ASTER: ARCHITECTING SIM-READY WORLDS FROM REAL SCENES FOR EMBODIED AI

Anonymous authors
Paper under double-blind review

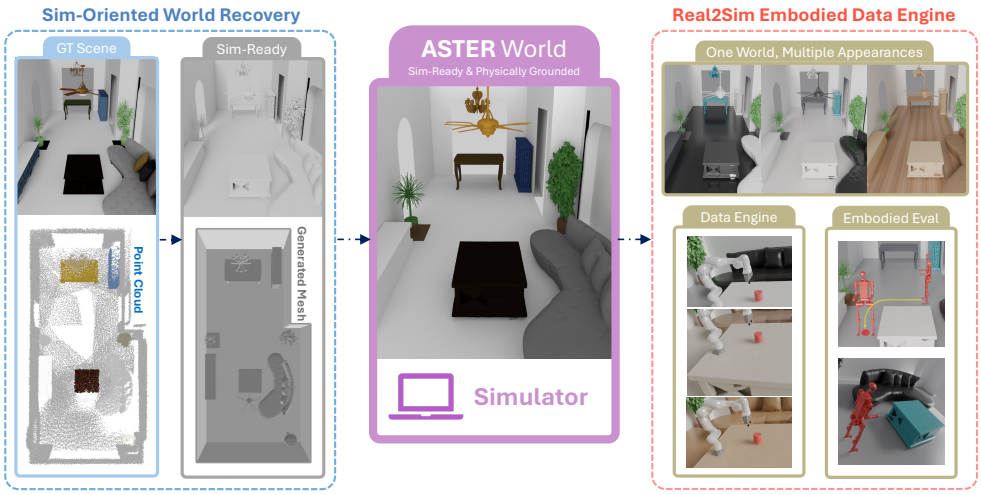


Figure 1: ASTER overview. ASTER transforms real scene observations into structured and physically executable simulation ready worlds. The recovered worlds preserve task relevant scene structure while supporting multiple visual appearances, scalable embodied data generation, and downstream policy evaluation.

ABSTRACT

Embodied AI increasingly relies on simulation for training and evaluation, creating a growing demand for scalable Real2Sim pipelines that transform real-world observations into simulation-ready environments. However, existing 3D reconstruction and digital twin approaches primarily optimize for visual fidelity, often producing unstructured meshes or cluttered scenes that are difficult to deploy in embodied simulators. We argue that Real2Sim should be formulated as a simulation-ready world generation problem rather than a geometry reconstruction problem. To this end, we present ASTER, a framework for architecting sim-ready worlds from real scenes for embodied AI. ASTER integrates simulation-oriented world understanding, world construction, and physics-based refinement to recover structured scene layouts and object assets, predict and assign physically plausible properties, and use simulation feedback to improve scene stability and executability. This design enables efficient world generation without extensive per-scene rendering. Evaluation of reconstructed scenes shows fewer collisions and higher stability, while policies trained in ASTER worlds achieve similar performance in matched ground-truth simulation scenes. In a challenging real-robot setting, adding ASTER-generated data increased observed real-robot success from 0/6 to 4/6 trials, providing a cheap way to scale up data for real-world robot deployment.

1 INTRODUCTION

Embodied AI has witnessed rapid progress with the emergence of vision-language-action (VLA) models (Kim et al., 2024; Zitkovich et al., 2023; Intelligence et al., 2025; Ma et al., 2026) and world models (Luo et al., 2026; Lyu et al., 2025; Ye et al., 2026), bringing foundation models closer to

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.

real-world deployment. Similar to large language models, these embodied AI models critically rely on massive amounts of high-quality embodied data to acquire robust perception, reasoning, and manipulation capabilities. However, collecting real-world embodied data at scale remains prohibitively expensive, requiring real robots to operate in diverse environments under extensive human supervision (Open X-Embodiment Collaboration et al., 2024; Khazatsky et al., 2024). While synthetic data produced by simulation provides a scalable alternative (Liu et al., 2023; Mees et al., 2022), existing simulated environments often lack the diversity, realism, and physical complexity of real-world scenes, resulting in a substantial sim-to-real gap. Consequently, in order to improve model generalization and downstream adaptation, developing efficient and scalable Real2Sim pipelines that transform real-world observations into high-quality simulation-ready environments has become a fundamental challenge for embodied AI (Wang et al., 2026).

To address this challenge, two parallel research directions have made remarkable progress. On one hand, recent advances in 3D scene reconstruction and digital twins enable increasingly realistic recovery of point clouds, meshes, and scene geometry from real-world observations (Mur-Artal et al., 2015; Mur-Artal & Tardós, 2017; Campos et al., 2021; Wang et al., 2025). On the other hand, synthetic simulators provide scalable environments for robot training (Todorov et al., 2012; Xia et al., 2026b; Ranawaka et al., 2026). Despite the success of both directions, a fundamental gap still exists between them. Existing 3D reconstruction methods primarily pursue geometric fidelity and visual realism (Siddiqui et al., 2026; Huang et al., 2025; Xia et al., 2026a), but typically do not recover the structured and physics-aware representations required for embodied simulation, such as object-level decomposition and plausible physical properties. In contrast, embodied simulation research largely focuses on constructing environments in simulators, assuming that scene assets and simulation configurations are already available rather than constructing them from real-world observations. The missing piece is a **Real2Sim pipeline for scalable embodied AI data generation** that recovers room layouts and independently usable object assets from real observations and configures them for simulation. Accordingly, **Real2Sim should be formulated as a simulation-oriented world generation problem rather than a 3D rendering or geometry reconstruction problem.**

To this end, we present **ASTER**: **A**rchitecting **S**imula**T**ion-Ready Worlds from **R**eal Scenes for Embodied AI, a Real2Sim framework that addresses three challenges in turning an indoor point cloud into an executable world. First, a point cloud does not directly specify the room layout, individual objects, or the attributes needed to simulate them. ASTER uses a 3D vision-language model (3D VLM) to recover the layout, object locations and captions, and predict mass, friction, and mobility attributes for use in simulation. Second, overlapping object observations can lead to duplicated or intersecting meshes. ASTER progressively reconstructs individual object assets from their point clouds and assembles them with the recovered layout and assigned simulation attributes. Third, the assembled world may still contain unstable or colliding objects. ASTER executes it in a physics simulator and uses the feedback to iteratively correct object placements and mobility assignments. The resulting world can be imported into simulators such as Isaac Sim and MuJoCo (Todorov et al., 2012) for synthetic data generation and embodied evaluation.

Specifically, this paper makes the following contributions:

- We formulate embodied Real2Sim as generating complete, structured, and executable worlds from real observations for scalable embodied AI data generation.
- We propose ASTER, an integrated Real2Sim system that combines simulation-oriented world understanding, progressive instance generation, and simulation-guided refinement to construct executable worlds from point clouds.
- Experiments show improved object reconstruction, fewer collisions and higher stability under the reported metrics, close policy performance in matched simulation scenes, and observed real-robot success of 4/6 trials with additional ASTER data versus 0/6 with real data alone under a challenging setting.

2 Related Work

Real2Sim scene reconstruction. SpatialLM (Mao et al., 2025) recovers room layouts and object boxes from point clouds, but does not generate object geometry or physical properties. ShapeR (Siddiqui et al., 2026) reconstructs individual object meshes, while RAS (Dong et al., 2026) composes scenes from videos with an emphasis on visual and geometric fidelity. HoloScene (Xia et al., 2025)

Table 1: Capability comparison of Real2Sim and simulation ready scene generation systems. The table summarizes whether each method supports complete and structured scenes, simulator execution, scalable generation, automated processing, physical properties, scene understanding, scene generation, and multiple robot embodiments. Purple blue headers highlight the capabilities targeted by ASTER.

	Real2Sim	Complete Scene	Structured Scene	Executable Scene	Scalable Scene	Automated Pipeline	Physics Properties	Scene Und.	Scene Gen.	Multi-Embodi.
SpatialLM (2025)	✗	✓	✓	✗	✗	✗	✗	✓	✗	✗
ShapeR (2026)	✗	✗	✓	✗	✗	✓	✗	✓	✓	✗
SimRecon (2026a)	✓	✓	✓	✓	✗	✓	✓	✓	✓	✗
HoloScene (2025)	✓	✓	✓	✓	✗	✗	✓	✓	✓	✗
RAS (2026)	✗	✓	✓	✗	✗	✓	✗	✓	✓	✗
SAGE (2026b)	✗	✓	✓	✓	✓	✓	✓	✗	✓	✗
PolaRiS (2025)	✓	✗	✓	✓	✓	✗	✗	✗	✓	✗
RobotArena (2025)	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓
R2S-Soft (2025)	✓	✗	✓	✓	✗	✗	✓	✗	✓	✗
SimFoundry (2026)	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓
ASTER (Ours)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

and SimRecon (Xia et al., 2026a) construct simulation ready scenes, but require reconstruction and optimization for each scene. ASTER instead recovers layout and object assets from point clouds, predicts physical attributes for simulation and assembles them into an executable world.

Simulation for robot learning and evaluation. PolaRiS (Jain et al., 2025), RobotArena∞ (Jangir et al., 2025), R2S-Soft (Zhang et al., 2025), and SimFoundry (Ranawaka et al., 2026) convert real observations into simulated environments, primarily for particular robot setups or tabletop tasks. SAGE (Xia et al., 2026b) generates physically validated indoor scenes at scale, but starts from text rather than real observations. ASTER recovers complete, structured indoor worlds from real point clouds for simulation and embodied learning. The full comparison of ASTER and baseline properties are listed in Table 1.

3 METHOD

3.1 PROBLEM FORMULATION

We argue that a Real2Sim system for embodied AI should generate a *simulation-oriented world* that serves as a scalable embodied data engine, rather than just recovering visually faithful geometry. Such a world should satisfy four key requirements.

(1) Completeness. The generated world should recover the full room or workspace, rather than an isolated tabletop scene, to support diverse embodied tasks.

(2) Structure. The world should represent interactive objects as separate entities with locations and semantic descriptions, so they can be manipulated independently and used to define tasks.

(3) Executability. The world should run in a physics simulator, with task relevant objects assigned plausible mass, friction, and mobility, while static structures remain fixed.

(4) Scalability. The world should preserve room structure and major objects while omitting incidental clutter. Small objects can be added and randomized from asset libraries during data generation.

These requirements motivate the world representation used by ASTER. Given an input point cloud $\mathcal{P}$, ASTER constructs

$$\mathcal{W} = \{\mathcal{L}, \mathcal{A}, \Phi\}, \tag{1}$$

where $\mathcal{L}$ denotes the room layout, $\mathcal{A} = \{a_i\}_{i=1}^{N}$ represents the set of simulation-ready object assets, and $\Phi = \{\phi_i\}_{i=1}^{N}$ denotes the simulator attributes predicted for them. The layout preserves the

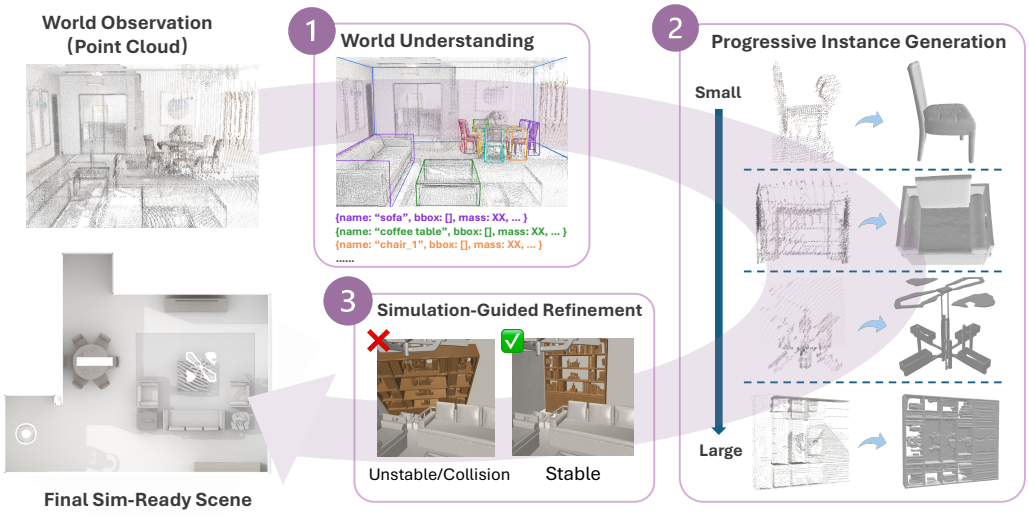


Figure 2: Overview of the ASTER pipeline. ASTER converts an input point cloud into a simulation ready world through three stages. Simulation oriented world understanding predicts the room layout and object level semantic and geometric attributes, together with predicted simulator attributes. Progressive instance generation reconstructs objects from small to large and assembles the recovered scene. Simulation guided refinement uses physics feedback to correct unstable or colliding configurations.

room structure, individual assets provide the structure needed to manipulate objects, and the predicted attributes configure those interactions for simulation. ASTER reconstructs the operable scene while leaving out small objects. These objects can be selected or designed from asset libraries for downstream tasks; recovering them from the input would constrain task diversity.

As shown in Figure 2, ASTER constructs $\mathcal{W}$ through three stages. First, world understanding recovers the room layout $\mathcal{L}$ and object-level spatial and semantic information from the input point cloud, together with predictions of simulator attributes. Second, progressive instance generation reconstructs object assets $\mathcal{A}$ from small to large instances and assembles them to construct the full recovered scene. Finally, physical refinement evaluates the constructed world in a physics simulator and corrects unstable or colliding configurations, producing the executable sim-ready world.

3.2 Simulation-Oriented World Understanding

Room layouts and object locations alone do not specify how a scene should behave in simulation. Given an input point cloud $\mathcal{P}$, ASTER first employs a 3D VLM to recover the structured scene information required for constructing the simulation-oriented world $\mathcal{W}$. The model generates a serialized textual representation of the scene, including the room layout $\mathcal{L}$ and, for each detected object i, its oriented 3D bounding box b_i, semantic caption c_i, and predicted simulator attributes $\phi_i \in \Phi$. The bounding box b_i specifies the object's position, size, and orientation, while

$$\phi_i = (m_i, \mu_i, s_i), \tag{2}$$

where m_i, μ_i, and s_i denote the predicted mass, friction coefficient, and mobility state, respectively. These predictions provide object descriptions and predicted simulator settings used to construct the simulation-ready object assets $\mathcal{A}$ in the subsequent stage.

We instantiate the 3D VLM with SpatialLM (Mao et al., 2025). Its original training data provides room layouts and object-level geometry and semantics, but does not include attributes for configuring objects in simulation. We therefore augment its textual training targets with mass, friction, and mobility annotations automatically inferred by a vision-language model from each object's semantic category and visual appearance. We retain the original SpatialLM architecture and continue training it on the augmented targets, enabling a single model to predict the scene layout, object instances, and physical attributes from the input point cloud. The mass and friction predictions are used as simulator inputs, not treated as measurements of the real objects or their contact pairs.

3.3 SIMULATION-READY WORLD CONSTRUCTION

Given the recovered room layout $\mathcal{L}$ and object-level predictions $\{(b_i, c_i, \phi_i)\}_{i=1}^{N}$, ASTER constructs a set of simulation-ready object assets $\mathcal{A} = \{a_i\}_{i=1}^{N}$. For each object, we crop its observation from the current scene point cloud $\mathcal{P}_{\text{cur}}$, initialized as $\mathcal{P}$, using the predicted bounding box b_i. A conditional 3D generative model $\mathcal{F}$ then reconstructs the corresponding object mesh:

$$\mathcal{P}_i = \text{Crop}(\mathcal{P}_{\text{cur}}, b_i), \qquad \mathcal{M}_i = \mathcal{F}(\mathcal{P}_i, c_i), \qquad a_i = (\mathcal{M}_i, b_i, c_i, \phi_i). \tag{3}$$

Unlike reconstruction pipelines that rely on precise instance masks, this bounding-box-based formulation directly uses the structured predictions from world understanding and avoids requiring exact reconstruction of every incidental scene detail. Small objects can instead be instantiated or randomized independently during downstream data generation (Ranawaka et al., 2026).

A key challenge is that neighboring bounding boxes may overlap, causing $\mathcal{P}_i$ to contain geometry from other instances and consequently producing duplicated or intersecting meshes. To reduce such cross-instance contamination, we introduce *progressive instance generation*. As shown in Figure 2, objects are ordered by physical size and reconstructed from smallest to largest. Before reconstructing each object, we filter points that are disconnected from its dominant point-cloud component and lie within the bounding boxes of neighboring objects.

After an object is reconstructed, its associated points are removed from $\mathcal{P}_{\text{cur}}$ before processing the next instance. The subsequent crop is therefore extracted from an increasingly cleaner scene observation. Repeating this process progressively reduces cross-instance contamination, duplicated geometry, and collisions in the assembled world.

3.4 SIMULATION-GUIDED PHYSICAL REFINEMENT

Even when the room layout and object meshes have been recovered, placement errors may leave the constructed world unstable in simulation. ASTER therefore uses simulator feedback to correct object placements and mobility assignments. After world construction, each object is assigned its model-predicted simulator attributes $\phi_i = (m_i, \mu_i, s_i)$ and simulated under gravity. The simulator returns a binary stability flag for each object indicating whether it reaches a stable resting state.

For unstable objects, ASTER first corrects their placement by aligning the bottom surface of the reconstructed mesh with its predicted supporting surface, such as the floor, a tabletop, or another object, and then re-evaluates the scene in simulation. This correction compensates for localization and mesh-generation errors that may introduce gaps, penetration, or unsupported geometry.

Remaining unstable cases are further used to refine the mobility state s_i of the assembled object assets. When simulation feedback indicates that an object should remain fixed to its surrounding structure, its mobility label is updated in the constructed world. After refinement, the final world $\mathcal{W}$ is exported in a simulator-compatible format for downstream embodied learning and data generation.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets. For simulation-oriented world understanding, we conduct continual pre-training on the SpatialLM dataset (Mao et al., 2025), which contains 12,328 synthetic indoor scenes comprising 54,778 rooms with ground-truth annotations of room layouts and oriented object bounding boxes. We augment the original annotations with mass, friction, and mobility labels produced by an offline VLM. We evaluate the resulting model on ScanNet (Dai et al., 2017), a real-world indoor scanning benchmark that is not used during training, to assess its generalization from synthetic training environments to real observations. For object generation, we use the validation data of 3D-FRONT (Fu et al., 2021) dataset, since it has the ground-truth objects. We initialize the generator from the ShapeR base model (Siddiqui et al., 2026) and further fine-tune it on object instances from 3D-FRONT training data. During inference, all objects detected by the world understanding model are passed to the object generation stage.

Baseline Selection. For layout recovery, we compare ASTER with V-DETR, SceneScript, and SpatialLM because they recover spatial structure from scene observations. For object generation,

Table 2: Layout recovery and object generation. Higher is better for F1 and F Score, while lower is better for CD.

Metric	V-DETR (2023)	SceneScript (2024)	SpatialLM (2025)	ShapeR (2026)	SimRecon (2026a)	HoloScene (2025)	ASTER
Layout F1@0.25 ↑	65.1	49.1	65.6	–	59.0	60.8	**66.1**
Layout F1@0.50 ↑	**56.8**	36.8	52.6	–	50.3	47.7	51.9
Object CD ↓	–	–	–	2.38	2.34	1.93	**1.67**
Object F Score ↑	–	–	–	72.25	69.65	68.11	**73.70**

we compare with ShapeR, SimRecon, and HoloScene because they reconstruct object geometry. For physical consistency, we compare with HoloScene, Holodeck, SceneWeaver, and SAGE because they report collision and stability of generated scenes; HoloScene provides a Real2Sim comparison, while the others provide context from scene generation. For the ablation study, we use SpatialLM + ShapeR (Fine Tuned) as the starting point because ASTER builds on these two components, then add the proposed stages one at a time.

Evaluation Metrics. For world understanding, we evaluate layout recovery using F1 scores at IoU thresholds of 0.25 and 0.50. For object generation, we report Chamfer Distance and F-Score between the generated and ground-truth object meshes. For physical consistency, we report collision and stability rates using MuJoCo (Todorov et al., 2012). A pair of objects is considered colliding before simulation if their intersection volume exceeds 5% of the smaller object's volume. Each scene is then simulated under gravity for a fixed warm-up period, and an object is considered stable if its displacement from the initial position is less than 0.2 m at the end of the warm-up period.

4.2 Simulation-Oriented World Recovery

We assess layout recovery, object generation, physical consistency, and the completeness and runtime of reconstructed scenes.

Layout Prediction and Object Generation Quality. Table 2 evaluates the two geometric components required to construct a simulation-ready world: room layout and object assets. For layout recovery, ASTER achieves F1 scores of 66.1 at IoU$= 0.25$ and 51.9 at IoU$= 0.50$, close to SpatialLM's 65.6 and 52.6 while extending its predictions with physical attributes for simulation. For object generation, ASTER achieves a Chamfer Distance of 1.67 and an F-Score of 73.70, compared with 2.38 and 72.25 for ShapeR. Together, these results show that ASTER recovers both the room structure and individual object geometry needed for subsequent world construction; the contributions of its stages are examined in Table 4.

Collision and Stability Analysis. Table 3 evaluates collisions and stability after room layouts and object assets are assembled. Compared with HoloScene, the Real2Sim baseline, ASTER reduces the collision rate from 25.2 to 18.0 and improves stability from 70.6 to 78.6. Holodeck, SceneWeaver, and SAGE provide additional context from synthetic scene generation; unlike ASTER, they do not need to preserve the structure of a real observed scene. These results show that ASTER can reduce collisions and improve stability under the reported metrics while retaining the recovered scene layout. Table 4 further examines how progressive instance generation and simulation-guided refinement contribute to this result.

Table 3: Physical consistency of generated scenes. Collision and stability scores for ASTER and representative prior systems. Lower collision and higher stability indicate better physical consistency. The gray rows denote synthetic scene generation methods rather than Real2Sim methods.

Method	Collision ↓	Stability ↑
HoloScene (2025)	25.2	70.6
Holodeck (2024)	22.0	63.8
SceneWeaver (2025)	32.8	67.7
SAGE (2026b)	1.9	99.9
ASTER (Ours)	**18.0**	**78.6**

Qualitative Results. Figure 3 shows ASTER's generated worlds alongside the ground truth across five rooms with different layouts and furniture arrangements. ASTER preserves the overall room geometry and places the major objects in their corresponding locations, producing coherent indoor scenes from varied inputs. This abstraction preserves the room structure and major objects used

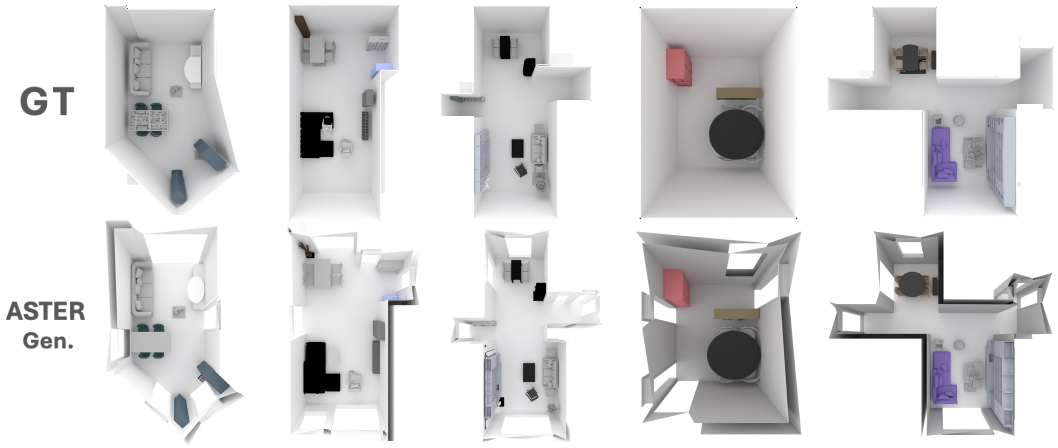


Figure 3: Qualitative comparison of reconstructed indoor scenes. The top row shows ground truth scenes and the bottom row shows the corresponding worlds generated by ASTER across diverse room layouts.

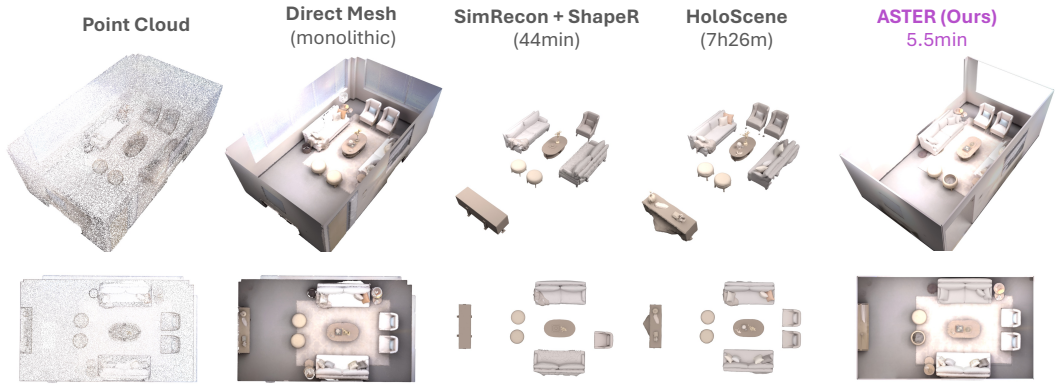


Figure 4: Qualitative comparison of scene reconstruction methods in one example. The top and bottom rows show perspective and top-down views, respectively. ASTER reconstructs the room and major objects in 5.5 minutes in this example.

here for embodied interaction while omitting incidental clutter; task-relevant small objects can be instantiated separately.

Figure 4 compares ASTER with alternative reconstruction approaches on the same input. Direct Mesh retains the room as a monolithic surface without independently represented furniture, while the SimRecon + ShapeR and HoloScene outputs shown here recover individual objects without assembling the surrounding room. ASTER reconstructs the room structure and furniture as separate object assets in this example. In this example, **ASTER takes only 5.5 minutes, compared with 44 minutes for SimRecon + ShapeR and 7 hours 26 minutes for HoloScene.** ASTER constructs the scene directly from the point cloud without per-frame rendering, enabling this substantially shorter generation time, which means higher scalability in real-world scenarios.

4.3 Embodied Usability

We evaluate embodied usability in two settings: policies trained in ASTER worlds and tested in corresponding GT simulation scenes, and real-robot policies trained with ASTER-generated data.

Simulation-Based Embodied Evaluation. We further evaluate whether ASTER-generated worlds preserve the scene properties required for downstream embodied learning. Starting from three 3D-FRONT scenes, we construct scanned point-cloud observations following the SpatialLM protocol and reconstruct the corresponding simulation environments using ASTER. For each reconstructed scene, we consider three manipulation tasks, including *Pick & Place*, *Dish Stacking*, and

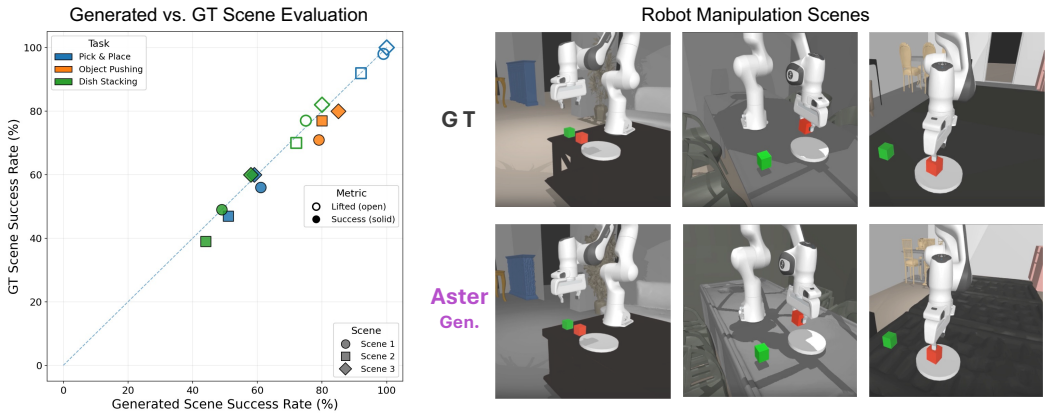


Figure 5: Embodied policy evaluation in matched simulation environments. Left: Policies are fine tuned in ASTER generated worlds and evaluated in both the generated worlds and their corresponding ground truth (GT) scenes. Points close to the diagonal indicate consistent performance across the two environments. Right: Representative manipulation tasks instantiated in the GT and ASTER worlds.

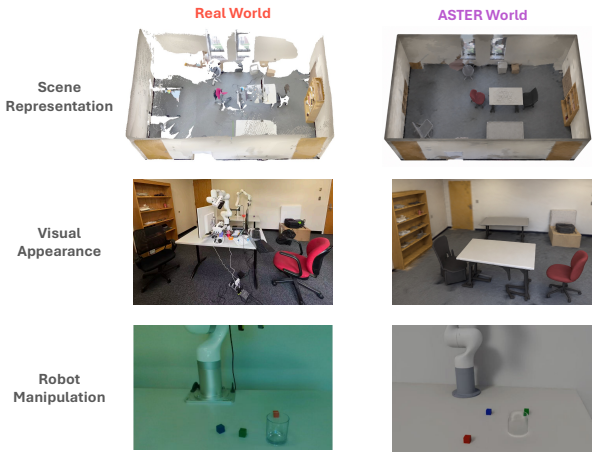
Object Pushing. We synthesize 100 training episodes for each task in the ASTER-generated world and fine-tune a $\pi_{0.5}$ policy on the resulting demonstrations. Each policy is then evaluated on 100 test episodes in both the ASTER-generated world and the corresponding GT scene. We report both the intermediate object-lifting rate and the final task success rate where applicable. We use Franka Panda robot arm in this experiment.

As shown in Figure 5, policies trained in ASTER worlds exhibit highly consistent performance when transferred to the corresponding GT environments. In particular, the lifting rates remain within two percentage points between ASTER and GT scenes across all evaluated Pick & Place and Dish Stacking settings, indicating that the generated geometry provides comparable object accessibility and manipulation affordances. Final task success also remains close across the two environments, with an average absolute gap of only 3.7 percentage points over all scene-task pairs and a maximum gap of 8 points. For example, on Scene 1, the Pick & Place policy achieves 61% success in the ASTER world and 56% in the GT scene, while Object Pushing achieves 79% and 71%, respectively. Similar trends are observed across the other scenes and tasks.

These results show that ASTER worlds preserve the spatial configurations and interactions needed for embodied policy learning. **Policies trained entirely on demonstrations generated in ASTER worlds achieve closely matched performance in the corresponding GT simulation scenes.** This supports using the reconstructed worlds to generate training data without manual adjustments.

Real-World Robot Evaluation. We run the complete pipeline on a real workspace. We record an RGB video of our laboratory, reconstruct a point cloud from it, and feed the point cloud to ASTER, which recovers the room layout and furniture as separate simulation-ready assets (Figure 6). We then add the task objects to this world and automatically synthesize 200 episodes with a simulated UFACTORY xArm 850. Separately, we manually collect 50 real episodes on the physical xArm 850. We fine-tune two $\pi_{0.5}$ policies: one on the 50 real episodes only, and one on the same 50 real episodes plus the 200 ASTER episodes. Both policies are deployed on the real robot in an easy setting (10 trials) and a hard setting (6 trials). In the easy setting, both policies succeed in 8 of 10 trials. In the hard setting, the real-only policy fails all six trials, while the policy that also uses ASTER data succeeds in four of six.

ASTER turns a single video of the lab into 200 automatically generated episodes, providing a cheap way to scale up data and fix the failure cases of the real-only policy. A small amount of real data is still mixed into training for deployment in the real scene. Policies trained purely on simulated data remain hard to deploy on their own, likely due to factors outside scene reconstruction, such as inaccurate robot arm parameters in simulation; the visualizations in Figure 6 show that the reconstructed scene closely matches the real one in both appearance and structure.



Aggregate Performance

Setting	Training	Trials	S/P/F	Succ.
Easy	Real Only	10	8/1/1	80%
	Sim + Real	10	8/1/1	80%
Hard	Real Only	6	0/0/6	0%
	Sim + Real	6	4/0/2	**66.7%**

Per Trial Outcomes

Setting	Training	Trial Scores
Easy	Real Only	1, 1, 1, 0.1, 1, 1, 0.3, 1, 1, 1
	Sim + Real	1, 1, 1, 0.1, 1, 1, 1, 1, 1, 0.3
Hard	Real Only	0.1, 0.1, 0.1, 0.1, 0.1, 0.1
	Sim + Real	1, 1, 0.1, 1, 0.1, 1

Figure 6: Real world reconstruction and real robot evaluation. Left: the real robotic workspace and its corresponding ASTER world, shown through both scene level representation and visual appearance. Right: raw trial outcomes under easy and hard evaluation settings. "Real Only" denotes training with real robot demonstrations, while "Sim + Real" additionally incorporates synthetic demonstrations generated in the ASTER world. In the reported trial scoring, 1 denotes success, 0.3 partial progress, and 0.1 failure.

Table 4: Ablation study of the main components in ASTER. Object metrics change in the second row because continued training alters the predicted boxes used for cropping.

Method	Layout Recovery		Object Generation		Physical Consistency	
	F1@0.25 ↑	F1@0.50 ↑	CD ↓	F Score ↑	Collision ↓	Stability ↑
SpatialLM + ShapeR (Fine Tuned)	65.6	**52.6**	1.75	73.25	25.8	69.9
+ Simulation Oriented World Understanding	**66.1**	51.9	**1.67**	73.51	26.6	68.5
+ Progressive Instance Generation	**66.1**	51.9	**1.67**	**73.70**	24.1	70.7
+ Simulation Guided Refinement (ASTER)	**66.1**	51.9	**1.67**	**73.70**	**18.0**	**78.6**

4.4 Ablation Studies

Table 4 isolates the contribution of each stage by adding it to the SpatialLM + ShapeR (Fine Tuned) baseline. Simulation oriented world understanding adds model-predicted physical attributes while largely preserving layout recovery quality. These predictions alone do not resolve intersections or unsupported placements in the assembled scene, and collision and stability do not improve at this stage. Progressive instance generation improves object quality and reduces collisions by limiting interference between neighboring objects. Simulation guided refinement then corrects remaining placement and mobility errors. Together, these stages reduce the collision rate from 25.8 to 18.0 and raise stability from 69.9 to 78.6, showing how the complete pipeline reduces collisions and improves stability under the reported metrics.

5 Conclusion

In this work, we formulated Real2Sim for embodied AI as the generation of complete, structured, and executable worlds from real scene observations. ASTER addresses this goal through simulation-oriented world understanding, progressive instance generation, and simulation-guided physical refinement. Together, these stages recover room layouts and object assets from point clouds, predict and assign physically plausible simulator properties, reduce interference between neighboring objects, and improve collision and stability outcomes in simulation. The generated scenes preserve the major spatial structure and interactive objects of the source environments without requiring extensive per-scene rendering or reconstruction of every visual detail. Experiments show improved object reconstruction, fewer collisions, and higher stability under the reported metrics, while policies trained in ASTER worlds achieve closely matched performance in the corresponding GT simulation scenes. In one real-robot evaluation, adding ASTER-generated simulation data to real demonstrations changes observed task success from 0/6 to 4/6 trials under a challenging setting. These results demonstrate the value of simulation-ready world generation for embodied learning and evaluation.

REPRODUCIBILITY STATEMENT

ASTER uses publicly available starting models and datasets. We describe the main stages, annotation procedure, and evaluation metrics in the paper. The simulation-oriented world understanding stage is initialised from the public SpatialLM checkpoint (Mao et al., 2025) and the object generator from the public ShapeR checkpoint (Siddiqui et al., 2026); Sec. 3 defines the three stages and the interfaces between them, and the Implementation Details paragraph in Sec. A describes the hardware used for training and inference. The physical-property annotations that augment the SpatialLM training data are produced by a one-time offline pass with a public open-weights vision-language model, and the prompt and label schema for mass, friction and mobility are given in Sec. A. The benchmark datasets used for quantitative scene evaluation are public: we train on the SpatialLM dataset, fine-tune the object generator on 3D-FRONT object instances, and evaluate on ScanNet, using the standard splits and protocols identified in Sec. 4. The physics-based validation stage and its refinement criteria are described in Sec. 3, and the reported metrics, layout F1 at IoU 0.25, Chamfer Distance, F-Score, collision and stability rates, are defined in the Evaluation Metrics paragraph of Sec. 4. We do not release model weights or our augmented annotations; the paper instead describes the public starting models and the pipeline.

AI USE STATEMENT

Large language and vision-language models appear in this work in two distinct roles, which we disclose separately.

Models used as components of ASTER. ASTER incorporates learned generative models as components of the proposed system. Specifically, it uses a 3D vision-language model for simulation-oriented scene understanding, a conditional 3D generative model for object mesh reconstruction, and an open-weights vision-language model in a single offline pass to annotate physical properties. The subsequent physics-based refinement stage uses deterministic feedback from the simulator to correct object placements and mobility assignments, without invoking an additional language model. These models are components of the research method, rather than tools used to produce the manuscript. Their roles, inputs, outputs, and evaluation are described in Secs. 3 and 4.

Generative AI used during research and writing. We used generative AI assistants to improve the clarity and readability of text written by the authors, search for and summarize relevant literature, check bibliographic information, and verify that baseline results quoted in the manuscript are consistent with their cited sources. We also used these tools to provide feedback on manuscript organization, experimental presentation, ablation design, and the clarity and internal consistency of our interpretation of the results. AI assisted tools were additionally used to revise LaTeX source and identify inconsistencies in terminology, equations, tables, and cross-references.

All research ideas, methodological decisions, experimental implementations, simulator configurations, data collection procedures, and final experimental choices were determined and approved by the authors. Generative AI tools did not autonomously conduct experiments or generate, modify, select, or discard experimental measurements. Every numerical result reported in the paper was produced by experiments conducted by the authors and was checked against the corresponding experimental record. Suggestions and text produced with AI assistance were treated as unverified drafts and were reviewed, corrected, and approved by the authors. Citations and externally reported baseline values were independently checked against their original sources. The authors take full responsibility for the final content of this paper, including all text, claims, experimental results, citations, and artifacts produced or revised with the assistance of generative AI.

REFERENCES

Armen Avetisyan, Christopher Xie, Henry Howard-Jenkins, Tsun-Yi Yang, Samir Aroudj, Suvam Patra, Fuyang Zhang, Duncan P. Frost, Luke Holland, Campbell Orme, Jakob Julian Engel, Edward Miller, Richard A. Newcombe, and Vasileios Balntas. Scenescript: Reconstructing scenes with an autoregressive structured language model. In *European Conference on Computer Vision*, 2024. URL https://api.semanticscholar.org/CorpusID:268536695.

Carlos Campos, Richard Elvira, Juan J. G'omez Rodr'iguez, José M. M. Montiel, and Juan D. Tardós. Orb-slam3: An accurate open-source library for visual, visualâ€“inertial, and multimap slam. *IEEE Transactions on Robotics*, 37:1874–1890, 2021. URL https://api.semanticscholar.org/CorpusID:220713377.

Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 2432–2443. IEEE, 2017.

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. *ArXiv*, abs/2604.10789, 2026. URL https://api.semanticscholar.org/CorpusID:287432560.

Huan Fu, Bowen Cai, Lin Gao, Ling-Xiao Zhang, Jiaming Wang, Cao Li, Qixun Zeng, Chengyue Sun, Rongfei Jia, Binqiang Zhao, et al. 3d-front: 3d furnished rooms with layouts and semantics. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10913–10922. IEEE, 2021.

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

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. pi0.5: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Arhan Jain, Mingtong Zhang, Kanav Arora, William Chen, Marcel Torne, Muhammad Zubair Irshad, Sergey Zakharov, Yue Wang, Sergey Levine, Chelsea Finn, Wei Ma, Dhruv Shah, Abhishek Gupta, and Karl Pertsch. Polaris: Scalable real-to-sim evaluations for generalist robot policies. *ArXiv*, abs/2512.16881, 2025. URL https://arxiv.org/abs/2512.16881.

Yash Jangir, Yidi Zhang, Kashu Yamazaki, Chenyu Zhang, Kuan-Hsun Tu, Tsung-Wei Ke, Lei Ke, Yonatan Bisk, and Katerina Fragkiadaki. Robotarena ∞: Scalable robot benchmarking via real-to-sim translation. *ArXiv*, abs/2510.23571, 2025. URL https://arxiv.org/abs/2510.23571.

Alexander Khazatsky, Karl Pertsch, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *ArXiv*, abs/2403.12945, 2024. URL https://arxiv.org/abs/2403.12945.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qian Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *ArXiv*, abs/2306.03310, 2023. URL https://arxiv.org/abs/2306.03310.

Hao Luo, Wanpeng Zhang, Yicheng Feng, Sipeng Zheng, Haiweng Xu, Chaoyi Xu, Ziheng Xi, Yuhui Fu, and Zongqing Lu. Being-h0.7: A latent world-action model from egocentric videos, 2026. URL https://arxiv.org/abs/2605.00078.

Jiangran Lyu, Ziming Li, Xuesong Shi, Chaoyi Xu, Yizhou Wang, and He Wang. Dywa: Dynamics-adaptive world action model for generalizable non-prehensile manipulation. *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11058–11068, 2025. URL https://api.semanticscholar.org/CorpusID:277244416.

Yueen Ma, Zixing Song, Yuzheng Zhuang, Jianye Hao, and Irwin King. A survey on vision–language–action models for embodied ai. *IEEE Transactions on Neural Networks and Learning Systems*, 2026.

Yongsen Mao, Junhao Zhong, Chuan Fang, Jia Zheng, Rui Tang, Hao Zhu, Ping Tan, and Zihan Zhou. Spatiallm: Training large language models for structured indoor modeling. *ArXiv*, abs/2506.07491, 2025. URL https://arxiv.org/abs/2506.07491.

Oier Mees, Lukás Hermann, Erick Rosete-Beas, and Wolfram Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 7:7327–7334, 2022. URL https://api.semanticscholar.org/CorpusID:244908821.

Raul Mur-Artal and Juan D. Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33:1255–1262, 2017. URL https://api.semanticscholar.org/CorpusID:206775640.

Raul Mur-Artal, José M. M. Montiel, and Juan D. Tardós. Orb-slam: A versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31:1147–1163, 2015. URL https://api.semanticscholar.org/CorpusID:206775100.

Open X-Embodiment Collaboration et al. Open x-embodiment: Robotic learning datasets and rt-x models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6892–6903, 2024. URL https://arxiv.org/abs/2310.08864.

Nadun Ranawaka, Josiah Wong, Wei-Lin Pai, Wei-Teng Chu, Tianyuan Dai, Masoud Moghani, Hang Yin, Yunfan Jiang, Wesley Durbano, Brandon Huynh, Linxi Fan, Danfei Xu, Ruohan Zhang, Fei-Fei Li, Bowen Wen, Ajay Mandlekar, and Yuke Zhu. Simfoundry: Modular and automated scene generation for policy learning and evaluation. 2026. URL https://api.semanticscholar.org/CorpusID:288856323.

Yichao Shen, Zigang Geng, Yuhui Yuan, Yutong Lin, Ze Liu, Chunyu Wang, Han Hu, Nanning Zheng, and Baining Guo. V-detr: Detr with vertex relative position encoding for 3d object detection. *ArXiv*, abs/2308.04409, 2023. URL https://arxiv.org/abs/2308.04409.

Yawar Siddiqui, Duncan P. Frost, Samir Aroudj, Armen Avetisyan, Henry Howard-Jenkins, Daniel DeTone, Pierre Moulon, Qirui Wu, Zhengqin Li, Julian Straub, Richard A. Newcombe, and Jakob Engel. Shaper: Robust conditional 3d shape generation from casual captures. *ArXiv*, abs/2601.11514, 2026. URL https://arxiv.org/abs/2601.11514.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, 2012. URL https://api.semanticscholar.org/CorpusID:5230692.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotný. Vggt: Visual geometry grounded transformer. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5294–5306, 2025. URL https://api.semanticscholar.org/CorpusID:277043968.

Ziyao Wang, Bingying Wang, Hanrong Zhang, Tingting Du, Tianyang Chen, Guoheng Sun, Yexiao He, Zheyu Shen, Wanghao Ye, and Ang Li. Vision-language-action in robotics: A survey of datasets, benchmarks, and data engines. *ArXiv*, abs/2604.23001, 2026. URL https://arxiv.org/abs/2604.23001.

Chong Xia, Kai Zhu, Zizhuo Wang, Fangfu Liu, Zhizheng Zhang, and Yueqi Duan. Simrecon: Simready compositional scene reconstruction from real videos, 2026a. URL https://arxiv.org/abs/2603.02133.

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. *ArXiv*, abs/2510.05560, 2025. URL https://arxiv.org/abs/2510.05560.

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*, abs/2602.10116, 2026b. URL https://arxiv.org/abs/2602.10116.

Yandan Yang, Baoxiong Jia, Shujie Zhang, and Siyuan Huang. Sceneweaver: All-in-one 3d scene synthesis with an extensible and self-reflective agent. *ArXiv*, abs/2509.20414, 2025. URL https://arxiv.org/abs/2509.20414.

Yue Yang, Fan-Yun Sun, Luca Weihs, Eli VanderBilt, Alvaro Herrasti, Winson Han, Jiajun Wu, Nick Haber, Ranjay Krishna, Lingjie Liu, Chris Callison-Burch, Mark Yatskar, Aniruddha Kembhavi, and Christopher Clark. Holodeck: Language guided generation of 3d embodied ai environments. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16227–16237, 2024. URL https://api.semanticscholar.org/CorpusID:266210109.

Angen Ye, Boyuan Wang, Chaojun Ni, Guan Huang, Guosheng Zhao, Hao Li, Hengtao Li, Jie Li, Jindi Lv, Jingyu Liu, Mingyu Cao, Peng Li, Qiuping Deng, Wenjun Mei, Xiaofeng Wang, Xinze Chen, Xinyu Zhou, Yang Wang, Yifan Chang, Yifan Li, Yukun Zhou, Yun Ye, Zhichao Liu, and Zheng Zhu. Gigaworld-policy: An efficient action-centered world-action model. *ArXiv*, abs/2603.17240, 2026. URL https://arxiv.org/abs/2603.17240.

Kaifeng Zhang, Shuo Sha, Hanxiao Jiang, Matthew Loper, Hyun Oh Song, Guangya 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*, abs/2511.04665, 2025. URL https://arxiv.org/abs/2511.04665.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pp. 2165–2183. PMLR, 2023.

A Additional Experimental Set-ups

A.1 Implementation Details

World understanding. We initialise from the public SpatialLM1.1-Llama-1B checkpoint and train both the language and point towers with nothing frozen, on four NVIDIA A100 80 GB GPUs (Table 5). Physical attributes are emitted in the same forward pass as the layout; no VLM is used at inference.

Table 5: World-understanding training hyperparameters.

Setting	Value
Initialisation	SpatialLM1.1-Llama-1B
Learning rate / schedule	1×10^{-5}, cosine, warm-up ratio 0.03
Epochs	10
Batch size	1 per device $\times$ 4 GPUs, grad. accumulation 4
Sequence length	8192 tokens
Precision	bf16, TF32 matmuls
Spatial discretisation	1280 bins
Augmentation	random rotation about the vertical axis

Inference cost. The full pipeline runs on a single A100. A room-scale scene with 13 objects takes 5.5 minutes end to end: about 1.5 minutes for world understanding, 2.6 minutes for instance generation, and 0.5 minutes for assembly.

Evaluation data. Layout recovery is evaluated on ScanNet (Dai et al., 2017), which is never seen during training. Object generation is evaluated on 6,143 objects from 1,000 3D-FRONT rooms (Fu et al., 2021), and physical consistency on the full 3D-FRONT test split, since collision and stability are scene-level statistics. HoloScene is run on 10 rooms only, as it takes 5–10 hours per scene.

Object generation metrics. We follow ShapeR's official protocol. Generated and ground-truth meshes are scaled by $2\max(\text{bounds})$, so metrics are normalised per object. Chamfer Distance is the L_1 form, reported $\times 10^2$; the F-Score threshold is $\texttt{linspace}(1/144, 2, 1000)[10] \approx 0.0269$ in the normalised scale. We sample 10^5 points per mesh for both metrics.

Physical consistency metrics. Collision and stability follow the definitions in Sec. 4. Scenes are simulated in MuJoCo (Todorov et al., 2012) at a 2 ms timestep with a two-stage warm-up that lets objects settle before full gravity: 500 steps at $g = -1.0\,\mathrm{m/s^2}$ with velocity damping 0.95 per step, then 1,000 steps at $g = -9.81\,\mathrm{m/s^2}$ with damping 0.98 (3.0 s in total). HoloScene, the only Real2Sim baseline, is re-run on our data and scored with this protocol. Holodeck, SceneWeaver and SAGE generate scenes rather than recover them, so there is no shared input to re-run them on; their numbers are quoted from the original papers for context only.

Simulation-guided refinement. Objects failing the stability test are corrected in two rounds. First, placement: among the floor plane and the upward-facing tops of objects whose xy footprint overlaps the target, we take the highest surface below the target's bottom face and translate the target vertically onto it, leaving the observed horizontal layout unchanged. This removes gaps and interpenetrations caused by localisation and mesh errors. Second, objects still unstable after re-simulation typically have no support beneath them (e.g. wall- or ceiling-mounted items); their mobility state s_i is set to static and they are attached to the world body. The loop stops when no state changes, usually within two rounds, with a hard limit of three. Corrected mobility labels are written back to the training annotations.

A.2 Physical Attribute Annotation

The SpatialLM corpus provides layouts and oriented boxes but no physical attributes. We add $\phi_i = (m_i, \mu_i, s_i)$ with a one-time offline pass of Qwen3-VL-8B, queried once per object with its category (from the existing annotation) and its highest-visibility crop:

```
You are annotating physical properties for a
rigid-body simulator.

Object category: {category}
Image: {object_crop}

Estimate the following three properties for this
object as it would be used in an indoor scene.
Answer with JSON only, no explanation.

  mass      total mass in kilograms, a positive
            float.
  friction  coefficient of sliding friction
            against a typical indoor floor,
            a float in [0, 1].
  is_static true if the object is fixed to the
            building structure (mounted on a wall
            or ceiling, or otherwise not intended
            to be moved during manipulation);
            false if it rests on a supporting
            surface and can be moved.

Answer format:
{"mass": <float>, "friction": <float>,
 "is_static": <bool>}
```

Responses are clamped to $m_i \in [0.5, 40]$ kg and $\mu_i \in [0.30, 0.65]$; unparsable responses fall back to the category median. Objects labelled static are those attached to the building shell, such as ceiling lights, wall-mounted paintings and mirrors, curtains and carpets. The attributes are appended to SpatialLM's box term, with mass written to two decimals and friction to three, so one autoregressive decode emits geometry and physics together:

```
bbox_N=Bbox(class, px,py,pz, az, sx,sy,sz,
            mass, friction, is_static)
```

The training prompt is extended accordingly ("Detect walls, doors, windows, boxes with physics attributes (mass, friction, is_static)"). These labels are VLM estimates used as simulator inputs, not measurements of real objects.

B Accuracy of Predicted Physical Attributes

Table 3 shows whether the recovered world behaves well in simulation, but not whether individual predictions are right: a scene can be stable simply because every object is heavy. We therefore compare the predicted attributes with the annotations directly. Predicted boxes are matched to annotated boxes on held-out rooms by Hungarian assignment on 3D IoU (IoU > 0.25); unmatched objects are excluded, so the numbers measure attribute accuracy given a correct detection. Because mass spans two orders of magnitude (0.5–40 kg), we report mean relative error $\frac{1}{N}\sum_i |\hat{y}_i - y_i|/y_i$ for mass and friction, and accuracy for mobility.

Table 6: Predicted physical attributes against the VLM annotations, over matched objects in held-out rooms.

Attribute	Metric	Value
Mass	mean relative error ↓	8.2%
Friction	mean relative error ↓	17.5%
Mobility	accuracy ↑	98.9%

The model reproduces the annotated mass closely, which is what matters for simulation, where behaviour under gravity depends mainly on mass ratios. Friction is harder, as it must be inferred from material appearance alone. Mobility is the most consequential attribute, since a mislabelled painting falls off the wall and a mislabelled sofa cannot be pushed; the remaining errors of the

movable-but-actually-fixed kind are caught by the stability test in the refinement stage. Note that all three numbers measure agreement with VLM annotations, not with physical ground truth.

C LIMITATIONS

ASTER targets indoor scenes represented as point clouds and prioritizes task-relevant structure over exhaustive reconstruction of every object and visual detail. Its predicted physical properties are simulator inputs intended to support executable scenes; they should not be interpreted as measurements of the corresponding real-world material or contact properties. Although simulation-based evaluation across three scenes shows similar policy performance in ASTER and ground-truth environments, this evidence does not establish generalization to arbitrary scenes, tasks, or simulators. The real-robot study further evaluates one workspace and one robot arm, and combines ASTER-generated data with real demonstrations. In the challenging setting, the reported improvement is based on six trials, so it should be viewed as preliminary evidence of the usefulness of ASTER-generated data rather than a broad estimate of real-world performance. Extending evaluation to more environments, object categories, robots, and tasks, and studying how to reduce the amount of real demonstration data required for deployment, remain important directions for future work.