GARDEN: Gravity-Aligned Reconstruction of Disentangled ENvironments from RGB images

Jiahao Sun[1,2⋆], Dingkun Wei[1,2⋆], Zehong Shen[2†], Hongyu Zhou[1,2⋆], Yujun Shen[2✉], and Liang Li[1✉]

[1] Zhejiang University, China
[2] Ant Group, China

Abstract. Converting multi-view RGB observations into simulation-ready 3D environments remains challenging because current reconstruction pipelines produce monolithic scene representations without explicit physical structure. They are typically defined up to an arbitrary global rotation and entangle rigid foreground objects with background geometry, which hinders stable physical interaction. Existing solutions often recover interactivity by replacing reconstructed objects with retrieved CAD assets, but this introduces a slow retrieval-and-replacement stage and weakens scene-specific geometric fidelity. We propose GARDEN, an RGB-only framework that reformulates reconstruction as *physically-grounded scene factorization* and outputs a structured hybrid scene representation. The key idea is to use gravity as a universal physical prior: we first align the reconstruction to a unified Gravity-View frame to resolve gauge ambiguity, then recover object-centric rigid meshes with accurate 6-DoF placement, and finally remove duplicate object geometry from the background through conditional 3D point classification. The resulting representation combines explicit rigid bodies with a decoupled background, enabling direct physics simulation while preserving visual realism. Experiments on both simulated and real multi-view scenes show that GARDEN improves object placement reliability, disentanglement quality, and rendering-simulation efficiency compared with retrieval-based baselines. Project page: `https://sunjiahaovo.github.io/garden/`

Keywords: Physically-Grounded Scene Factorization · Gravity-Aligned Reconstruction · Structured Hybrid Representation

1 Introduction

While recent RGB-only reconstruction systems [20, 27, 29, 30, 41, 45] achieve impressive visual quality, they typically produce weakly structured "3D photographs" that struggle to support physically grounded downstream applications. These outputs are largely monolithic, coupling foreground rigid objects

⋆ Work done while J. Sun, D. Wei, and H. Zhou were research interns at Ant Group.
✉ Corresponding author.
† Project lead.

Fig. 1: GARDEN factorizes unstructured RGB images into a structured, gravity-aligned representation. By decoupling scene-specific rigid objects from the background as meshes, the proposed method achieves high-fidelity reconstruction that enables interactive downstream simulation without the requirement of external CAD models.

and background geometry into a single representation that hinders independent object-level manipulation. Furthermore, they are defined only up to an arbitrary global rotation without an explicit gravity direction. Without this physically grounded global frame, rigid-body anchoring and physics-aware interactions in environments like Embodied AI [12] and simulations [28, 39] become fundamentally unstable.

To inject the necessary structure for interaction, existing pipelines often resort to CAD retrieval [16, 18, 22, 23], which unfortunately introduces a severe trade-off between semantic structure and scene-specific realism. These systems typically extract layout bounding boxes and replace real objects with retrieved artist-designed 3D assets. However, this CAD substitution compromises *instance-level geometric fidelity*, as pre-modeled assets rarely match the exact shape and fine-grained details of in-the-wild objects. Moreover, these pipelines are complex and brittle, accumulating errors across layout estimation, retrieval, and alignment stages. Finally, scaling this approach to diverse, long-tailed real-world objects is difficult, often requiring manual refinement.

In this work, our goal is to eliminate this compromise between structure and fidelity by proposing a **structured hybrid representation**. We seek a representation that is *both* structurally disentangled for independent object manipulation *and* strictly faithful to the scene-specific geometry captured in the real world. As visualized in Fig. 1, our approach decouples foreground rigid objects—represented as explicit standalone meshes with accurate 6-DoF poses—from a clean background environment, represented as high-fidelity point clouds or 3DGS. This combination of physical independence and geometric authenticity directly facilitates stable downstream simulation.

To construct this representation without CAD retrieval, our core insight is that gravity provides a universal physical prior to resolve gauge ambiguity and anchor objects consistently. We introduce GARDEN, an RGB-only pipeline that factorizes unstructured multi-view images into our structured hybrid representation. First, we resolve gauge ambiguity by extracting global camera tokens from a multi-view foundation model to regress the gravity direction, establishing a unified Gravity-View [38] frame where object placement is naturally constrained to maintain upright orientations. Next, guided by a lightweight target box that can be supplied by a user or generated by a vision-language model, we reconstruct scene-specific objects using 3D foundation models [7] and refine their 6-DoF poses [46] within this gravity-aligned space. Finally, to eliminate object-background entanglement, we design a conditional 3D point classification network that identifies and removes redundant object geometry from the background, yielding a clean environment for hybrid rendering.

We evaluate GARDEN against state-of-the-art scene generation and retrieval pipelines. By replacing heavy CAD retrieval and layout estimation with physically grounded factorization, our method substantially reduces processing time (e.g., from 4330s in LiteReality [16] to 560s), while improving geometric accuracy and rendering fidelity. In summary, our main contributions are:

- We propose a novel paradigm of *physically-grounded scene factorization* for RGB-only reconstruction, utilizing gravity as a physical prior to resolve gauge ambiguity and enable consistent rigid-body anchoring.
- We design a complete, CAD-free pipeline that extracts scene-specific explicit object meshes and accurately refines their 6-DoF poses within a unified Gravity-View coordinate frame.
- We introduce a conditional 3D point classification network to remove object redundancy from the background, enabling a *structured hybrid representation* with strong rendering fidelity and computational efficiency.

2 Related Work

Non Factorizable Scene Reconstruction. The task of 3D surface reconstruction has been comprehensively investigated via both optimization-driven and learning-based paradigms [17, 19, 31]. While recent advancements, including NeRF [29], 3DGS [20], and subsequent models [2, 4, 30, 51], produce exceptional quality in novel view synthesis, the primary focus of these techniques remains on visual rendering rather than the extraction of precise underlying geometry. To address the limitations in geometric accuracy, implicit representations based on SDF [26,42,50] have been developed to ensure geometric fidelity without compromising the quality of novel views. Furthermore, to bypass the extensive computational costs of per-scene optimization, feed-forward architectures [24,27,41,43,45] have been proposed to deduce the global geometry of scenes directly from unposed input images. Despite the efficiency of these approaches, the standard practices model entire scenes as holistic continuous surfaces. Consequently, individual object representations often suffer from severe incompleteness under

occlusions. Moreover, these frameworks leave a fundamental ambiguity in global orientation unresolved. Lacking an explicitly defined gravity direction or physical vertical axis, this gauge freedom entangles rigid objects with the background environment. Such monolithic structures present significant challenges for consistently anchoring, independently manipulating, or physically simulating individual entities within the reconstructed scene.

Factorizable Scene Reconstruction. The creation of factorizable environments is a fundamental problem for advanced visual synthesis and physically-consistent scene composition. Early systems tackle the factorization of scenes through joint detection and completion [8, 14, 35], procedural generation [11, 21, 25, 33], or the retrieval of CAD models [18, 22, 23]. Recent pipelines, such as LiteReality [16] and other language-guided frameworks [9, 10, 48], facilitate the construction of interactive environments derived from real-world observations. HoloScene [47] also targets simulation-ready interactive worlds from a single video with a scene-graph representation. However, many practical systems rely on proxy asset replacement. While this strategy supports basic scene arrangement, it frequently causes the reconstructed environments to drift from the exact geometry of the original objects. To better preserve object specificity, recent progress reframes reconstruction as conditional generation. Compositional pipelines [1, 7, 15, 49] utilize diffusion priors and generative models to assemble coherent scenes from single images. Despite offering open-vocabulary flexibility, these generative techniques struggle to integrate stably with multi-view reconstructed environments. More fundamentally, isolating entities from standard neural representations often inherits the aforementioned global orientation ambiguity. Lacking an explicitly defined gravity direction, extracted objects lack physically consistent global anchoring, hindering reliable 3D manipulation or rearrangement. To address these critical limitations, the proposed method reconstructs independent 3D shapes that are explicitly grounded within a physical coordinate system. Compared to prior techniques that may degrade across diverse scenes, our approach remains robust to casual capture conditions and successfully resolves the inherent gauge freedom. Consequently, this ensures that individual objects can be consistently anchored and independently manipulated for realistic scene simulation.

3 Method

GARDEN targets multi-view scene reconstruction for applications that require both physical interaction and high-fidelity rendering. Conventional multi-view reconstruction [27, 36, 41] produces a single monolithic geometry without explicit object-level separation, and the reconstructed scene is defined up to an arbitrary rotation, and is therefore not physically grounded. These properties limit structured scene decomposition and object-level physical interaction.

We formulate the problem as *physically-grounded scene factorization* driven by gravity. As illustrated in Fig. 2, we first establish a unified Gravity-View [38]

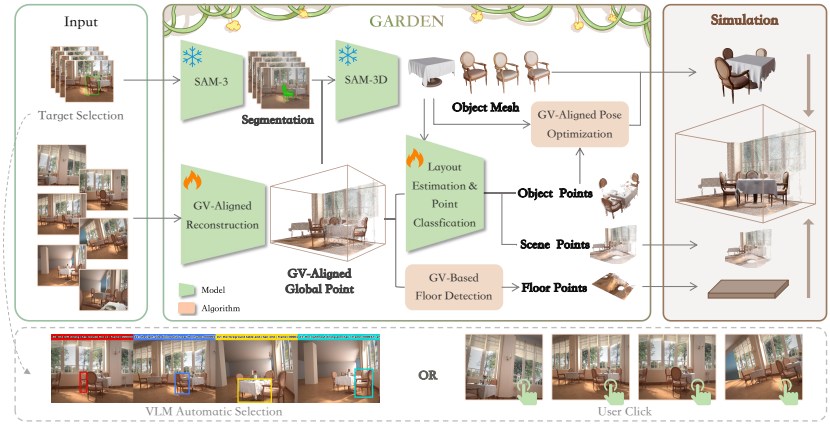

Fig. 2: Overview of GARDEN. Given multi-view RGB images, we first perform (1) *multi-view reconstruction with gravity alignment* to establish a physically grounded coordinate system. Next, we conduct (2) *target-driven object generation and layout estimation* to reconstruct standalone rigid objects from user- or VLM-provided boxes. This is followed by (3) *point-conditional background disentanglement* to remove duplicate object geometry from the static scene. Finally, these decoupled components enable (4) *unified physics simulation* with high-fidelity rendering.

(GV) coordinate frame to resolve global orientation ambiguity, and then decompose the scene into independent rigid bodies and a static background. Specifically, user-specified interactive objects are represented as standalone rigid meshes with accurate 6-DoF poses to enable simulation, while the background is represented by colored point clouds or 3D Gaussian Splatting (3DGS) to preserve visual fidelity. This hybrid representation supports both physical simulation (e.g., MuJoCo [39]) and photorealistic rendering.

3.1 Multi-View Reconstruction with Gravity-View Alignment

Unlike methods [16] that rely on depth or IMU sensors, our approach generates scene reconstructions solely from multi-view RGB images. This is facilitated by recent advancements in multi-view foundation models (e.g., VGGT [41], Pi3 [45], and DepthAnything-3 [27]), which enable the recovery of camera poses and scene structures from pure visual input. However, a significant challenge remains: the global coordinate systems of these models are implicitly determined by their training priors. For instance, VGGT and DepthAnything-3 typically anchor the scene to a reference camera frame, while Pi3 employs a permutation-equivariant approach that yields an indeterminate global orientation. Consequently, performing direct scene factorization on such reconstructions is hindered by global orientation ambiguities, and the recovered entities lack the gravity alignment required for downstream physical simulations. To address this, inspired by GVHMR [38], we propose the Gravity-View Align Module to align the reconstructed coordinate system with the physical direction of gravity.

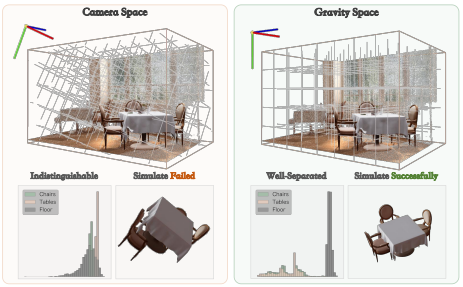

Fig. 3: Motivation for Gravity-View (GV) alignment. Point clouds reconstructed in the camera frame are globally tilted, whereas GV alignment restores an upright geometry consistent with physical gravity. This correction improves structural separability, as evidenced by the histogram in this figure: floor and object points, heavily entangled in the camera frame, become more distinguishable in GV space and therefore easier to factorize. In downstream physics simulation, GV-aligned scenes remain stable under gravity, while camera-frame reconstructions are gravity-inconsistent and tend to produce unstable object poses (e.g., tilting or falling).

We first construct the GV coordinate system by aligning the reference frame with the gravity direction. Taking the standard OpenCV camera convention as a reference: (1) within the camera coordinate system, we define the gravity direction as $\mathbf{g}$; (2) the y-axis of the GV frame is aligned with the gravity direction, such that $\mathbf{y} = \mathbf{g}$; (3) given the view direction in the camera frame $\mathbf{v} = [0, 0, 1]^\top$, we derive the x-axis of the GV frame as $\mathbf{x} = \mathbf{y} \times \mathbf{v}$; (4) finally, the z-axis is determined via the right-hand rule as $\mathbf{z} = \mathbf{x} \times \mathbf{y}$. Through this construction, the GV coordinate system is established, allowing for the direct derivation of the rotation matrix $\mathbf{R}_{c2gv}$ that transforms the original camera coordinates to the GV frame. As illustrated in Fig. 3, this alignment is not merely a coordinate normalization step: it improves geometric separability between support structures and objects, and is critical for stable downstream physical simulation.

To predict the transformation to the GV frame, we leverage the multi-view foundation model Depth-Anything-3 [27] for its robust scene representation capabilities. Specifically, we extract the global camera tokens $\mathbf{F} \in \mathbb{R}^{N \times C}$ from the backbone across the N input images. The base model dynamically identifies a reference view c_{ref}, from which we select its corresponding token $\mathbf{f}_{ref} \in \mathbb{R}^{1 \times C}$ to serve as the query. To effectively aggregate the global multi-view context, we employ a Context Transformer Decoder that takes $\mathbf{f}_{ref}$ as the query and the complete token sequence $\mathbf{F}$ as keys and values. This cross-attention mechanism allows the network to gather gravity-aware geometric clues from all views. The output feature is then processed by an MLP head to regress a continuous 6D rotation representation, ensuring stable optimization. This 6D representation is finally mapped into a 3×3 rotation matrix, denoting the rotation from the reference view to the GV frame, $\hat{\mathbf{R}}_{c_{ref} \to gv}$.

For supervision, we derive the ground-truth alignment matrix $\mathbf{R}^{gt}_{c\to gv}$ by leveraging artist-designed simulation datasets, where the global coordinate systems are inherently gravity-aligned. Assuming the gravity direction in the world coordinate system is defined as $\mathbf{g}_w = [0, 0, -1]^\top$, given the ground-truth camera extrinsics $\mathbf{R}_{w\to c}$, we first project the world gravity vector into the camera coordinate system to obtain the Gravity-View y-axis: $\mathbf{y} = \mathbf{R}_{w\to c}\mathbf{g}_w$. Following the aforementioned GV formulation, given the camera view direction $\mathbf{v} = [0, 0, 1]^\top$, the x-axis is computed as $\mathbf{x} = \frac{\mathbf{y}\times\mathbf{v}}{\|\mathbf{y}\times\mathbf{v}\|}$. To ensure robustness against collinear singularities (i.e., when the camera looks directly along the gravity axis), we fall back to the camera's original x-axis if the cross-product norm approaches zero. The z-axis is subsequently derived via $\mathbf{z} = \mathbf{x}\times\mathbf{y}$. Finally, the ground-truth rotation matrix is formulated as $\mathbf{R}^{gt}_{c\to gv} = [\mathbf{x}, \mathbf{y}, \mathbf{z}]^\top$.

During training, the module is optimized by applying an L_1 loss directly between the predicted rotation matrix of the reference view and its corresponding ground truth:

$$\mathcal{L}_{rot} = \|\hat{\mathbf{R}}_{c_{ref}\to gv} - \mathbf{R}^{gt}_{c_{ref}\to gv}\|_1 \tag{1}$$

Once $\hat{\mathbf{R}}_{c_{ref}\to gv}$ is accurately predicted, the entire reconstructed scene can be robustly transformed into the gravity-aligned coordinate system, facilitating subsequent physical simulations.

3.2 Target-Driven Object Generation and Layout Estimation

We decouple task-relevant objects through a lightweight target-selection interface. In the simplest mode, the target is specified by a single 2D box on a reference view; alternatively, the same box can be generated by a vision-language model or open-vocabulary grounding module without changing any downstream stage.

Given the bounding box, we first employ SAM-3 [6] to extract a precise 2D mask. This mask is then processed by SAM-3D [7] to lift 2D semantics into 3D space. We choose SAM-3D specifically for its amodal reconstruction capability, which recovers complete geometry even under partial occlusion. While SAM-3D inherently predicts an initial spatial layout alongside the generated mesh, our empirical evaluations reveal that this native pose estimation is often insufficiently accurate for strict physical simulations.

To refine the spatial layout, we integrate FoundationPose [46] as a dedicated module to compute the 6-DoF pose of the reconstructed mesh (Fig. 4(a)). By leveraging the gravity direction through our GV Alignment Module, we transform the placement into a physically grounded problem. Since SAM-3D meshes are inherently axis-aligned, a deterministic rotation first aligns their vertical axis with gravity. FoundationPose then only needs to estimate the translation and yaw angle, effectively reducing the rotational search space. Finally, the locally aligned mesh is transformed into the global Gravity-View space via $\mathbf{R}_{c_{ref}\to gv}$ for downstream simulation.

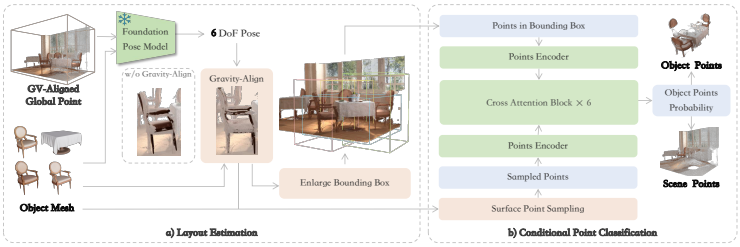

Fig. 4: Layout Estimation and background disentanglement in GARDEN. (a) Layout Estimation: Given a generated object mesh and the Gravity-View (GV) aligned global reconstruction, we first use FoundationPose [46] to estimate 6-DoF pose parameters and physically align with gravity direction. The *w/o Gravity Align* inset illustrates that omitting this alignment causes physically inconsistent placement. We then build a relaxed 3D region by gravity-aligned box expansion. **(b) Conditional Point Classification:** Cropped scene points within the box and mesh-derived condition points are fed into our conditional point classifier. The network predicts point-level labels to separate target-object from static scene points, removing duplicated geometry while preserving high-fidelity background.

3.3 Point-Conditional Background Disentanglement

To complete the physically-grounded factorization motivated in Sec. 1, placing an explicit object mesh is not sufficient by itself: we must also remove the duplicated object geometry that still remains in the original monolithic scene representation. Otherwise, the same object is represented twice, re-introducing foreground-background entanglement and causing visual and physical inconsistencies in downstream simulation. A naive 3D box crop is too coarse, because it often removes nearby background structures (e.g., supporting floors or occluding walls). Cross-view consistent 3D segmentation remains brittle, particularly in indoor scenes with repeated instances (e.g., similar chairs). Long-range association errors often cause incomplete or incorrect object removal.

To maintain scene-specific fidelity while achieving structural disentanglement, we formulate this task as a conditional 3D point classification problem, as illustrated in Fig. 4(b). We developed a Transformer-based point classification network that processes two distinct inputs: scene points cropped via a relaxed bounding box and condition points sampled from the generated object mesh. Both point sets are mapped into a shared feature space using coordinate and type-specific embeddings. A multi-layer Transformer architecture, utilizing self- and cross-attention mechanisms, then models fine-grained correspondences between the scene and the object. The network predicts a binary probability for each scene point to determine its membership to either the target object or the static background, guided by a standard Binary Cross-Entropy (BCE) loss during training.

To avoid manual point-level annotations on real scans, we leverage InternScenes [52], a large-scale simulation dataset providing paired scene and object meshes for precise label construction. We bridge the domain gap between

clean simulations and artifact-prone real reconstructions by introducing artifact-oriented 3D augmentations during training. These techniques, including multi-layer ghosting, thin-part deformation, virtual multi-view scanning, and jittering of mesh pose and scale, simulate systematic reconstruction errors. Training with these realistic perturbations enables robust simulation-to-reality transfer, ensuring reliable point-level disentanglement and cleaner object removal in real scenes while preserving background geometry. Detailed augmentation specifications are provided in the *Supplementary Material*.

3.4 Unified Physics Simulation

The final stage of our pipeline bridges the gap between static 3D reconstruction and interactive physical simulation. By leveraging the decoupled scene representation and the established Gravity-View coordinate system, we seamlessly construct a unified environment within a physics engine (e.g., MuJoCo [39]).

To physically ground the interactions, we first establish the static environment. Thanks to the gravity-aligned multi-view consistency, we identify the supporting floor plane by extracting the Top-K coordinate peaks along the gravity vector. This floor mesh is instantiated within the simulator as a static collision body. Concurrently, the user-specified objects—which have been reconstructed as complete local meshes and accurately localized via our layout estimation module—are imported as independent, dynamic rigid bodies.

A key advantage of our pipeline is the inherent physical consistency provided by the Gravity-View frame. Since all scene components (points, local meshes, and layout poses) share this unified coordinate system, we can directly align the simulator's gravity vector with the canonical vertical axis (e.g., the $+y$-axis).

Furthermore, we adopt a dual-representation strategy for the final environment: while the extracted floor and object meshes govern the rigid-body collision dynamics, the remaining cleaned background representation (such as the colored point cloud or 3DGS) is rendered as a high-fidelity visual backdrop. This design ensures that the generated environment not only strictly obeys physical laws for downstream embodied AI interactions but also preserves the photorealistic visual context of the original scene.

4 Experiments

Following the LiteReality [16] protocol, we evaluate GARDEN from three complementary perspectives: object-centric perceptual quality, holistic scene quality, and end-to-end computational efficiency. We further conduct two additional diagnostic experiments on gravity estimation accuracy and NVS-based factorization ablation.

4.1 Experimental Setup

Benchmarks and Evaluation Protocols We use three LiteReality evaluation settings: object-centric assessment, holistic scene assessment, and computational

efficiency. In addition, we evaluate gravity estimation accuracy and conduct an NVS-based ablation to isolate the effectiveness of our factorization design.

Object-Centric Assessment: We adopt the *object-centric material recovery* benchmark from LiteReality [16]. In the standard baseline protocol, methods are evaluated in a static setting using accurate cropped object bounding boxes. In contrast, we evaluate our method under a simulation-ready protocol: reconstructed assets are imported into MuJoCo [39], simulated for 10 seconds under gravity, and then rendered for metric computation using the user-provided query bounding box as ROI. This setting explicitly penalizes both appearance errors and physically induced pose/layout errors.

Holistic Scene Assessment: We adopt the *graphics-ready reconstruction* benchmark [16], which measures similarity between input RGB frames and rendered reconstructed scenes. Different from static evaluation, our scene-level results are also measured after the same 10-second MuJoCo [39] simulation. Therefore, floating, intersection, or gravity-inconsistent placement errors directly reduce the final perceptual scores.

Gravity Estimation Assessment: We evaluate the predicted gravity direction by angular error against ground truth on held-out Hypersim [34] and TartanAir [44] scenes. We report mean error, 90th percentile error, and the failure rate above 10°.

NVS Ablation Assessment: To validate our design choices, we perform NVS-based ablation on 5 Hypersim [34] scenes. We skip full-scene evaluation for point-cloud geometry. Given that NVS benchmarks provide limited viewpoints, under-observed regions degrade metrics and reduce method discriminability. We instead compute distances within an object-centric local region (including immediate support) where methods are comparably observable. This ablation jointly evaluates rendering quality and local reconstruction, revealing factorization effects beyond direct 6-DoF placement.

Computational Efficiency: We compare inference latency with LiteReality [16] on a workstation with an Intel Xeon w7-3445 CPU and an NVIDIA RTX A6000 GPU. We exclude evaluation-specific rendering overhead.

Baselines and Variants *Baselines:* For object-centric assessment, we compare against PhotoShape [32] and MIR [13], together with LiteReality [16]-related combinations and variants, including Visual+Language Search (Sem&Vis), MIR + Albedo-Only Optimization (AO), and the full LiteReality system. For holistic scene assessment, we compare against Phone2Proc [10] and ACDC [9], as well as ACDC + MIR, ACDC + Sem&Vis, and LiteReality. For gravity estimation, we compare against Plane-RANSAC, normal clustering, COLMAP's Manhattan alignment [36], GeoCalib [40], and GeoCalib+RANSAC. *Variants:* For NVS ablation, we compare **Ours** with representative variants: **DA3**, which directly uses the reconstruction backbone—DepthAnything3 [27] output; **w/ GV, full FP pose w/o sim**, which places generated objects via FoundationPose [46] without the complete physically-grounded integration; and two **w/o GV, full FP pose** variants that remove the learned Gravity-View alignment.

Table 1: Object-centric material recovery comparison. We follow LiteReality's [16] object benchmark and report RMSE/SSIM/LPIPS. Baselines are evaluated with the standard static cropped-bbox protocol. Our main results use the stricter post-MuJoCo [39] stabilization protocol, and static GARDEN results are included for protocol comparison. Best results are highlighted as **first**, second, third.

Methods	RMSE ↓	SSIM ↑	LPIPS ↓
MIR [13]	0.2377	0.3981	0.6111
PhotoShape [32]	0.3225	0.2371	0.6558
MIR [13] + AO [16]	0.2156	0.4203	0.5899
Sem&Vis [16]	0.2835	0.3758	0.6362
LiteReality [16]	0.2163	0.4353	0.5854
Ours (3DGS, post-sim)	0.1880	0.4181	0.5035
Ours (point cloud, post-sim)	0.1887	0.4240	0.4444
Ours (3DGS, static)	0.1796	0.4494	0.4182
Ours (point cloud, static)	**0.1736**	**0.4656**	**0.3680**

Implementation Details Our pipeline is implemented in PyTorch, and all trainable modules are optimized using AdamW. Training is conducted on 32 NVIDIA H20 GPUs. *Gravity-View Align Module:* We freeze the pre-trained DepthAnything3 [27] backbone and train the alignment module for 10k steps on TartanAir [44], Hypersim [34], and vKitti [5]. The peak learning rate is 5×10^{-6} with a 1k-step linear warm-up. The batch size is dynamically adjusted to keep an approximately constant token count per step. *Conditional Background Point Classification Module:* We train the classification network for 2k steps on pre-processed InternScenes [52] with a 5×10^{-4} base learning rate and batch size 48. *Inference:* Input multi-view images are resized to a maximum side length of 504 (following DepthAnything3 [27]) to recover camera poses, global point clouds, and 3DGS. For object generation, we cascade SAM-3 [6] (2D mask extraction) and SAM-3D [7] (amodal mesh reconstruction). FoundationPose [46] refines the layout with 5 iterations. In background disentanglement, we remove object-associated points using a 0.5 probability threshold. For simulation-backed evaluations, we use MuJoCo [39] with 0.001s timestep, 50-iteration Newton solver per step, and a pyramidal friction cone. The gravity vector $\mathbf{g}_{sim} = [0, 9.81, 0]^\top$ is set to align with our Gravity-View coordinate system.

4.2 Results

Object-Centric Assessment.. Tab. 1 shows that our method consistently improves over LiteReality [16] in RMSE and LPIPS, while keeping SSIM close. Specifically, under the stricter post-simulation protocol, Ours (3DGS) reduces RMSE from 0.2163 to 0.1880, and Ours (point cloud) reduces LPIPS from 0.5854 to 0.4444. The static GARDEN rows further improve these scores, confirming

Table 2: Holistic graphics-ready scene comparison. We report full-scene RMSE/SSIM/LPIPS between rendered reconstructions and input frames. Our main rendering is measured after physical simulation, directly testing gravity-consistent placement and global scene stability; static GARDEN results are included for protocol comparison. Best results are highlighted as **first**, second, third.

Methods	RMSE ↓	SSIM ↑	LPIPS ↓
Phone2Proc [10]	0.3604	0.5512	0.7338
ACDC [9]	0.3653	0.5531	0.7364
ACDC [9] + Sem&Vis [16]	0.3226	0.5425	0.6717
ACDC [9] + MIR [13]	0.3046	0.5492	0.6648
LiteReality [16]	0.2664	0.5818	0.6522
Ours (3DGS, post-sim)	0.1670	0.5573	0.4709
Ours (point cloud, post-sim)	0.1616	0.5735	0.4081
Ours (3DGS, static)	0.1593	0.5718	0.4379
Ours (point cloud, static)	**0.1511**	**0.5916**	**0.3770**

that the gains are not an artifact of post-simulation evaluation. Although LiteReality reports strong SSIM under its static protocol, the gap is small. We attribute this primarily to two factors: (1) unlike LiteReality, our pipeline does not include a dedicated surface-texture prediction stage; and (2) our post-simulation evaluation introduces minor physically induced pose perturbations, to which SSIM is sensitive. Overall, the stronger RMSE/LPIPS results indicate better geometry-aware and perceptual fidelity under a stricter simulation-ready protocol.

Holistic Scene Assessment.. Tab. 2 demonstrates consistent scene-level gains over prior methods. Compared with LiteReality [16], Ours (point cloud, post-sim) improves RMSE from 0.2664 to 0.1616 and LPIPS from 0.6522 to 0.4081, while maintaining comparable SSIM (0.5735 vs. 0.5818). The static rows again show stronger scores than the post-simulation rows, indicating that the post-simulation protocol is a conservative stress test rather than a source of artificial gains. This trend is consistent with the object-level findings: even without a dedicated texture prediction stage, our physically-grounded factorization provides stronger global scene fidelity and perceptual quality.

Gravity Estimation Accuracy Tab. 3 evaluates GV prediction accuracy against classical and learned gravity-estimation baselines. The proposed GV predictor achieves the lowest mean and P90 angular errors on both datasets, with 0.0% Fail@10. Plane-RANSAC and normal clustering can work when clean dominant planes are visible, but they are brittle under noisy or incomplete reconstructed geometry. GeoCalib [40] and COLMAP's Manhattan alignment [36] are stronger alternatives, yet they remain less accurate than our multi-view GV predictor.

Table 3: Gravity estimation accuracy. Angular error (degrees) on held-out Hypersim [34] and TartanAir [44] scenes.

Method	Hypersim			TartanAir		
	Mean ↓	P90 ↓	Fail@10 ↓	Mean ↓	P90 ↓	Fail@10 ↓
Plane-RANSAC	24.87	89.68	26.7	19.26	83.22	34.4
Normal clustering	30.98	89.77	33.3	21.33	88.70	37.5
COLMAP's Manhattan	2.62	6.38	**0.0**	8.37	15.91	12.5
GeoCalib [40]	7.59	4.79	6.7	3.01	6.50	9.4
GeoCalib+RANSAC	1.71	3.79	**0.0**	2.69	5.91	9.4
Ours GV	**1.40**	**1.90**	**0.0**	**1.56**	**3.48**	**0.0**

Table 4: NVS ablation on Hypersim [34]. We compare full GARDEN with DA3 and full-FP-pose variants with or without GV using rendering metrics (RMSE/SSIM/LPIPS) and point-cloud geometry metrics (Acc./Comp./N.C.). Point-cloud metrics are computed in an object-centric local region.

Methods	RMSE ↓	SSIM ↑	LPIPS ↓	Comp. ↓	Acc. ↓	N.C. ↑
3DGS Background						
DA3	0.3068	0.3141	0.6623	-	-	-
w/o GV, full FP pose w/o sim	0.3134	0.3000	0.6617	-	-	-
w/o GV, full FP pose w/ sim	0.3171	0.2885	0.6725	-	-	-
w/ GV, full FP pose w/o sim	0.3077	0.3154	0.6460	-	-	-
Ours	**0.2840**	**0.3809**	**0.5575**	-	-	-
Point Cloud Background						
DA3	0.2788	0.3482	0.5311	0.0516	**0.0082**	**0.9129**
w/o GV, full FP pose w/o sim	0.2961	0.3211	0.5917	0.0447	0.0726	0.7320
w/o GV, full FP pose w/ sim	0.3002	0.3096	0.5929	0.0743	0.3132	0.7078
w/ GV, full FP pose w/o sim	0.2753	0.3562	0.5351	0.0470	0.0674	0.7275
Ours	**0.2751**	**0.3565**	**0.5338**	**0.0387**	0.0658	0.7497

Ablation on Factorization and Placement Tab. 4 validates the effectiveness of our factorization design. Compared with DA3(point cloud), Ours(point cloud) achieves better completeness (Comp., 0.0387 vs. 0.0516), while improving RMSE and SSIM in rendering. Although DA3 obtains lower Acc. and higher N.C., its weaker completeness suggests that it tends to recover a more limited subset of target-object-region geometry. Our method reconstructs more complete local object-centric geometry while preserving competitive rendering quality. Compared with w/ GV, full FP pose w/o sim(point cloud), Ours(point cloud) is better across all reported metrics. Removing GV from the same full-FP-pose setting causes substantial degradation both without simulation and after simulation, confirming that GV improves pose initialization, background disentanglement, and downstream physical stability rather than serving as a cosmetic coordinate transform.

Fig. 5: Qualitative comparison with LiteReality [16]. Compared with LiteReality, GARDEN preserves scene-faithful background geometry and better maintains original object structure and placement.

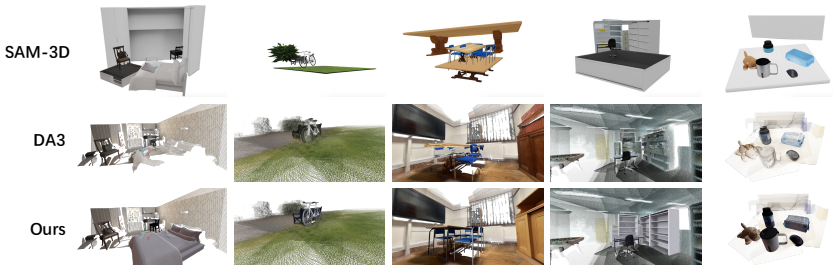

Fig. 6: Cross-dataset qualitative comparison against SAM3D [7] and DepthAnything3 [27]. Across Hypersim [34], Mip-NeRF360 [3], LiteReality [16], ETH3D [37], and our own tabletop captures, GARDEN produces more complete geometry, higher-quality object reconstruction, and more consistent object layout while preserving high-fidelity background context. Our results are from post-simulation states.

The 3DGS background directly uses DA3's GS head; its lower NVS quality in this ablation is mainly caused by low-opacity Gaussian primitives, while the fixed background still isolates the relative effect of our factorization components.

Qualitative Results We provide qualitative results under the same physically grounded protocol used throughout this paper: all visualizations are captured after a 10-second MuJoCo [39] simulation. This setting evaluates not only appearance quality but also gravity-consistent placement and structural stability.

Fig. 5 compares GARDEN with LiteReality [16] using the input view, LiteReality, and our two rendering variants (Ours (3DGS background) and Ours (point cloud background)). LiteReality tends to replace scene content with retrieved assets and does not preserve the original background context, which can also alter the native object structure. In contrast, our physically-grounded factorization keeps a pristine scene-faithful background while preserving scene-

Table 5: End-to-end inference latency. Runtime comparison between LiteReality [16] and GARDEN on a unified hardware platform. We exclude evaluation-time rendering overhead.

Scenes	LiteReality [16]	**Ours**
BoardRoom-CUED	7686s	**1244s**
Darwin-BedRoom	2781s	**321s**
Girton-Common-Room	4507s	**512s**
Girton-large-study-room	2308s	**318s**
Girton-Study-Room	4382s	**406s**
Mean	4332.8s	**560.2s**

specific objects and their layout, yielding more coherent and interaction-ready reconstructions in both background representations.

Fig. 6 further compares our method with DepthAnything3 [27] and SAM-3D [7] across diverse domains, including Hypersim [34], Mip-NeRF360 [3], LiteReality [16], ETH3D [37], and our own tabletop-object captures. DepthAnything3 fails to fully reconstruct key objects, whereas SAM-3D—constrained by single-view perspectives—struggles with occluded structures, generates incorrect object geometries, and produces unreliable layout estimates with poor background fidelity. Our method consistently recovers more complete geometry and higher-quality objects, while simultaneously preserving background fidelity and restoring physically plausible object layouts.

Computational Efficiency Tab. 5 shows that GARDEN achieves substantially lower inference latency than LiteReality [16] across all five scenes. The average runtime drops from 4332.8s to 560.2s (approximately 7.7× speedup), demonstrating practical efficiency gains for simulation-ready scene generation.

5 Conclusion

This paper introduced GARDEN, an RGB reconstruction framework that reconciles structural interactability with geometric fidelity via physically-grounded factorization. By leveraging gravity to resolve orientation ambiguity, our pipeline enables efficient user-prompted 6-DoF object anchoring and background disentanglement within a structured hybrid representation. GARDEN enhances both object and scene-level fidelity while achieving a 7.7× latency reduction compared to retrieval-based baselines. This formulation offers a practical, scalable framework for generating interactive digital twins in embodied AI.

Acknowledgements

This work is supported by the National Key R&D Program of China (No. 2024YFB3909903).

References

1. Ardelean, A., Özer, M., Egger, B.: Gen3dsr: Generalizable 3d scene reconstruction via divide and conquer from a single view. In: 2025 International Conference on 3D Vision (3DV). pp. 616–626. IEEE (2025)
2. Barron, J.T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P.P.: Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 5855–5864 (2021)
3. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5470–5479 (2022)
4. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Zip-nerf: Anti-aliased grid-based neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 19697–19705 (2023)
5. Cabon, Y., Murray, N., Humenberger, M.: Virtual kitti 2. arXiv preprint arXiv:2001.10773 (2020)
6. Carion, N., Gustafson, L., Hu, Y.T., Debnath, S., Hu, R., Suris, D., Ryali, C., Alwala, K.V., Khedr, H., Huang, A., Lei, J., Ma, T., Guo, B., Kalla, A., Marks, M., Greer, J., Wang, M., Sun, P., Rädle, R., Afouras, T., Mavroudi, E., Xu, K., Wu, T.H., Zhou, Y., Momeni, L., Hazra, R., Ding, S., Vaze, S., Porcher, F., Li, F., Li, S., Kamath, A., Cheng, H.K., Dollár, P., Ravi, N., Saenko, K., Zhang, P., Feichtenhofer, C.: Sam 3: Segment anything with concepts (2025), `https://arxiv.org/abs/2511.16719`
7. Chen, X., Chu, F.J., Gleize, P., Liang, K.J., Sax, A., Tang, H., Wang, W., Guo, M., Hardin, T., Li, X., et al.: Sam 3d: 3dfy anything in images. arXiv preprint arXiv:2511.16624 (2025)
8. Dahnert, M., Hou, J., Nießner, M., Dai, A.: Panoptic 3d scene reconstruction from a single rgb image. Advances in Neural Information Processing Systems **34**, 8282–8293 (2021)
9. Dai, T., Wong, J., Jiang, Y., Wang, C., Gokmen, C., Zhang, R., Wu, J., Fei-Fei, L.: Automated creation of digital cousins for robust policy learning. arXiv preprint arXiv:2410.07408 (2024)
10. Deitke, M., Hendrix, R., Farhadi, A., Ehsani, K., Kembhavi, A.: Phone2proc: Bringing robust robots into our chaotic world. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9665–9675 (2023)
11. Deitke, M., VanderBilt, E., Herrasti, A., Weihs, L., Ehsani, K., Salvador, J., Han, W., Kolve, E., Kembhavi, A., Mottaghi, R.: Procthor: Large-scale embodied ai using procedural generation. Advances in Neural Information Processing Systems **35**, 5982–5994 (2022)
12. Duan, J., Yu, S., Tan, H.L., Zhu, H., Tan, C.: A survey of embodied ai: From simulators to research tasks. IEEE Transactions on Emerging Topics in Computational Intelligence **6**(2), 230–244 (2022)
13. Fang, Y., Sun, Z., Wu, T., Wang, J., Liu, Z., Wetzstein, G., Lin, D.: Make-it-real: Unleashing large multimodal model for painting 3d objects with realistic materials. Advances in Neural Information Processing Systems **37**, 99262–99298 (2024)
14. Hou, J., Dai, A., Nießner, M.: Revealnet: Seeing behind objects in rgb-d scans. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2098–2107 (2020)

15. Huang, Z., Guo, Y.C., An, X., Yang, Y., Li, Y., Zou, Z.X., Liang, D., Liu, X., Cao, Y.P., Sheng, L.: Midi: Multi-instance diffusion for single image to 3d scene generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 23646–23657 (2025)
16. Huang, Z., Wu, X., Zhong, F., Zhao, H., Nießner, M., Lasenby, J.: Litereality: Graphic-ready 3d scene reconstruction from rgb-d scans. In: The Thirty-ninth Annual Conference on Neural Information Processing Systems
17. Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., et al.: Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In: Proceedings of the 24th annual ACM symposium on User interface software and technology. pp. 559–568 (2011)
18. Izadinia, H., Shan, Q., Seitz, S.M.: Im2cad. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5134–5143 (2017)
19. Kazhdan, M., Bolitho, M., Hoppe, H.: Poisson surface reconstruction. In: Proceedings of the fourth Eurographics symposium on Geometry processing. vol. 7 (2006)
20. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G., et al.: 3d gaussian splatting for real-time radiance field rendering. ACM Trans. Graph. **42**(4), 139–1 (2023)
21. Kolve, E., Mottaghi, R., Han, W., VanderBilt, E., Weihs, L., Herrasti, A., Deitke, M., Ehsani, K., Gordon, D., Zhu, Y., et al.: Ai2-thor: An interactive 3d environment for visual ai. arXiv preprint arXiv:1712.05474 (2017)
22. Kuo, W., Angelova, A., Lin, T.Y., Dai, A.: Mask2cad: 3d shape prediction by learning to segment and retrieve. In: European Conference on Computer Vision. pp. 260–277. Springer (2020)
23. Langer, F., Bae, G., Budvytis, I., Cipolla, R.: Sparc: Sparse render-and-compare for cad model alignment in a single rgb image. arXiv preprint arXiv:2210.01044 (2022)
24. Leroy, V., Cabon, Y., Revaud, J.: Grounding image matching in 3d with mast3r. In: European conference on computer vision. pp. 71–91. Springer (2024)
25. Li, C., Zhang, R., Wong, J., Gokmen, C., Srivastava, S., Martín-Martín, R., Wang, C., Levine, G., Lingelbach, M., Sun, J., et al.: Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation. In: Conference on Robot Learning. pp. 80–93. PMLR (2023)
26. Li, Z., Müller, T., Evans, A., Taylor, R.H., Unberath, M., Liu, M.Y., Lin, C.H.: Neuralangelo: High-fidelity neural surface reconstruction. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8456–8465 (2023)
27. Lin, H., Chen, S., Liew, J., Chen, D.Y., Li, Z., Shi, G., Feng, J., Kang, B.: Depth anything 3: Recovering the visual space from any views. arXiv preprint arXiv:2511.10647 (2025)
28. Makoviychuk, V., Wawrzyniak, L., Guo, Y., Lu, M., Storey, K., Macklin, M., Hoeller, D., Rudin, N., Allshire, A., Handa, A., et al.: Isaac gym: High performance gpu-based physics simulation for robot learning. arXiv preprint arXiv:2108.10470 (2021)
29. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021)
30. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM transactions on graphics (TOG) **41**(4), 1–15 (2022)

31. Nießner, M., Zollhöfer, M., Izadi, S., Stamminger, M.: Real-time 3d reconstruction at scale using voxel hashing. ACM Transactions on Graphics (ToG) **32**(6), 1–11 (2013)
32. Park, K., Rematas, K., Farhadi, A., Seitz, S.M.: Photoshape: Photorealistic materials for large-scale shape collections. arXiv preprint arXiv:1809.09761 (2018)
33. Puig, X., Undersander, E., Szot, A., Cote, M.D., Yang, T.Y., Partsey, R., Desai, R., Clegg, A.W., Hlavac, M., Min, S.Y., et al.: Habitat 3.0: A co-habitat for humans, avatars and robots. arXiv preprint arXiv:2310.13724 (2023)
34. Roberts, M., Ramapuram, J., Ranjan, A., Kumar, A., Bautista, M.A., Paczan, N., Webb, R., Susskind, J.M.: Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 10912–10922 (2021)
35. Runz, M., Li, K., Tang, M., Ma, L., Kong, C., Schmidt, T., Reid, I., Agapito, L., Straub, J., Lovegrove, S., et al.: Frodo: From detections to 3d objects. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14720–14729 (2020)
36. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4104–4113 (2016)
37. Schops, T., Schonberger, J.L., Galliani, S., Sattler, T., Schindler, K., Pollefeys, M., Geiger, A.: A multi-view stereo benchmark with high-resolution images and multi-camera videos. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3260–3269 (2017)
38. Shen, Z., Pi, H., Xia, Y., Cen, Z., Peng, S., Hu, Z., Bao, H., Hu, R., Zhou, X.: World-grounded human motion recovery via gravity-view coordinates. In: SIGGRAPH Asia 2024 Conference Papers. pp. 1–11 (2024)
39. Todorov, E., Erez, T., Tassa, Y.: Mujoco: A physics engine for model-based control. In: 2012 IEEE/RSJ international conference on intelligent robots and systems. pp. 5026–5033. IEEE (2012)
40. Veicht, A., Sarlin, P.E., Lindenberger, P., Pollefeys, M.: Geocalib: Learning single-image calibration with geometric optimization. In: European Conference on Computer Vision. pp. 1–20. Springer (2024)
41. Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotny, D.: Vggt: Visual geometry grounded transformer. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 5294–5306 (2025)
42. Wang, P., Liu, L., Liu, Y., Theobalt, C., Komura, T., Wang, W.: Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. arXiv preprint arXiv:2106.10689 (2021)
43. Wang, S., Leroy, V., Cabon, Y., Chidlovskii, B., Revaud, J.: Dust3r: Geometric 3d vision made easy. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 20697–20709 (2024)
44. Wang, W., Zhu, D., Wang, X., Hu, Y., Qiu, Y., Wang, C., Hu, Y., Kapoor, A., Scherer, S.: Tartanair: A dataset to push the limits of visual slam. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 4909–4916. IEEE (2020)
45. Wang, Y., Zhou, J., Zhu, H., Chang, W., Zhou, Y., Li, Z., Chen, J., Pang, J., Shen, C., He, T.: π^3: Permutation-Equivariant Visual Geometry Learning. arXiv preprint arXiv:2507.13347 (2025)
46. Wen, B., Yang, W., Kautz, J., Birchfield, S.: Foundationpose: Unified 6d pose estimation and tracking of novel objects. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 17868–17879 (2024)

47. Xia, H., Lin, C.H., Hsu, H.Y., Leboutet, Q., Gao, K., Paulitsch, M., Ummenhofer, B., Wang, S.: Holoscene: Simulation-ready interactive 3d worlds from a single video. Advances in Neural Information Processing Systems **38**, 32501–32524 (2026)
48. Yang, Y., Sun, F.Y., Weihs, L., VanderBilt, E., Herrasti, A., Han, W., Wu, J., Haber, N., Krishna, R., Liu, L., et al.: Holodeck: Language guided generation of 3d embodied ai environments. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16227–16237 (2024)
49. Yao, K., Zhang, L., Yan, X., Zeng, Y., Zhang, Q., Xu, L., Yang, W., Gu, J., Yu, J.: Cast: Component-aligned 3d scene reconstruction from an rgb image. ACM Transactions on Graphics (TOG) **44**(4), 1–19 (2025)
50. Yariv, L., Gu, J., Kasten, Y., Lipman, Y.: Volume rendering of neural implicit surfaces. Advances in neural information processing systems **34**, 4805–4815 (2021)
51. Yu, Z., Chen, A., Huang, B., Sattler, T., Geiger, A.: Mip-splatting: Alias-free 3d gaussian splatting. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 19447–19456 (2024)
52. Zhong, W., Cao, P., Jin, Y., Luo, L., Cai, W., Lin, J., Wang, H., Lyu, Z., Wang, T., Dai, B., et al.: Internscenes: A large-scale simulatable indoor scene dataset with realistic layouts. arXiv preprint arXiv:2509.10813 (2025)