UNRAVEL: RECOVERING AN INTERACTIVE 3D SCENE FROM A SINGLE IMAGE

Anonymous authors
Paper under double-blind review

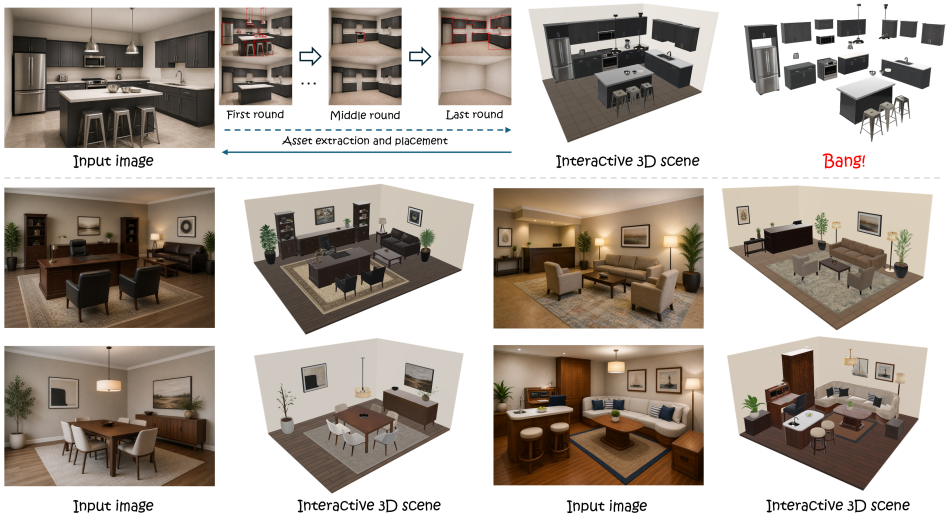

Figure 1: Unravel turns a single image into an interactive 3D scene. In the top row, objects are removed from the input image layer by layer until the room is empty, and the scene is then rebuilt with every object as an independent asset (exploded view, right). The rows below show more input images and their recovered interactive scenes.

ABSTRACT

We present Unravel, a training-free framework that turns a single image into an interactive 3D scene, where every object is an independent and movable asset. Existing generators either produce one entangled geometry in which no object can be moved, or predict all objects in one feed-forward pass, where a wrong guess cannot be detected or repaired. Unravel instead takes a different approach: it first takes the scene apart, and then puts it back together. Starting from the input image, an agentic harness that iteratively peels the image layer by layer, and adopts point cloud as major modality for spatial relationship recovery. Specifically, a vision-language model plans which objects are fully visible, an image editor removes them and restores the occluded regions. Every edit is verified before being incorporated into the reconstruction. This process gradually turns occluded parts of the scene into direct observation, allowing objects to be reconstructed when they are best exposed rather than hallucinated from the original image alone. The scene is then rebuilt in reverse removal order, so that the supporting objects are naturally placed before the objects they support. Since a single-view image only reveals part of an object, we treat the observed points as a lower bound on its size: an asset may grow to cover its points, but can not shrink away from them. Finally, we physically settle the reconstructed scene to remove floating objects and interpenetrations, followed by an agentic audit that corrects remaining placement errors. Extensive experiments on diverse indoor scenes show that Unravel recovers more objects, places them more faithfully, and produces scenes that are more physically-plausible than existing methods.

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.

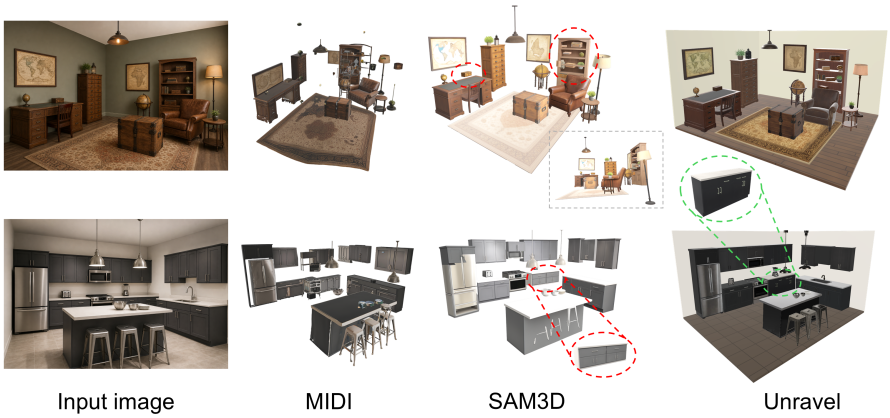

Figure 2: Single-image interactive scene reconstruction. One-shot generators such as MIDI (Huang et al., 2025) and SAM 3D (Chen et al., 2026) recover independent assets, but under occlusion they misplace objects and distort the geometry of hidden parts (red highlights). Unravel recovers occluded objects with faithful geometry and placement (green highlights).

1 Introduction

Embodied agents act on the world around them. They move objects, open containers, and rearrange furniture. Training them in real spaces is too expensive, so they are trained in interactive 3D simulated scenes, where every object can be moved, replaced, or re-arranged (Yang et al., 2024c; Xia et al., 2026). Building these simulated scenes by hand is costly as well, and recovering them from single images offers an appealing alternative. A photograph is cheap to collect, but what we recover from it should go beyond appearance. The scene must stay physically meaningful when its objects are moved.

Many reconstruction methods recover scene geometry without explicit object decomposition (Wang et al., 2024; 2025a; Kong et al., 2026). They reconstruct visible surfaces but leave occluded regions incomplete and do not directly produce movable assets. Object-centric generators instead predict separate meshes and poses from a cluttered image (Huang et al., 2025; Meng et al., 2026; Chen et al., 2026). However, learning 3D priors requires costly data and training. Under heavy occlusion, errors in shape completion can distort individual assets and propagate into the layout (Fig. 2). Because object recovery and placement are coupled, these errors are difficult to isolate and correct.

We present Unravel, a training-free framework that deconstructs a scene before recomposing it (Fig. 1). At each round, a vision-language model selects exposed objects for removal. An image editor removes them and completes the freed regions, and a verifier checks the edit before it contributes to reconstruction (Fig. 4). Occluded content is therefore completed in image space, where intermediate predictions can be inspected and corrected. Each object is captured immediately before removal and converted from its best-observed snapshot into an independent 3D asset.

Iterative removal can provide cleaner inputs for object reconstruction. Our key insight is to use the deconstruction record as a geometric reference for recomposition. We align and fuse the per-round point maps while retaining each point's source round and pixel. Under a fixed camera, source rounds distinguish successive surface observations along the same image ray. Instance masks associate these observations with assets, producing point clusters that anchor their placement. Reversing the removal order also guides assembly from supporting objects to the objects they carry.

Recomposition must respect the incompleteness of these clusters. Unravel registers each asset under one-sided extent constraints on under-observed axes. The observed spans provide lower bounds rather than forcing a complete asset to match a partial surface. Physical settling reduces floating and interpenetration while limiting displacement from the geometric anchors. A final agentic audit inspects scene renderings and refines remaining placement errors. Ablations show improved asset fidelity over direct reconstruction. Qualitative comparisons with feed-forward and agentic baselines show more faithful layouts. Our contributions are threefold.

- We introduce Unravel, a training-free framework for reconstructing interactive 3D scenes from a single image. Verified layer-wise deconstruction generates independently movable assets from their best-observed snapshots.
- We develop a recomposition strategy that reuses the deconstruction record. Round-and-pixel provenance anchors each asset to its instance cluster, and reverse removal order guides support-aware assembly. The strategy combines one-sided extent constraints with physical settling.
- Extensive experiments across diverse indoor scenes demonstrate that Unravel can faithfully reconstruct interactive 3D scenes from a single image.

2 Related Work

2.1 Feed-forward 3D Reconstruction

Feed-forward models offer an alternative to per-scene optimization, including in pose-free settings. DUSt3R (Wang et al., 2024) regresses aligned pointmaps from uncalibrated image pairs. Its successors support robust matching (MASt3R (Leroy et al., 2024)), streaming reconstruction (Spann3R (Wang & Agapito, 2025), CUT3R Wang et al. (2025b)), and thousand-view input (Fast3R (Yang et al., 2025)). VGGT (Wang et al., 2025a) jointly predicts cameras, depth, pointmaps, and tracks for hundreds of views in one forward pass. For monocular input, Metric3Dv2 (Hu et al., 2024) estimates zero-shot metric depth and surface normals. Depth Anything V2 (Yang et al., 2024a) provides relative depth. The MoGe series progresses from affine-invariant pointmaps (Wang et al., 2025c) to detailed metric geometry (Wang et al., 2026; Kong et al., 2026). These methods recover per-pixel geometry without explicit object decomposition, leaving occluded regions incomplete and objects inseparable.

2.2 3D Object Asset Generation

Image-to-3D models generate textured meshes from clean object images. TripoSR (Tochilkin et al., 2024) directly infers geometry in one forward pass. Multi-view methods (InstantMesh (Xu et al., 2024), LGM (Tang et al., 2024b), Unique3D (Wu et al., 2024)) synthesize views before reconstructing geometry. Native 3D latent models (CLAY (Zhang et al., 2024), TRELLIS (Xiang et al., 2025), Hunyuan3D-2 (Zhao et al., 2025), TripoSG (Li et al., 2025), Direct3D-S2 (Wu et al., 2026)) learn directly from asset corpora. TRELLIS.2 (Xiang et al., 2026) generates high-resolution textured assets, while NI-Tex (Shan et al., 2026a) produces garment textures from non-isometric image inputs. Part-aware models add component-level structure. PartCrafter (Lin et al., 2026) jointly generates part meshes from a single image, and HoloPart (Yang et al., 2026) completes segmented parts. These models provide strong priors for individual asset generation. Extending them to complete scene generation remains challenging.

2.3 Interactive 3D Scene Generation

Object-level scene generation produces independently editable assets rather than fused geometry. Text-driven methods arrange objects through layout diffusion (DiffuScene (Tang et al., 2024a), InstructScene (Lin & Mu, 2024), PhyScene (Yang et al., 2024b)) or LLM-based planning (Holodeck (Yang et al., 2024c)). Agentic systems (SAGE (Xia et al., 2026), SceneSmith (Pfaff et al., 2026), SceneAssistant (Luo et al., 2026)) refine scenes through visual feedback. GPT-6 Astra (OpenAI, 2026) also supports image understanding, editing, and 3D tool use. Yet semantic instructions alone underspecify the geometry of a reference image. ControlRoom3D (Schult et al., 2024) and WonderWorld (Yu et al., 2025) offer layout-conditioned and single-image scene synthesis, respectively, without targeting independent object editing.

Closer to our setting, MIDI (Huang et al., 2025), SceneGen (Meng et al., 2026), SAM 3D (Chen et al., 2026), and CAST (Yao et al., 2025) infer object meshes and poses from cluttered views. Their 3D priors complete occluded regions but require costly data and training. Training-free pipelines such as Interact3D (Shan et al., 2026b) and SimFoundry (Ranawaka et al., 2026) use modular optimization and self-correction. Precise placement remains challenging in crowded scenes. Seeing

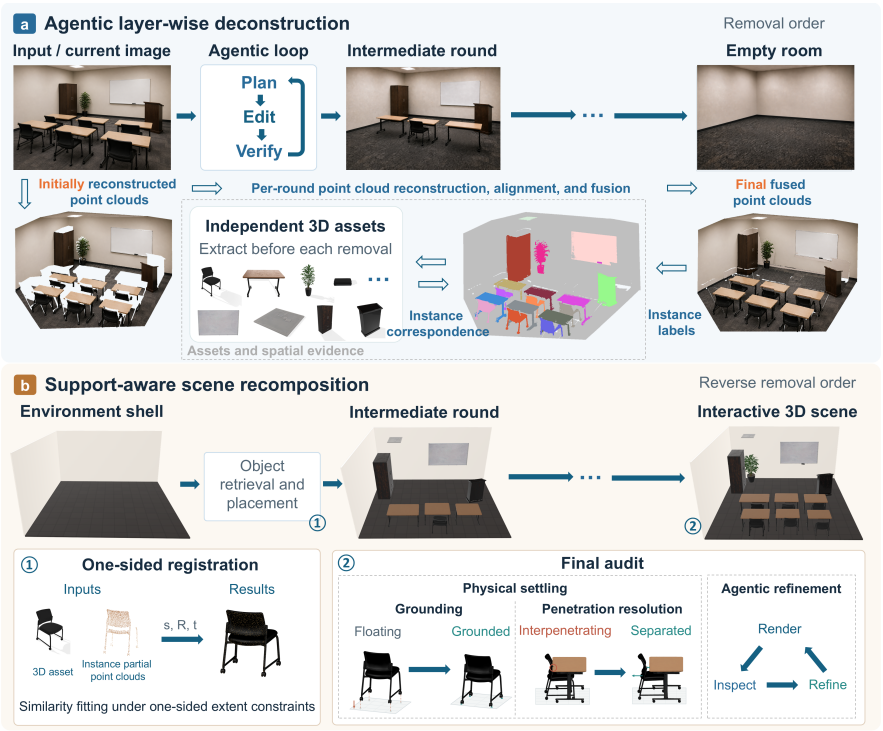

Figure 3: Overview of Unravel. (a) Agentic layer-wise deconstruction. An agentic loop plans, edits, and verifies removals until the room is empty. Independent 3D assets are extracted before removal. Per-round reconstructions are aligned and fused into an instance-labeled point cloud. (b) Support-aware scene recomposition. Assets return to the environment shell in reverse removal order. One-sided extent constraints guide registration to their point clusters. Physical settling and a final audit refine placement.

Through Clutter (Aguina-Kang et al., 2026) iteratively removes objects to obtain cleaner reconstruction inputs. Unravel links verified deconstruction to physically constrained recomposition. Round-and-pixel provenance anchors each asset to its reconstructed geometry, and reverse removal order guides support-aware assembly.

3 Methods

Unravel deconstructs the scene before recomposing it (Fig. 3). The deconstruction stage (Section 3.2) removes objects layer by layer and extracts independent assets from their best-observed snapshots. Per-round reconstructions are aligned and fused into a point cloud that retains each object's location. Recomposition (Section 3.3) places assets in reverse removal order. Each asset is registered to its point cluster, and physical settling refines the assembled scene.

3.1 Problem Formulation

Given a single image I of an indoor scene, our goal is to recover an interactive 3D scene $\mathcal{S}$ consisting of an environment shell and a set of independently placed assets,

$$\mathcal{S} = \big(\mathcal{E},\ \{(a_i, T_i)\}_{i=1}^{N}\big).$$

The shell $\mathcal{E}$ comprises the floor and the walls. Each a_i is an independent 3D asset placed by $T_i = (\mathbf{s}_i, R_i, t_i)$: a rotation $R_i \in SO(3)$, a translation $t_i \in \mathbb{R}^3$, and a scale $\mathbf{s}_i \in \mathbb{R}^3_+$. The scale is resolved axis by axis in the asset's upright frame, since a single view bounds the extent of each axis only from below (Section 3.3). We additionally record which asset or shell surface each object rests on, so that objects can be moved, replaced, or re-arranged, and the scene re-settled into a physically valid state after any edit.



Figure 4: Layer-wise deconstruction and point cloud fusion. The upper block shows planner outputs, removal targets, and edited images across six rounds. Red boxes mark targets, and green boxes mark protected objects. The lower block shows aligned per-round reconstructions and cumulative fused point clouds, starting with the original-image reconstruction. Fusion retains removed-object geometry and adds surfaces reconstructed from edited images.

3.2 Agentic Layer-Wise Deconstruction

We use 2D foundation models to plan, edit, and verify layer-wise removals without task-specific training. Occluded content is inferred through verified image completion before 3D reconstruction. With the camera fixed, we align each reconstruction to the first round and fuse them into a point cloud. This cloud provides the primary geometric reference for recovering spatial relationships.

The Deconstruction Loop The loop uses a VLM planner and verifier with an image editor (Fig. 4, upper block). The planner selects fully visible objects for removal and marks the rest for protection. Their bounding boxes are drawn on a plan image. Walls and the floor are preserved, and contents are removed before containers. Guided by the plan, the editor removes targets from I_k and inpaints the freed regions to produce I_{k+1}. These edits provide more complete object views for later rounds. The verifier checks for missed targets, unintended removals, and fabricated content before an edit is committed. Changed pixels between I_k and the verified I_{k+1} form the edit mask E_k. The loop repeats until only the empty room remains.

We extract each asset immediately before removal from its best-observed snapshot. A segmentation model (Carion et al., 2026) operates inside its bounding box to obtain the instance mask S_i, improving precision over full-image segmentation. Appendix A.1 details box localization. We store the snapshot and semantics. gpt-image-2 (OpenAI, 2026) converts the snapshot to a canonical image, which an image-to-3D model (Xiang et al., 2026) turns into a textured mesh. The asset index records the mesh, mask, semantics, box, and removal round for recomposition.

Recovering Scene Geometry Across Rounds The key challenge in this loop is drift. Each round rebuilds the scene from a different edited image. Without explicit control, errors in scale and pose can misalign surfaces across rounds and distort the fused geometry.

We fix camera intrinsics across rounds. The monocular estimator MoGe3 (Kong et al., 2026) predicts a point map $P_k \in \mathbb{R}^{H\times W\times 3}$ and per-pixel confidence $c_k(p) \in [0,1]$ from I_k. The validity mask $M_k = \{p \mid c_k(p) > \tau\}$ retains pixels above a fixed threshold τ. We also exclude depth discontinuities, where estimates may mix surfaces. Each later reconstruction is aligned directly to the first with a similarity transformation.

We estimate this transformation using only stable pixels, which we call anchors. A pixel serves as an anchor only if its geometry is valid in both the first and the current round and it has never been edited in earlier rounds, so that it depicts the same surface in both rounds. Formally, the anchor set

is $\mathcal{A}_k = \{p \mid M_k(p) \wedge M_1(p) \wedge p \notin \bigcup_{j<k} E_j\}$, and we minimize

$$\min_{s,R,t} \sum_{p \in \mathcal{A}_k} \left\| P_1(p) - \left(sR\,P_k(p) + t\right) \right\|^2.$$

We solve this objective in closed form (Umeyama, 1991). Some anchors nevertheless carry inconsistent geometry across rounds, mostly near editing boundaries and shadow residues. We treat anchors with large residuals after a first fit as outliers, discard them, and refit on the remaining anchors.

The fused cloud accumulates only new information. Unedited pixels keep their existing points. Pixels edited in round $k-1$ contribute the surfaces they now reveal,

$$\mathrm{PC} \leftarrow \mathrm{PC} \cup \left\{ sR\,P_k(p) + t \mid p \in E_{k-1} \cap M_k \right\},$$

where E_{k-1} is the previous edit mask and M_k is the current validity mask. Edit masks select each point's source round, while validity masks filter unreliable estimates. The lower block of Fig. 4 shows how fusion preserves removed-object geometry and adds surfaces reconstructed from completed regions. Alignment places these points in a shared frame for asset placement.

Point Attribution and Instance Clusters Labeling a single-view cloud is normally ambiguous: a pixel admits one label, while the surfaces stacked behind it belong to different objects. Deconstruction dissolves this ambiguity. The camera is locked, so a pixel p is one fixed ray, and every round observes this ray at a different depth. We call the points added to the cloud at round k the layer of round k, with the layer 1 the whole first reconstruction. A pixel now holds one point per layer, and we label points rather than pixels.

For the point x of layer k at pixel p, the candidates are the objects a_i whose instance masks S_i cover p and that are still present in round k, namely, whose removal round satisfies $r_i \geq k$. They queue along the ray in the order of removal, and we label x by the frontmost object. If nothing in the queue survives to round k, the ray shows the room, and we label x by 0, the room. Because a label depends only on its own layer and on the fixed masks, later removals never change labels already assigned. Points labeled i form the cluster C_i of asset a_i, and points labeled 0 form the room structure, from which the environment shell $\mathcal{E}$ is fitted. The removal sequence provides the depth ordering along each ray. Masks come from each object's best-observed image and may include content completed by earlier edits.

3.3 Support-aware Scene Recomposition

Deconstruction records each object's removal round and geometric observations. We use both to recompose the scene (Fig. 3(b)). Removal rounds guide support-aware assembly. Point provenance anchors registration and physical settling.

From Occlusion Order to Support Order The deconstruction planner removes contents before their containers, since a container becomes fully visible only after its contents leave. For a support relation $i \to j$, where object i rests on object j, the removal order therefore satisfies $r_i < r_j$. We assemble the scene in descending removal round, starting from the empty shell. For every support relation $i \to j$, asset j is then placed before asset i. Every object arrives when its support is already in place, so placement always has a physical reference. Large supports from later rounds enter first, and small foreground objects from early rounds enter last.

The fused cloud provides the placement evidence. By the attribution rule of Section 3.2, every point belongs to one asset, and the points of asset a_i form its cluster C_i. The remaining points belong to the room structure, and fitting them gives the environment shell $\mathcal{E}$.

One-Sided Registration under Physical Constraints Each generated asset is complete, while its cluster C_i covers only surfaces reconstructed from the original and edited images. Registration must fit the asset to its cluster without shrinking below what was observed. Let X_i be surface samples from asset a_i. We register them to C_i with a scaled rigid transformation T_i. Candidate transformations are scored by a bidirectional median Chamfer distance (Fan et al., 2017),

$$D_i(T) = \frac{1}{2}\left[\mathrm{med}_{x \in T(X_i)} \min_{y \in C_i} \|x - y\|_2 + \mathrm{med}_{y \in C_i} \min_{x \in T(X_i)} \|y - x\|_2 \right],$$

and the best candidate is refined by truncated similarity ICP. The first term measures asset-to-cluster fit, and the second measures coverage of the cluster. Median distances reduce sensitivity to neighboring-object fragments and spurious generated parts. Truncated ICP similarly retains reliable correspondences at each step to limit distortion from partial observations.

A single view records only the front of an object, so a cluster provides a lower bound on the object's size rather than the exact size. We therefore compare the asset and its cluster along the three axes of the object's upright frame. The height axis points up along gravity. The other two axes lie in the horizontal plane. The depth axis runs front to back, closer to the viewing direction, while the width axis runs across the view. Occlusion hides the object's back, so the depth axis is always under-observed. The width or height axis may also be under-observed when the object is cut off by the image border or largely occluded along that direction. These axes form $\mathcal{U}_i$. Registration solves

$$T_i^* = \arg\min_T D_i(T) \quad \text{s.t.} \quad \text{span}_u\big(T(X_i)\big) \geq \text{span}_u(C_i), \quad \forall u \in \mathcal{U}_i,$$

where $\text{span}_u(\cdot)$ measures how far a point set extends along axis u. In words, along an under-observed axis the asset must reach at least as far as the observed points. Partial observation can therefore grow an asset, but never shrink it.

In practice the transform stays close to a similarity map. It combines a global scale with bounded per-axis adjustments in the object's frame, and never becomes a free affine map. Appendix A.2 gives the exact update rules.

Physical Settling and Final Audit Registration places every asset close to where it was seen, but it does not guarantee a plausible scene. Objects may float above the floor or sink into their neighbors. We correct these defects with small translations. To decide which objects may move, we infer the support graph from semantic relations and local support geometry, and classify each object as mounted or movable. Mounted objects, such as pictures and pendant lights, are attached to walls or the ceiling and stay fixed. Writing the settled asset as $a_i' = T_i(X_i) + \delta_i$, we formulate settling as

$$\min_{\{\delta_i\}_{i\in\mathcal{D}}} \sum_{i\in\mathcal{D}} \|\delta_i\|_2^2 \quad \text{s.t.} \quad \text{supported}(a_i') = 1, \quad \phi(a_i', a_j') \geq 0 \ \ \forall i < j, \quad \delta_i \in \mathcal{B}_i,$$

where $\mathcal{D}$ indexes the movable objects during settling. The objective asks for the smallest total movement, and the constraints state what a plausible scene must satisfy.

Every movable object must be supported, with its bottom resting on the floor or the top of another object. The clearance $\phi(a_i', a_j')$ between two assets turns negative when they overlap, so the second constraint prevents interpenetration. The budget $\mathcal{B}_i$ keeps corrections faithful to the observation, so an asset aligned with its cluster may move only slightly, and a misplaced asset may move toward the cluster but never further away. Appendix A.2 gives the exact definitions. Settling lowers floating objects onto their supports and separates interpenetrating pairs. A final agentic audit renders the settled scene and checks for remaining placement errors. Each detected error triggers a local correction followed by another settling pass. The audit repeats until no further error is detected.

4 Experiments

4.1 Experimental Setup

Models and Evaluation Setup. We compare Unravel with feed-forward object and pose predictors and agentic baselines based on iterative reasoning and tool use. We use GPT-5.5 (OpenAI, 2026) as the vision-language model for per-round planning, semantic localization, and edit verification. The image editor is gpt-image-2 (OpenAI, 2026), which also produces the canonical object images. SAM 3 (Carion et al., 2026) extracts instance masks inside the bounding boxes provided by the planner. MoGe-3 (Kong et al., 2026) reconstructs the per-round point maps, and TRELLIS.2 (Xiang et al., 2026) turns the canonical images into textured assets. GPT-6 Astra (Light) (OpenAI, 2026) performs the final agentic audit on the rendered scene. The agentic baseline uses the stronger Astra (Ultra) variant. Calls to GPT-6 Astra are through the official web interface. Every component is used off the shelf, and none is fine-tuned on our data. The framework is training-free.

Baseline Methods. Among feed-forward baselines, MIDI (Huang et al., 2025) generates object instances through multi-instance diffusion. SAM 3D (Chen et al., 2026) predicts object geometry,

Table 1: Ablation on five scenes. Area ratio and UCR summarize coverage for objects removed after the first round, using Unravel's masks and point clusters as references. UCR (under-coverage rate) denotes the percentage with coverage below 70%. VLM and Human report mean fidelity scores (1 to 10) over all 160 objects. Point counts are scene averages.

	2D masks		3D assets ↑		Scene geometry	
Method	Area ratio ↑	UCR (%) ↓	VLM	Human	Points (M) ↑	UCR (%) ↓
Direct reconstruction	0.772	11.3	4.7	4.9	1.53	10.1
Unravel	**1.000**	**0.0**	**8.2**	**8.6**	**2.81**	**0.0**

texture, and layout from the input image. Both directly recover objects and their arrangement, so we evaluate recovery and placement together. For the agentic comparison, we select GPT-6 Astra (Ultra) (OpenAI, 2026) for its strong visual understanding and reasoning. We evaluate two settings that differ in mesh source. In the direct setting, Astra generates meshes and their arrangement from the input image. In the shared-asset setting, it receives our 3D assets and 2D segmentation masks (Section 3.2), along with the input image. It estimates only each asset's scale and pose. We denote this variant GPT-6 Astra (Ultra) + TRELLIS2 (Fig. 5). Astra generates coarser meshes than image-to-3D models. Sharing assets therefore isolates placement from mesh quality. This recomposition ablation compares Astra's agentic strategy with our point-cluster registration and physical settling. We also tested Seeing Through Clutter (Aguina-Kang et al., 2026). The evaluated pipeline lacks an explicit edit-verification harness. Our tests on complex scenes showed substantial hallucinations and unstable editing. Figure 7 in Appendix A.4 presents one comparison.

4.2 Qualitative Comparisons

Figure 5 compares the four baselines on multiple scenes, each shown from two views. MIDI relies on priors learned from limited 3D data, so heavily occluded objects often suffer severe generation errors. SAM 3D produces reasonable assets in most cases, but its textures tend to drift. Its asset quality falls short of TRELLIS.2, a dedicated image-to-3D model, and its training is expensive. Its placement estimates are likewise imprecise, which easily introduces floating and interpenetration. The direct Astra variant captures broad spatial relations, yet its self-generated meshes are visibly coarser. Sharing our assets removes this gap, yet the red boxes still reveal placement discrepancies. Since the meshes are shared, these errors come from placement rather than asset quality. Across these scenes, Unravel preserves the input layout most closely. Appendix A.4 reports more reconstructions with per-round edits, extracted assets, and multi-view renderings. Figure 11 shows reconstructions from six real indoor images.

4.3 Ablation Study

To evaluate the deconstruction loop, we compare Unravel with direct reconstruction on five scenes containing 160 objects. Both use SAM 3 for segmentation, TRELLIS.2 for asset generation, and MoGe3 for point cloud reconstruction. Direct reconstruction uses only the input image. Unravel extracts each object's mask when it is ready for removal and fuses the point clouds from successive rounds. We compare the resulting 2D masks, 3D assets, and scene geometry in Table 1.

For 2D masks, we report each object's direct reconstruction mask area as a percentage of its Unravel counterpart. Objects removed after the first round account for roughly one third of the test set. Their mean area ratio is 77.2%, with means of 85.0% for round 2 and 60.4% for round 4. Across all objects, 5.6% have ratios below 50%, with a minimum of 23.8%. Larger masks give TRELLIS.2 more visual detail for asset generation. To assess the resulting 3D assets, a vision-language model and human raters score each mesh against its reference image. Scores range from 1 to 10, with higher values indicating greater fidelity. Appendix A.3 details the protocol and examples. For scene geometry, fusing four to eight rounds increases the mean point count per scene from 1.53 million to 2.81 million. We measure the fraction of points in Unravel's fused object clusters covered by direct reconstruction. Mean coverage is 76.9% for objects removed after the first round. Across evaluated objects, 10.1% have coverage below 70% and 3.8% below 50%. The minimum is 11.9%. These measurements quantify the additional geometry available for asset placement after deconstruction.



Figure 5: Qualitative comparison of single-image scene reconstruction. Each input is followed by two views of each reconstructed scene. The two Astra variants separate direct scene generation from placement with shared assets. GPT-6 Astra (Ultra) + TRELLIS2 receives the same generated meshes and corresponding 2D instance masks used by Unravel. Red boxes highlight placement discrepancies in the agentic results.

5 Limitations and Future Work

Unravel is limited by current image editing models. In highly cluttered scenes, edits may alter the appearance or position of remaining objects. Such drift can accumulate across rounds and disrupt point-cloud alignment and asset placement. Our verification improves editing reliability, but minor errors may persist. Future work could combine stronger image editing models with geometric feedback to better preserve object identity and spatial relationships across rounds. Uncertainty-aware checks could trigger targeted retries before small errors accumulate.

AI USE STATEMENT

We used generative AI tools to generate the synthetic test data, and we also used them as components of our experimental agentic workflow. Specifically, GPT-Image-2 generated input images used in testing, while GPT-5.5, GPT-Image-2, and GPT-6 Astra served within the agentic workflow. The roles of these models and the procedures for their use are explicitly described in the paper. Additionally, we used generative AI tools to polish text originally written by the authors, improving its grammar, clarity, and readability. We have not used generative AI tools for any other task requiring disclosure. We have reviewed all AI-assisted work. Specifically, we checked the polished text for accuracy and consistency with the intended meaning, inspected the AI-generated test images for their suitability for the experimental tasks, and reviewed the outputs of the agentic workflow and the associated experimental results. We take full responsibility for the final content of this paper, including all text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Rio Aguina-Kang, Kevin James Blackburn-Matzen, Thibault Groueix, Vladimir Kim, and Matheus Gadelha. Seeing through clutter structured 3d scene reconstruction via iterative object removal. In *2026 International Conference on 3D Vision (3DV)*, pp. 630–639. IEEE, 2026.

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

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

Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 605–613, 2017.

Mu Hu, Wei Yin, Chi Zhang, Zhipeng Cai, Xiaoxiao Long, Hao Chen, Kaixuan Wang, Gang Yu, Chunhua Shen, and Shaojie Shen. Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):10579–10596, 2024.

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

Lingyu Kong, Ruicheng Li, Ruicheng Wang, Sicheng Xu, Chengtang Yao, Jianfeng Xiang, and Jiaolong Yang. Moge-3: Fine-detail monocular geometry estimation with self-guided sparse volumetric refinement. *arXiv e-prints*, pp. arXiv–2607, 2026.

Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European conference on computer vision*, pp. 71–91. Springer, 2024.

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

Chenguo Lin and Yadong Mu. Instructscene: Instruction-driven 3d indoor scene synthesis with semantic graph prior. In *International Conference on Learning Representations*, volume 2024, pp. 25687–25718, 2024.

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

Jun Luo, Jiaxiang Tang, Ruijie Lu, and Gang Zeng. Sceneassistant: A visual feedback agent for open-vocabulary 3d scene generation. *arXiv preprint arXiv:2603.12238*, 2026.

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

OpenAI. GPT-5.5, GPT-Image-2, and GPT-6 Astra model documentation. `https://developers.openai.com/api/docs/models/all`, 2026. Accessed: 2026-09-18.

Nicholas Pfaff, Thomas Cohn, Sergey Zakharov, Rick Cory, and Russ Tedrake. Scenesmith: Agentic generation of simulation-ready indoor scenes. In *Forty-third International Conference on Machine Learning*, 2026.

Nadun Ranawaka, Josiah Wong, Wei-Lin Pai, Wei-Teng Chu, Tianyuan Dai, Masoud Moghani, Hang Yin, Yunfan Jiang, Wesley Durbano, Brandon Huynh, et al. Simfoundry: Modular and automated scene generation for policy learning and evaluation. *arXiv preprint arXiv:2606.28276*, 2026.

Jonas Schult, Sam Tsai, Lukas Höllein, Bichen Wu, Jialiang Wang, Chih-Yao Ma, Kunpeng Li, Xiaofang Wang, Felix Wimbauer, Zijian He, et al. Controlroom3d: Room generation using semantic proxy rooms. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6201–6210. IEEE, 2024.

Hui Shan, Ming Li, Haitao Yang, Kai Zheng, Sizhe Zheng, Yanwei Fu, and Xiangru Huang. Nitex: Non-isometric image-based garment texture generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19884–19893, 2026a.

Hui Shan, Keyang Luo, Ming Li, Sizhe Zheng, Yanwei Fu, Zhen Chen, and Xiangru Huang. Interact3d: Compositional 3d generation of interactive objects. *arXiv preprint arXiv:2603.16085*, 2026b.

Jiapeng Tang, Yinyu Nie, Lev Markhasin, Angela Dai, Justus Thies, and Matthias Nießner. Diffuscene: Denoising diffusion models for generative indoor scene synthesis. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20507–20518. IEEE, 2024a.

Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In *European Conference on Computer Vision*, pp. 1–18. Springer, 2024b.

Dmitry Tochilkin, David Pankratz, Zexiang Liu, Zixuan Huang, Adam Letts, Yangguang Li, Ding Liang, Christian Laforte, Varun Jampani, and Yan-Pei Cao. Triposr: Fast 3d object reconstruction from a single image. *arXiv preprint arXiv:2403.02151*, 2024.

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.

Hengyi Wang and Lourdes Agapito. 3d reconstruction with spatial memory. In *2025 International Conference on 3D Vision (3DV)*, pp. 78–89. IEEE, 2025.

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

Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10510–10522. IEEE, 2025b.

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

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

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

Kailu Wu, Fangfu Liu, Zhihan Cai, Runjie Yan, Hanyang Wang, Yating Hu, Yueqi Duan, and Kaisheng Ma. Unique3d: High-quality and efficient 3d mesh generation from a single image. *Advances in Neural Information Processing Systems*, 37:125116–125141, 2024.

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

Hongchi Xia, Xuan Li, Zhaoshuo Li, Qianli Ma, Jiashu Xu, Ming-Yu Liu, Yin Cui, Tsung-Yi Lin, Wei-Chiu Ma, Shenlong Wang, et al. Sage: Scalable agentic 3d scene generation for embodied ai. *arXiv preprint arXiv:2602.10116*, 2026.

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

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

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

Jianing Yang, Alexander Sax, Kevin J Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21924–21935. IEEE, 2025.

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

Yandan Yang, Baoxiong Jia, Peiyuan Zhi, and Siyuan Huang. Physcene: Physically interactable 3d scene synthesis for embodied ai. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16262–16272. IEEE, 2024b.

Yue Yang, Fan-Yun Sun, Luca Weihs, Eli VanderBilt, Alvaro Herrasti, Winson Han, Jiajun Wu, Nick Haber, Ranjay Krishna, Lingjie Liu, et al. Holodeck: Language guided generation of 3d embodied ai environments. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16277–16287. IEEE, 2024c.

Yunhan Yang, Yuanchen Guo, Yukun Huang, Zi-Xin Zou, Zhipeng Yu, Yangguang Li, Yanpei Cao, and Xihui Liu. Holopart: Generative 3d part amodal segmentation. In *International Conference on Learning Representations*, volume 2026, pp. 52657–52678, 2026.

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

Hong-Xing Yu, Haoyi Duan, Charles Herrmann, William T Freeman, and Jiajun Wu. Wonderworld: Interactive 3d scene generation from a single image. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5916–5926. IEEE, 2025.

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

Zibo Zhao, Zeqiang Lai, Qingxiang Lin, Yunfei Zhao, Haolin Liu, Shuhui Yang, Yifei Feng, Mingxin Yang, Sheng Zhang, Xianghui Yang, et al. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation. *arXiv preprint arXiv:2501.12202*, 2025.

A Appendix

A.1 Accurate and Stable Bounding Box Localization

Every removal target needs an accurate box, since the editor removes exactly what the boxes mark. Each target comes with a name and a short description. A vision-language model boxes all targets in one call, one box per index, so every box stays bound to its target. The prompt requires tight boxes that touch the outermost pixels of the object. Two boxes with IoU of at least τ_{dup} are treated as one object boxed twice, and the extra one is detected again.

A global call can drift systematically in cluttered scenes, and it cannot notice its own drift. Each global box is therefore probed with one independent local sample of the same target. If the probe box overlaps the global box with IoU at least τ_p, the global box is confirmed. Otherwise the box is considered drifted, and the target is re-detected by local detection. Local detection samples the VLM n times independently for that target and keeps the samples that agree with at least one other sample,

$$\mathcal{C} \;=\; \{\, b_i \;:\; \exists\, j \neq i,\ \mathrm{IoU}(b_i, b_j) \geq \tau_c \,\}.$$

The element-wise median of these samples suppresses outliers and replaces the drifted box. Whenever the probe or local detection fails to return a box, the global box is kept. The global call covers all targets at once, while local detection stays stable. The cross-check keeps the strengths of both. We use $n = 3$, $\tau_{dup} = 0.85$, $\tau_c = 0.7$, and $\tau_p = 0.5$ in all experiments.

The verified target boxes are drawn in red with their indices, and the objects to protect in green, on the plan image. The annotated image is fed back to the VLM once more, and any movable object still without a box is added as an object to protect. A missed object is therefore protected rather than deleted by mistake.

A.2 Registration and Settling Details

The registration transform combines a global scale with bounded per-axis adjustments in the object's frame. An under-observed axis may only grow, since its measured span underestimates the true one. A well-observed axis may also contract, since its span is reliable. When all three axes agree that the asset is too small, we apply a single uniform scale instead, so the asset keeps its proportions. Each correction is kept only if it leaves the fit D_i no worse, so an asset may grow to cover its points but never shrink away from them. Image cues can refine orientation and translation, but only when this 3D fit is preserved.

An object is supported when its bottom rests on a surface beneath it, either the floor or the top of another object, as a cup rests on a tabletop. Formally, let $\mathrm{bot}(a_i')$ be the height of the lowest point of the settled asset, let h_{floor} be the floor height, and let $\mathrm{top}_j(a_i')$ be the height of the top surface of another asset a_j' within the footprint of a_i'. We set $\mathrm{supported}(a_i') = 1$ when the highest such surface at or below the bottom lies within a small contact tolerance ϵ_s of it, that is, when

$$0 \leq \mathrm{bot}(a_i') - \max H_i \leq \epsilon_s, \quad H_i = \{h_{\mathrm{floor}}\} \cup \{\, \mathrm{top}_j(a_i') \mid j \neq i,\ \mathrm{top}_j(a_i') \leq \mathrm{bot}(a_i') \,\}.$$

The floor is always in H_i, so an object with nothing beneath it is supported exactly when it touches the floor. We use $\epsilon_s = 3\,\mathrm{cm}$ in all experiments.

The clearance $\phi(a_i', a_j')$ between two assets turns negative as soon as they overlap. The budget $\mathcal{B}_i$ limits how far an asset may move from where it was observed. Let $d(a_i, C_i)$ be the distance between

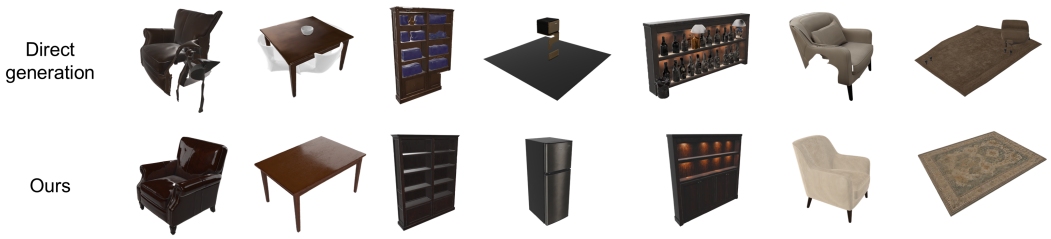

Figure 6: Generated assets with and without deconstruction. Each column shows one object. The top row is generated from the mask in the input view, and the bottom row from the mask captured at the best-observed moment. Incomplete masks lead to broken geometry, spurious parts, and hallucinated contents, while Unravel keeps the assets faithful.

the center of the asset and the center of its cluster before settling. We require

$$\mathcal{B}_i = \left\{ \delta \in \mathbb{R}^3 \,\middle|\, d(a_i', C_i) \le \max\left(d(a_i, C_i),\, \epsilon \right) \right\},$$

where ϵ is a small tolerance. An asset that starts at its observation may move by at most ϵ, and an asset that starts misplaced may move toward the observation but never further away. We set ϵ to $0.3\,\mathrm{m}$ in the horizontal plane and $0.15\,\mathrm{m}$ along the vertical direction in all experiments.

The validity mask M_k in Section 3.2 uses a fixed threshold $\tau = 0.5$.

A.3 Asset Scoring Protocol

The asset comparison in Section 4.3 is scored as follows. We evaluate all 160 objects from the five test scenes. The human evaluation involved 15 participants. Each object contributes two meshes, one from direct reconstruction and one from Unravel. Both meshes are rendered from the same four viewpoints at 90° intervals. The two renders appear side by side in random order, and the rater does not know which mesh comes from which pipeline. Each pair is judged against the same reference, a crop of the object's region from the image of its best-observed round. The rater answers one question, how faithfully each mesh reproduces the object in the reference.

Scores range from 1 to 10. A score of 1 to 3 means the shape is wrong or essential parts are missing. A score of 4 to 6 means the object category is recognizable but the structure is deformed or incomplete. A score of 7 or 8 means the overall shape is right with minor surface artifacts. A score of 9 or 10 means the mesh is faithful in shape and structure, including the parts that were occluded in the input image. We use GPT-5.5 (OpenAI, 2026) as the VLM rater. It receives the same renders, reference, and rubric. The model and the human raters score independently, and neither sees the other's ratings. Table 1 reports the mean score of each side over all 160 objects, and Fig. 6 shows representative examples of the quality gap.

A.4 Additional Scene Reconstruction Results

Figure 7 compares Unravel with Seeing Through Clutter (Aguina-Kang et al., 2026) on a cluttered image. We follow its open-source configuration, except that we upgrade its planner to GPT-5.5 to match Unravel. In this example, Seeing Through Clutter produces fragmented geometry and misplaced objects. Unravel more closely preserves the main furniture arrangement.

Figures 8, 9, and 10 present sixteen additional scenes reconstructed by the same pipeline with identical settings. The scenes cover kitchens, living rooms, and offices, as well as specialty spaces such as a yoga studio, a music room, a bar, and a conference room. Figure 8 traces the full loop and shows the assets extracted in every round. Figure 9 follows the same loop on four more scenes and highlights how the edited image evolves round by round. Figure 10 presents eight further scenes with their input images and multi-view renderings. Across all scenes, the planner removes objects from front to back, the editor restores the freed regions plausibly, and the reconstructed scenes preserve the input layout. These results demonstrate the feasibility and robustness of our pipeline.

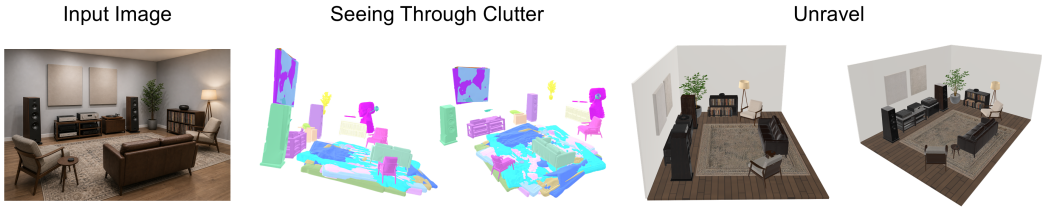

Figure 7: Comparison with Seeing Through Clutter. The input image is on the left, followed by two views of Seeing Through Clutter in the middle and two views of Unravel on the right. Seeing Through Clutter is shown as colored geometry and Unravel as a textured scene.

We also present reconstructions from six real indoor photographs in Fig. 11. The photographs were selected from Wikimedia Commons and show hotel interiors, a residential living room, and a clinic waiting room. The renderings retain the main furniture arrangements.

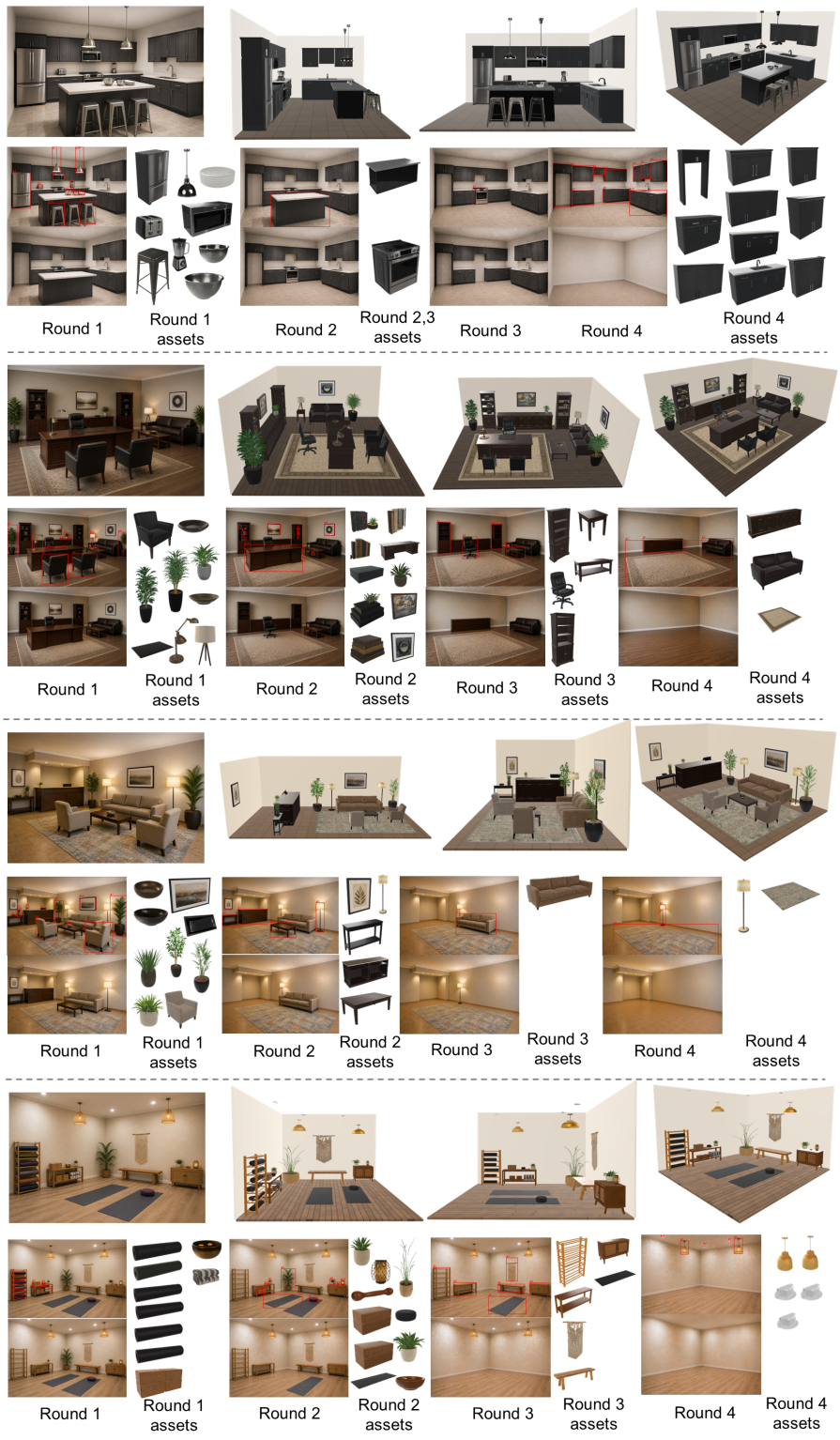

Figure 8: Complete deconstruction traces on four scenes. For each scene we show the input image, three views rendered from the reconstructed scene, the edited image of every round, and the assets extracted in each round. Red boxes mark the removal targets of each round, and duplicate assets are omitted for space.



Figure 9: Deconstruction traces on four more scenes. For each scene we show the input image, three views rendered from the reconstructed scene, and the edited image of every round. Red boxes mark the removal targets of each round.



Figure 10: Additional reconstructions on eight further scenes. For each scene we show the input image and three views rendered from the reconstructed scene.



Figure 11: Reconstructions from real photographs. Each row shows an input image on the left and three views of the reconstructed scene on the right.