

Dynamic objects. Each dynamic object is decimated to at most $20{,}000$ faces and decomposed into at most 32 CoACD (Wei et al., 2022) convex hulls (threshold 0.05).

Per-object simulation. For each test, all other objects and static geometry are fixed. Simulation runs for $3\ \mathrm{s}$ with $dt = 0.002\ \mathrm{s}$, gravity $-9.81\ \mathrm{m/s^2}$, uniform density $300\ \mathrm{kg/m^3}$, friction 1.0, and an infinite ground plane at the estimated floor height. If a static background slab initially penetrates the test object by more than $2\ \mathrm{cm}$, we remove that slab and restart, for at most three retries; this avoids artificial ejection from reconstruction overlaps such as a door embedded in its frame.

Fall criterion. An object counts as fallen if, after settling, either its local z-axis tilts by more than 30° or its displacement exceeds $\max(5\ \mathrm{cm},\ 0.2d)$, where d is its bounding-box diagonal.

Scores. Let N be the number of dynamic objects, N_g the number of *ground objects* whose lowest point lies within $5\ \mathrm{cm}$ of the floor, and n_f and $n_f^{(g)}$ the numbers of fallen objects in the two sets.

$$\text{Stable(All)} = 1 - \frac{n_f}{N}, \qquad \text{Stable(Ground)} = 1 - \frac{n_f^{(g)}}{N_g}. \tag{7}$$

Stable(All) also includes supported objects such as items on desks or shelves, so it depends on the reconstructed supports.

C.3 LATENT SIMILARITY

We use BVB's Latent Similarity (LS) (Tang et al., 2026), comparing source and reconstructed videos in a frozen vision encoder's feature space.

Frame sampling. We uniformly sample 64 one-to-one frame pairs from the input and rendered sequences, which share the same camera trajectory.

Encoding. Both clips pass through frozen V-JEPA 2.1 (Mur-Labadia et al., 2026) ViT-G (`bf16`). Last-layer tokens are arranged on a $T \times H \times W$ grid, with each temporal step grouping two consecutive frames.

Motion. We average all tokens into one vector per clip; Motion is the cosine similarity between these vectors.

Layout. For Layout, tokens are averaged over time into a spatial feature map, and we average per-patch cosine similarity between the two maps. If tokens cannot form a rectangular spatial grid, Layout reduces to Motion.

LS. The combined score is $\text{LS} = (\text{Layout} + \text{Motion})/2$.

Remark. LS is affected by rendering style: our method uses fully lit Cycles renders, whereas ReplicateAnyScene uses unshaded vertex-colored meshes. We therefore treat LS as a complementary feature-space similarity measure, not as reconstruction quality alone or a direct test of downstream scene understanding.

D REIMPLEMENTATIONS

We preserve each baseline's native pipeline and supply only required external inputs. HoloScene receives instance masks, cameras, and depth from our pipeline (Appendix D.2); ReplicateAnyScene and GPT-6 Astra start from RGB, with ReplicateAnyScene using its official VLM + SAM3 segmentation for the quantitative comparison (Appendix D.1).

D.1 REPLICATEANYSCENE

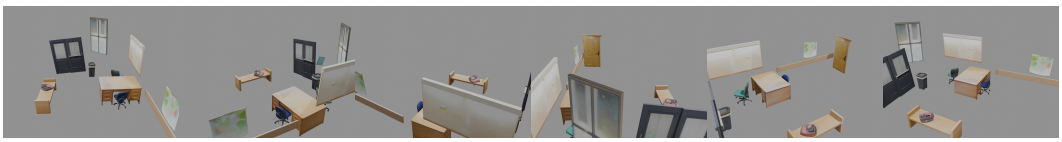
The official ReplicateAnyScene release omits the pose-alignment and VLM relation-reasoning modules described in the paper. We reimplement both and tune the reimplementation per scene.

Object segmentation. ReplicateAnyScene prompts a VLM for an object inventory from sampled frames, uses those names as SAM3 (Carion et al., 2025) text prompts, and propagates masks with SAM3-Video. The quantitative comparison in Tab. 3 uses this default segmentation pipeline. The qualitative comparisons (Fig. 3 and Appendix G) instead show ReplicateAnyScene results obtained with ground-truth instance masks.

Verification on the official example. As a sanity check, we run the full reimplementation on the hallway scene distributed with the official release. Fig. 10 shows the input frames and resulting object reconstruction.



(a) Input images.



(b) Reconstruction results (VLM + SAM3 masks).

Figure 10: **ReplicateAnyScene reimplementation on the official hallway example.** (a) Six sampled input frames. (b) Reconstruction with default VLM + SAM3 segmentation; the room shell is hidden to expose individual objects.

D.2 HOLOSCENE

We use the official HoloScene implementation without modification. HoloScene receives our mask-clustering instance masks (Sec. 3.1) and the same VGGT-Omega camera poses and per-frame depth as CoDimRecon, controlling for these external inputs in the comparison.

E PROMPTS

Listing 1 gives the GPT-6 Astra baseline prompt from Sec. 4. It is issued once per scene at reasoning effort `xhigh`. Only two strings vary: `<FRAME_DIR>` points to the scene's RGB frames and `<OUTPUT_DIR>` to its output directory. All other prompt text is identical across scenes and is provided verbatim.

Listing 1: Prompt used for the GPT-6 Astra scene-reconstruction baseline.

```
You are given a directory containing hundreds of RGB frames extracted from a video:

`<FRAME_DIR>`

You can only take the hundreds of RGB frames extracted from a video as input, do not take
any other files as reference.
```

```
Your task is to autonomously reconstruct the **complete visible indoor scene** shown
throughout the video as an explicit, object-decomposed, editable 3D scene in Blender.

This is a **full-scene reconstruction task**, not a single-object reconstruction task. Do
not select only the most salient object, such as a desk, and stop after reconstructing it.
You must recover the room structure and all visually significant, sufficiently observable
objects across the video.

You are running on macOS. Blender is already installed. Use Blender through the command line
and the Blender Python API. A typical Blender executable location is:

`/Applications/Blender.app/Contents/MacOS/Blender`

If necessary, locate the installed Blender executable yourself.

Use the following output directory:

`<OUTPUT_DIR>`

Create it if it does not exist.

Complete the entire task autonomously. Do not stop at planning, frame inspection, or the
first modeling attempt.

## 1. Inspect the complete frame sequence

Inspect the frames in:

`<FRAME_DIR>`

The frames are temporally sampled from a video, so neighboring frames may be highly
redundant. Efficiently inspect the complete sequence using scripts, image metadata,
thumbnails, contact sheets, clustering, or other appropriate local tools.

Your inspection must cover the entire video timeline. Do not inspect only the beginning of
the sequence or only frames containing one dominant object.

Determine:

* the overall room or hallway layout;
* which areas of the environment are visited by the camera;
* the major structural elements;
* all distinct, visually significant objects;
* repeated instances of the same object category;
* approximate spatial relationships among objects;
* which regions are well observed and which remain ambiguous;
* whether the video contains multiple connected areas or viewpoints of the same area.

Use image evidence rather than filenames alone.

## 2. Build a scene inventory before modeling

Before constructing the Blender scene, create:

`<OUTPUT_DIR>/scene_inventory.md`

The inventory must list every structural element and distinct physical object that should be
reconstructed.

Include, where applicable:

* floor;
* walls;
* ceiling;
* doors and door frames;
* windows;
* openings;
* columns;
* baseboards and built-in architectural structures;
* desks and tables;
* chairs;
* cabinets and shelves;
* equipment and fixtures;
* lamps;
* screens, computers, keyboards, telephones, and similar devices;
* objects placed on furniture;
* cables or other visually meaningful thin structures;
* repeated object instances;
* any other significant objects visible in the scene.

For each item, record:
```

```
* a unique semantic name;
* object category;
* approximate scene location;
* the frames providing visual evidence;
* estimated geometric form;
* whether it is a unique object or a repeated instance;
* planned modeling method;
* reconstruction status;
* uncertainty or occlusion.

Do not collapse multiple distinct physical objects into one inventory entry merely because
they share the same category.

The inventory must be created before detailed modeling and updated as reconstruction
progresses.

## 3. Select primary reference views

Select at most 16 frames that collectively provide the most useful evidence for
reconstructing the **entire scene**.

Save the exact absolute paths of these frames to:

`<OUTPUT_DIR>/selected_frames.txt`

Choose frames that jointly provide:

* coverage of different regions of the environment;
* diverse camera viewpoints;
* broad geometric coverage;
* complementary visibility of different surfaces;
* evidence for the room layout;
* evidence for object count and placement;
* clear and sharp observations;
* minimal occlusion;
* useful views of geometrically ambiguous regions.

Avoid near-duplicate neighboring frames.

Do not select all primary frames around only one object. The primary set must represent the
complete visible environment as broadly as possible.

These frames should remain the primary reference set throughout reconstruction and
refinement. Do not replace them merely to optimize the result after observing reconstruction
errors.

If an object or geometric region remains ambiguous, occluded, or absent from the primary
reference set, inspect additional frames as supplementary evidence.

Record every supplementary frame used, together with the reason it was needed, in:

`<OUTPUT_DIR>/supplementary_frames.txt`

There is no strict limit on supplementary frames, but avoid unnecessary redundant inspection.

## 4. Infer a coherent global scene layout

Before detailed object modeling, infer a consistent global coordinate system and approximate
scene layout.

Establish:

* a consistent up direction;
* a ground plane;
* approximate global scale;
* room or hallway boundaries;
* major architectural dimensions;
* camera-traversed regions;
* approximate placement, orientation, and size of all major objects;
* support relationships, such as objects standing on the floor or resting on desks;
* repeated structures and repeated object spacing.

If exact metric scale cannot be recovered, choose a reasonable real-world scale based on
familiar objects such as doors, desks, chairs, or computers, and document the assumption.

The final scene must be globally coherent. Do not model each reference frame as an unrelated
local arrangement.

## 5. Reconstruct the complete scene in Blender
```

```
Reconstruct the environment as an explicit, editable, object-decomposed Blender scene.

The reconstruction is not complete merely because the most prominent desk or another
dominant object has been modeled.

Reconstruct all visually significant and sufficiently observable elements identified in the
scene inventory, including the architectural shell, major furniture, equipment, fixtures,
and meaningful smaller objects.

### Object decomposition

Each distinct physical object should be represented as a separate, clearly named Blender
object or logically organized collection whenever practical.

Use semantic names such as:

* `wall_left`;
* `door_01`;
* `desk_01`;
* `desk_02`;
* `chair_01`;
* `monitor_01`;
* `telephone_01`.

Do not merge the complete environment into a single undifferentiated mesh.

Organize the Blender scene into logical collections, for example:

* `Architecture`;
* `Furniture`;
* `Equipment`;
* `Small_Objects`;
* `Cables`;
* `Cameras`;
* `Lights`.

Repeated objects may share linked mesh data when appropriate, but each physical instance
must have its own transform and identifiable object name.

### Geometry requirements

The final reconstruction should prioritize:

1. completeness of the visible scene;
2. correct global room layout and approximate scale;
3. correct object count;
4. correct object placement and orientation;
5. correct overall geometry and proportions;
6. correct spatial, support, and containment relationships;
7. correct major surface structure;
8. reasonable finer details supported by the images;
9. clean, editable, semantically organized Blender geometry.

Use real 3D geometry for important shapes. Do not rely on flat image planes, camera-facing
billboards, or view-dependent geometry that only appears correct from one viewpoint.

The reconstructed scene must remain spatially plausible when viewed from novel viewpoints.

Avoid intersections between separate physical objects unless the images clearly support
contact or overlap. Internal intersections between components belonging to the same
assembled object are acceptable when reasonable.

### Materials

Assign reasonable materials and colors based on the visual evidence.

Exact photorealistic texture recovery is secondary to geometry, completeness, proportions,
and layout. However, different semantic objects and visibly different surfaces should not
all use one generic material.

Use procedural or simple image-based materials where appropriate. Keep materials editable
and clearly named.

## 6. Use an iterative reconstruction workflow

Do not stop after generating the first scene.

Perform repeated reconstruction and verification cycles:

1. construct or modify the Blender scene;
2. render the current reconstruction from useful viewpoints;
```

```
3. inspect the renders;
4. compare them with the selected reference frames;
5. identify missing objects and structural elements;
6. identify incorrect object counts;
7. identify incorrect geometry, proportions, placement, or orientation;
8. correct the detected problems;
9. render and inspect again.

Every iteration must evaluate both:

* the accuracy of objects already reconstructed; and
* the completeness of the full scene.

Do not spend all refinement iterations polishing one desk or another individual object while
other significant scene elements remain absent.

First obtain a complete coarse reconstruction of the entire scene. Then refine the geometry
of individual objects according to visual importance and available evidence.

## 7. Render verification views

Create Blender cameras that provide useful views of the reconstructed scene.

Where camera poses cannot be recovered exactly, create approximate comparison views that
show similar visible regions and viewing directions.

Produce:

* several scene overview renders;
* renders covering different regions of the environment;
* close-up renders of important or ambiguous objects;
* novel-view renders that help verify that the geometry is coherent in 3D.

Save verification renders under:

`<OUTPUT_DIR>/renders`

Do not rely only on Blender viewport screenshots. Render actual images through Blender.

After rendering, inspect the generated images yourself. Do not assume that a successfully
executed render means the reconstruction is visually correct.

## 8. Perform a mandatory completeness audit

Before finishing, systematically compare the reconstructed Blender scene against:

* the full scene inventory;
* all selected primary frames;
* any supplementary frames used.

For every inventory item, assign one of the following final statuses:

1. `Reconstructed`;
2. `Partially reconstructed`;
3. `Not reconstructable from available evidence`;
4. `Intentionally omitted`.

For any item not fully reconstructed, document the specific reason. "Not a main object" is
not a valid reason for omission.

Explicitly check for:

* missing structural elements;
* missing furniture;
* missing repeated instances;
* incorrect object counts;
* objects modeled in the wrong location;
* implausible floating objects;
* unintended intersections;
* inconsistent global scale;
* important objects represented only as vague placeholders;
* areas of the video that are not represented in the Blender scene.

A scene containing only the desk or another dominant object must be treated as incomplete
unless the complete video genuinely contains no other reconstructable scene elements.

If significant omissions are detected, return to modeling and correct them before finishing.

## 9. Save the final deliverables

Save the final editable Blender scene to:
```

```
`<OUTPUT_DIR>/final_scene.blend`

Also save the Blender Python scripts used for reconstruction and rendering under:

`<OUTPUT_DIR>/scripts`

Save a final report to:

`<OUTPUT_DIR>/reconstruction_report.md`

The report must include:

* a summary of the reconstructed environment;
* the selected primary frames;
* supplementary frames consulted and why;
* the inferred global layout and scale assumptions;
* a complete list of reconstructed objects;
* the final status of every scene-inventory item;
* important geometric or material assumptions;
* remaining ambiguities;
* intentionally omitted elements and exact reasons;
* verification renders produced;
* the refinement iterations performed;
* the absolute path to the final `.blend` file.

## 10. Completion criteria

Do not declare the task complete until all of the following are true:

* the full video frame sequence has been inspected;
* the scene inventory has been created;
* the primary reference frames have been selected;
* the global scene layout has been established;
* the architectural structure has been reconstructed;
* all significant and sufficiently observable objects have been reconstructed;
* repeated objects are represented with the correct approximate instance count;
* the scene has been rendered from multiple useful viewpoints;
* the rendered results have been visually inspected;
* at least one substantive refinement pass has been completed after the initial full-scene
model;
* a final completeness audit has been performed;
* the scene inventory and reconstruction report have been updated;
* the final editable Blender file has been saved successfully.

If some geometry remains ambiguous after consulting the available frames, make the most
reasonable inference, model a plausible approximation, and document the uncertainty.

Work autonomously from start to finish. Continue inspecting, inventorying, modeling,
rendering, comparing, and refining until you have produced the strongest complete-scene
reconstruction possible under these constraints.
```

F Supplementary Demos

The supplementary material includes a self-contained web page that shows the reconstructed scenes and the deformable-object demos of Sec. 4; models, Draco decoders, previews, and videos are loaded from local relative paths. To view it, download and extract the supplementary material, serve the extracted folder with any local HTTP server, and open `index.html` through that server. For example, run

```
python3 -m http.server 8000
```

in the extracted folder and visit `http://localhost:8000`. `VIDEO_INDEX.txt` lists every included demonstration video.

G Additional Visualizations

Figs. 11–15 extend the qualitative comparison of Fig. 3 to the remaining ScanNet++ and Replica scenes, showing rendered appearance (top) and geometry (bottom) for each method from the same input view.

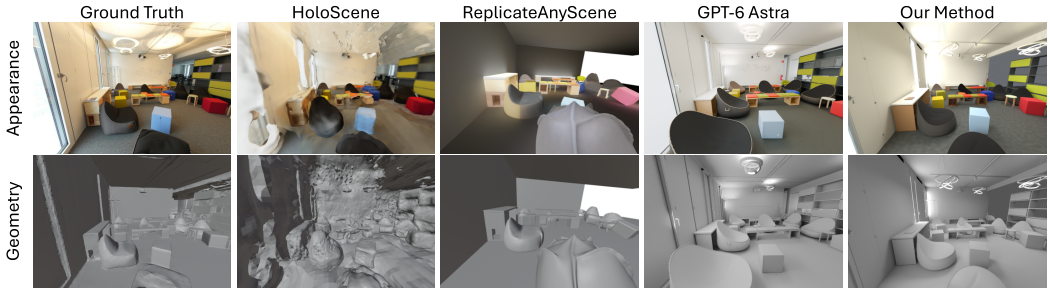


Figure 11: Qualitative comparison on ScanNet++ scene `7831862f02`.

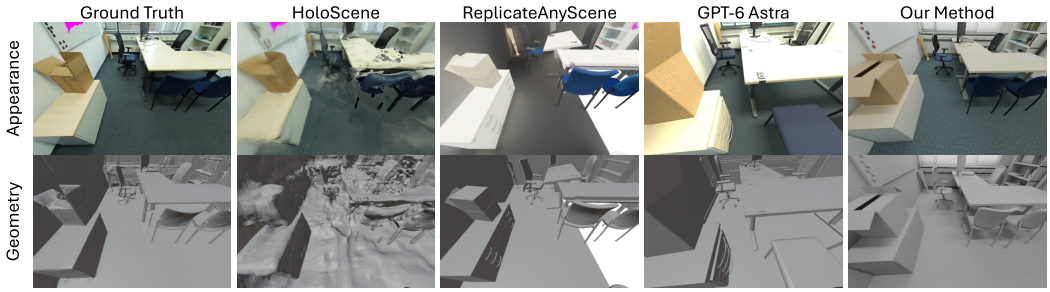


Figure 12: Qualitative comparison on ScanNet++ scene `acd69a1746`.



Figure 13: Qualitative comparison on Replica `room_0`.

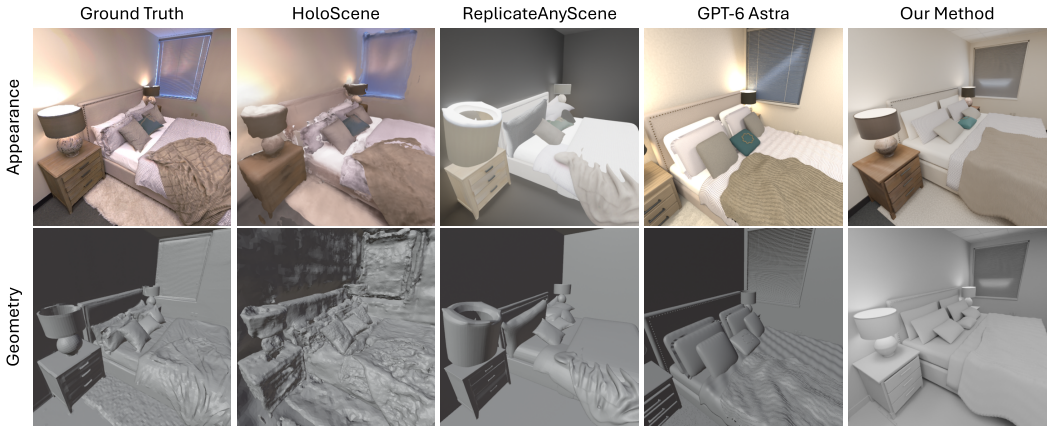


Figure 14: Qualitative comparison on Replica `room_1`.

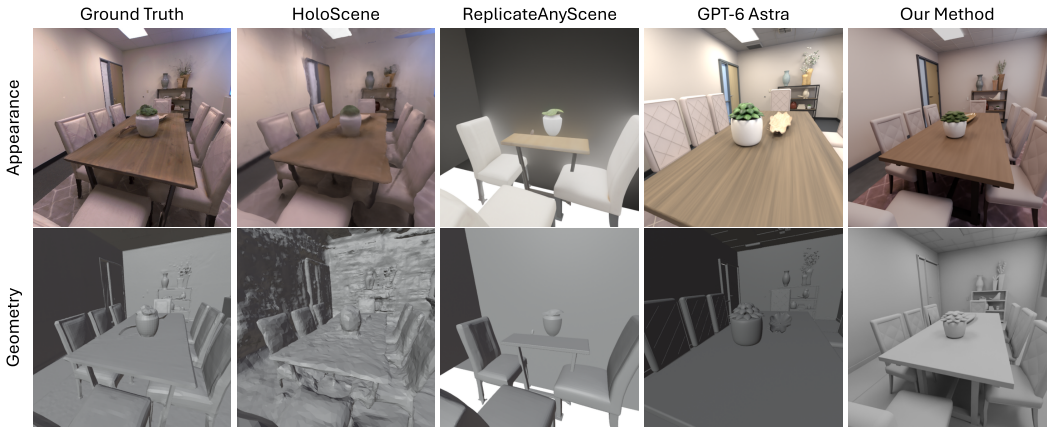


Figure 15: Qualitative comparison on Replica `room_2`.