DexMirror: Real-to-Sim Scene Mirroring for Sim-to-Real Dexterous Manipulation

Pu Hua*
Tsinghua University
Shanghai Qi Zhi Institute
hp24@mails.tsinghua.edu.cn

Zhecheng Yuan*
Tsinghua University
Shanghai Qi Zhi Institute
yuanzc23@mails.tsinghua.edu.cn

Meng Chu*
Tsinghua University
chumeng23@mails.tsinghua.edu.cn

Peiqi Duan*
Tsinghua University
dpq23@mails.tsinghua.edu.cn

Huazhe Xu
Tsinghua University
Shanghai Qi Zhi Institute
huazhe_xu@mail.tsinghua.edu.cn

Abstract

Dexterous manipulation requires large-scale robot interaction data, yet collecting real-world demonstrations is costly, while sim-to-real transfer remains challenging due to visual and geometric discrepancies. We present **DexMirror**, a unified real-to-sim-to-real framework that enables photorealistic simulation learning and reliable deployment for dexterous manipulation. Our method reconstructs real scenarios using a compositional 3D Gaussian Splatting (3DGS) representation with tri-level optimization, obtaining high-fidelity objects and background while preserving global scene consistency. A dexterous-hand-centric calibration pipeline, supported by an interactive online platform, further achieves millimeter-level robot–scene consistency for efficient Gaussian-simulation alignment. Built upon the reconstructed scenes, we train privileged policies in simulation and distill them into visuomotor policies operating on rendered 3DGS observations, combined with closed-loop execution for robust transfer. Experiments demonstrate accurate real-to-sim reconstruction (F1 score 0.981) and strong sim-to-real performance, achieving high success rates across six real-world dexterous manipulation tasks.

1 Introduction

Data is the fuel that powers modern AI, and robot data likewise forms a cornerstone of embodied intelligence [4, 3, 32, 49]. Among available data sources, teleoperation trajectories [32, 22, 47] are often regarded as the gold standard, as they provide high-fidelity demonstrations; however, this quality comes at a steep price: collection is slow, labor-intensive, and costly, making it difficult to scale to the diversity and volume required for general-purpose manipulation. At the other end of the spectrum lie human videos [57, 19, 20], which enable rapid acquisition at scale, yet remain only weakly grounded in robot embodiment, leading to indirect supervision while leaving the embodiment gap a major obstacle to behavior transferring. In contrast, simulation offers a compelling middle ground [3, 48, 15, 14]: despite the sim-to-real gap, simulation can leverage massive computation to

*Equal Contribution.

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



Figure 1: We propose **DexMirror**, a real-to-sim-to-real framework tailored for dexterous manipulation that reconstructs photorealistic 3D Gaussian scenes, achieves precise Gaussian–real-simulation consistency, and enables strong sim-to-real transfer with robust visual generalization.

synthesize large amounts of physically plausible robot data with rich state supervision and repeatable interaction. Consequently, narrowing the gap between simulation and reality has the potential to become one of the most scalable pathways toward high-quality robot data.

Most existing sim-to-real pipelines rely heavily on randomization or augmentation [55, 39, 1]. While effective to a degree, the considerable visual mismatch between simulation and reality still makes learning transferable policies highly challenging. Recent work has begun to address this issue by leveraging 3D Gaussian Splatting (3DGS) [21], a photorealistic 3D representation to reshape and improve simulation rendering [34, 18, 11, 13, 10, 17, 26]. However, these advances have largely been demonstrated only with parallel-jaw grippers. Extending such photorealistic sim-to-real rendering to dexterous hands, where high degrees of freedom and precise geometric modeling are essential, remains comparatively under-explored. Therefore, we propose DexMirror (as presented in Figure 1), a unified real-to-sim-to-real framework for dexterous manipulation that connects the real world with physically grounded simulation through photorealistic reconstruction and precise geometric calibration, enabling reliable sim-to-real transfer.

At a high level, DexMirror consists of two coupled stages. In the real-to-sim stage, we reconstruct manipulation environments using a compositional 3DGS representation that disentangles background, manipulated objects, and composed interactions through a tri-level optimization strategy. Unlike conventional pipelines requiring manual object or robot segmentation, our formulation automatically separates objects from the background during optimization, enabling scene acquisition with minimal human effort. This design preserves both object and scene fidelity while maintaining global consistency, allowing independent rigid-body manipulation in 3DGS. To ensure geometric consistency, we further introduce a dexterous-hand-centric calibration pipeline that achieves millimeter-level correspondence between reconstructed Gaussians, simulation, and real robot geometry. Supported by an interactive online platform, the calibration process is efficient and user-friendly, substantially reducing manual setup effort for reliable behavior transfer. In the sim-to-real stage, the reconstructed 3DGS serves as a photorealistic simulation interface for policy learning. We train an expert policy via demonstration-conditioned reinforcement learning with full state supervision, and subsequently distill it into a visuomotor policy with 3DGS observations. To further improve transfer robustness, we incorporate semantically guided lighting augmentation and a closed-loop leader–follower execution interface that stabilizes deployment under sensing and actuation uncertainties.

Extensive experiments across single-arm and bimanual dexterous manipulation demonstrate that DexMirror enables accurate real-to-sim reconstruction, precise robot–scene consistency, and reliable sim-to-real transfer across diverse contact-rich tasks. In the real-to-sim stage, our method achieves high-fidelity reconstruction and precise geometric consistency, reaching an alignment F1 score of 0.981. In the sim-to-real stage, policies trained in simulation achieve an average real-world task success rate of 75.2%. The policies remain robust under spatial and visual variations, highlighting a pathway toward data-efficient real-to-sim-to-real learning for dexterous manipulation.

In summary, our main contributions are three-fold: (1) We present a unified 3DGS-based real-to-sim-to-real framework for dexterous manipulation, enabling photorealistic simulation and effective

policy transfer. (2) We propose a compositional real-to-sim reconstruction and calibration pipeline with tri-level Gaussian optimization and a user-friendly online platform, enabling efficient scene setup and precise robot–hand correspondence. (3) We introduce a sim-to-real learning strategy that learns RGB-based policies from a single demonstration and transfers them to diverse tasks, including non-prehensile manipulation, bimanual coordination, and contact-rich dexterous manipulation.

2 Related Work

Scene and Object Reconstruction. Scene reconstruction has traditionally relied on structure-from-motion (SfM) [40] and multi-view stereo (MVS) [12] pipelines to recover explicit scene geometry, followed by neural rendering approaches such as NeRF [30] that greatly improve photorealism through volumetric modeling. Recently, 3D Gaussian Splatting (3DGS) and its variants [21, 16] have emerged as a state-of-the-art solution, enabling fast training and real-time rendering while maintaining high visual fidelity. Its explicit, point-based representation is naturally compatible with simulation systems, making it attractive for real-to-sim applications. For object modeling, besides reconstruction-based methods such as NeRF and 3DGS, recent AI-driven 3D generation approaches [51, 50, 44, 43, 42, 41] can synthesize objects directly from images, but they often lack accurate geometry and physical consistency required for manipulation. In this work, we adopt a compositional 3DGS-based reconstruction framework to jointly model scenes and objects with precise and consistent geometry.

Real-to-Sim Pipeline. Recent work explores using real-to-sim as an evaluation and training interface for robot learning [23, 17, 18], where reconstructed environments serve as realistic testing grounds for manipulation policies. A parallel line of research leverages 3D reconstruction techniques to generate more faithful object or scene meshes [10, 9, 45, 6, 27], improving visual realism when rendered in simulation. Most closely related to our work are approaches that replace the rendering backend of simulators with 3DGS while retaining the underlying physics engines. By combining the visual fidelity of 3DGS with the physical consistency of simulators, these systems enable physically grounded yet photorealistic simulation. However, existing methods mainly focus on parallel grippers and relatively simple manipulation settings. In contrast, our work targets dexterous manipulation, which demands higher accuracy in scene reconstruction and robot–scene correspondence due to the fine-grained contacts and complex kinematics involved.

Sim-to-Real for Dexterous Manipulation. A common approach trains visuomotor policies directly in simulation using RL [39, 25], but these methods are often limited to relatively simple tasks or require carefully designed rewards and extensive tuning. Existing sim-to-real methods often rely on heavy visual augmentation of simulated observations [55] or instead use modalities with a smaller sim–real gap [24, 5, 56], such as depth or estimated object state. In contrast, our work adopts a real-to-sim-to-real pipeline that reconstructs photorealistic manipulation environments from the real world, enabling policy learning in simulation while improving the reliability of sim-to-real for dexterous manipulation.

3 Method: Real-to-Sim Reconstruction

3.1 Compositional 3D Gaussian Splatting Pipeline

To support manipulation-oriented reconstruction, we adopt a compositional 3DGS pipeline that models the background and manipulated objects separately while preserving global scene consistency.

Data Collection and Preprocessing. Original 3DGS typically begins with the acquisition of a multi-view image dataset of a static scene. To enable separate modeling of the background and manipulated objects, we instead capture two distinct datasets for joint 3DGS optimization: a *background* dataset containing the empty scene, and a *composed* dataset in which the objects are placed within the scene. We capture views along a 360° trajectory around the workspace and include close-ups near the robot hand to improve coverage under occlusion. Following the standard 3DGS pipeline, we preprocess both datasets using COLMAP [36, 37] to obtain camera poses and geometric initialization for training. In addition, we apply Depth-Anything [54] to estimate depth maps for both datasets, which are further incorporated as depth regularization during optimization. Subsequently, we employ Grounded-SAM [35] to generate binary segmentation masks for each object, enabling separation of

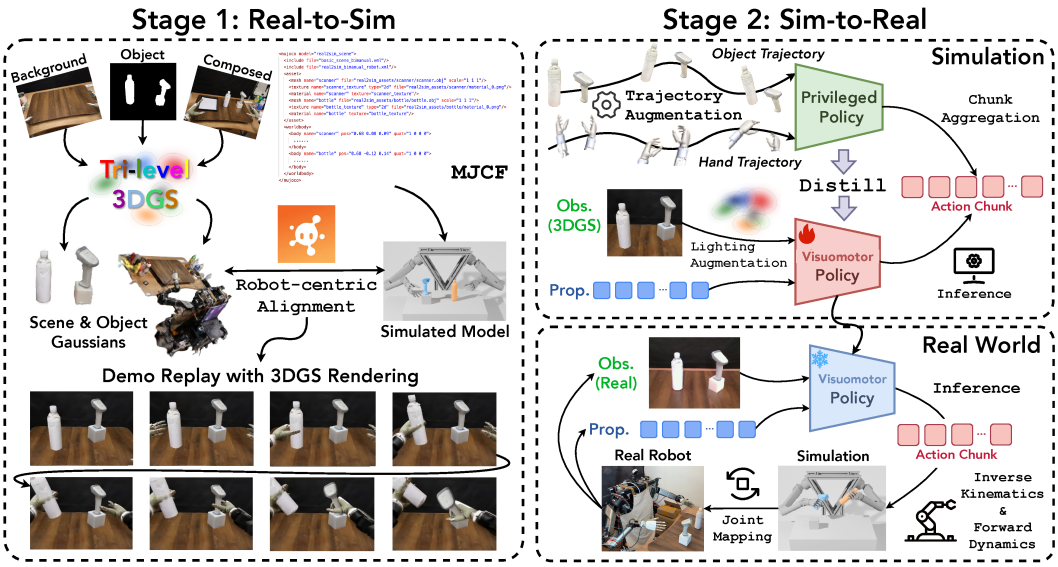


Figure 2: **Overview of DexMirror framework.** DexMirror consists of two stages. In the real-to-sim stage, we employ a tri-level optimization scheme for photorealistic scene reconstruction and build a lightweight calibration platform. In the sim-to-real stage, we introduce trajectory augmentation for demonstration-conditioned RL and use visual distillation to learn a robust real-world policy.

object regions, which we denote as the *object* dataset. In parallel, we use AR Code to extract object meshes for MuJoCo. Further details are provided in Section A.1.

Tri-Level 3DGS Reconstruction. Unlike the original 3DGS formulation, which relies on a single rendering objective tailored to static scenes, our setting necessitates the joint optimization of the background together with multiple object models, while maintaining spatial consistency during scene composition. To this end, we establish a *tri-level optimization framework*, consisting of background-only rendering, object-isolated rendering, and composed-scene rendering objectives. Specifically, we allocate and train an individual Gaussian model for the background and each object. The composed scene is rendered by simply concatenating all Gaussian models into a unified representation. The original 3DGS supports incorporation of depth regularization besides a photometric loss, leading to its optimization objective:

$$\mathcal{L}_{\text{GS}} = (1 - \lambda_{\text{dssim}}) \cdot \mathcal{L}_{\text{L1}} + \lambda_{\text{dssim}} \cdot \big(1 - \text{SSIM}(\hat{\mathbf{I}}_{\text{rec}}, \mathbf{I}_{\text{gt}})\big) + \mathcal{L}_{\text{depth}}$$

During the tri-level optimization phase, we extend each component in $\mathcal{L}_{\text{GS}}$ by aggregating supervision signals from the three rendering levels. As an example, the original L1 loss is replaced with a weighted formulation $\mathcal{L}_{\text{L1}}^{\text{weighted}}$:

$$\mathcal{L}_{\text{L1}}^{\text{weighted}} = \frac{\lambda_{\text{bg}}\mathcal{L}_{\text{bg}} + \lambda_{\text{comp}}\mathcal{L}_{\text{comp}} + \lambda_{\text{obj}} \sum_{i=1}^{N} \mathcal{L}_{\text{obj}}^{(i)}}{\lambda_{\text{bg}} + \lambda_{\text{comp}} + N \cdot \lambda_{\text{obj}}},$$

where $\mathcal{L}_{\text{bg(comp)}} = \ell_1\big(\hat{\mathbf{I}}_{\text{bg(comp)}}, \mathbf{I}_{\text{gt}}^{\text{bg(comp)}}\big)$ supervise reconstruction of the background (composed) scene. For each object i and its mask $\mathbf{M}^{(i)}$, $\mathcal{L}_{\text{obj}}^{(i)} = \ell_1(\hat{\mathbf{I}}_{\text{obj}}^{(i)} \odot \mathbf{M}^{(i)}, \mathbf{I}_{\text{gt}}^{(i)} \odot \mathbf{M}^{(i)})$ optimizes fine-grained object appearance via masked rendering. The SSIM and depth regularization terms are extended analogously under the same tri-level aggregation scheme. In practice, we set $\lambda_{\text{bg}} = 1.0$, $\lambda_{\text{comp}} = 0.1$, and $\lambda_{\text{obj}} = 10.0$, prioritizing object-level optimization while maintaining global composition consistency. For densification, we operate on each Gaussian model independently rather than on the composed scene, using gradients from background and object renderings to update their respective models, thereby preventing their Gaussians from being mixed during densification.

3.2 Robot-Centric Scene Calibration

Following reconstruction, we establish geometric consistency between the reconstructed Gaussians and the robot system through a robot-centric calibration process supported by interactive annotation and kinematic-aware segmentation.

Hand-Centric Alignment. Given the 3DGS reconstruction of the scene and objects, another crucial step is to establish consistent scale and coordinate alignment between the 3D Gaussians and the simulated environment. Beyond assuming that the simulation and the real scene share the same setting, prior works often rely on visual fiducial markers such as ArUco and registration methods such as Iterative Closest Point (ICP) [2] to recover metric scale and camera extrinsics, and thereby estimate the relative transformation between 3DGS and simulation. However, because the dexterous hand contains far fewer points than the arm in the 3D Gaussians, ICP tends to favor the arm, leading to larger errors on the hand and causing points from one link to be mismatched to adjacent links. To address this limitation, we propose a *dexterous-hand-centric* keypoint-based registration strategy. We annotate approximately 10 corresponding keypoints per arm on the robot model and the 3DGS point cloud, covering the fingertips, the base of the palm, and 3–4 arm reference points to provide stable and task-relevant anchors. Given paired point sets in 3DGS and simulation, we estimate a similarity transformation $[c, \mathbf{R}, \mathbf{t}] \in \mathrm{SIM}(3)$ that aligns them. This is solved in closed form using the Umeyama algorithm [46] via SVD.

Online Platform. To facilitate accurate and fast annotation, we build an online platform (shown in Figure 7) upon SuperSplat [33]. The interface allows users to interactively explore the reconstructed scene point cloud while simultaneously rendering the corresponding Gaussian splatting image from the same viewpoint. By leveraging the photorealistic visual cues provided by the 3DGS, users can precisely localize and select semantically meaningful keypoints in the point cloud, thereby obtaining reliable point correspondence for registration.

Kinematic Segmentation. After establishing a consistent coordinate frame between the reconstructed Gaussian splats and the simulated robot model, we segment the 3D Gaussians associated with each kinematic link of the robot arm and hand. We first perform a coarse segmentation by identifying Gaussians that lie within the mesh volume of each link using Trimesh [29]. Building on this initial partition, we apply a K-Nearest Neighbour (KNN) [8] classifier with a distance threshold to assign the remaining segmented Gaussians. Points that exceed this threshold are treated as part of the static scene rather than the robot body.

4 Method: Sim-to-Real Transfer

In this section, we present how DexMirror can facilitate policy learning and sim-to-real transfer, which operates on the 3DGS reconstruction of the real-world scene and objects proposed in Section 3.

4.1 Reinforcement Learning for Expert Policy Acquisition

Training Paradigm. In general, we follow [56] to train the expert policy, which utilizes DrM [52], an off-policy algorithm incorporating a dormant ratio mechanism to promote effective exploration. The privileged expert interacts with the MuJoCo assets constructed during real-to-sim, with physical parameters randomized around VLM-inferred priors; the full asset-construction procedure is detailed in Section A.1. Moreover, a generalizable reward formulation is combined with a single reference demonstration to mitigate the burden of task-specific reward engineering. The reward function integrates object- and hand-level trajectory tracking, contact consistency, and energy penalty to guide reinforcement learning. For policy architecture, we adopt a residual action learning strategy that combines coarse guidance from reference with learned fine-grained adjustments for precise hand-object interactions.

Novel Demonstration Synthesis. As the scene is randomized at the beginning of each training episode, the reward terms conditioned on privileged states from the reference demonstration will become misaligned with a new object configuration. To address this, we introduce a demonstration augmentation procedure that synthesizes a consistent demonstration aligned with the randomized scene, derived from the original reference trajectory captured under a fixed initial pose. We decompose the demonstration into two categories of motion clips: object-interactive motions, which establish functional contact and induce state changes in task-relevant objects (e.g., grasping), and object-agnostic motions, which serve as geometric transitions for spatial alignment in which the objects remain stationary or are held in a stable grasp. Suppose the object is initialized with a random transformation $\mathbf{T}^{\text{trans}} \in \mathrm{SE}(3)$. Let $\mathbf{T}_t^h$ and $\mathbf{T}_t^o$ denote the hand and object poses at time step t in a motion clip. We handle these two different categories of motion clips as follows:

Object-interactive motion. We preserve the relative hand-object pose at each time step and apply $\mathbf{T}^{\text{trans}}$ jointly to both the hand and object trajectories: $\mathbf{T}_t^{h\prime} = \mathbf{T}^{\text{trans}} \cdot \mathbf{T}_t^h$, $\mathbf{T}_t^{o\prime} = \mathbf{T}^{\text{trans}} \cdot \mathbf{T}_t^o$, thereby precisely reproducing the demonstrated hand–object interactions.

Object-agnostic motion. We compute the transformed starting and ending hand poses $\mathbf{T}_0^{h\prime}$ and $\mathbf{T}_T^{h\prime}$ under the randomized object configuration and interpolate between them to synthesize a new hand trajectory: $\mathbf{T}_t^{h\prime} = \mathrm{Interp}\left(\mathbf{T}_0^{h\prime},\ \mathbf{T}_T^{h\prime},\ t/T\right)$, where T is the clip horizon and $\mathrm{Interp}(\cdot)$ denotes a pose interpolation operator. Finally, when the object remains stationary, we simply replicate $\mathbf{T}_t^{o\prime} = \mathbf{T}_0^{o\prime}$ for all t across the clip. When the object is stably grasped, we keep the hand-object pose and propagate the object pose from the new hand motion: $\mathbf{T}_t^{o\prime} = \mathbf{T}_t^{h\prime} \cdot \mathbf{T}^{ho}$, where $\mathbf{T}^{ho} = (\mathbf{T}_0^h)^{-1} \cdot \mathbf{T}_0^o$.

Meanwhile, given the original and generated hand and object trajectories, we update the corresponding hand–object distance profiles required by the reward formulation. Moreover, the hand joint positions and contact states remain unchanged, as the underlying hand–object interactions are preserved throughout the transformation. An example schematic diagram is provided in the appendix. Our demonstration synthesis enables *object-level independent randomization* while preserving full hand–object interaction information, including contact patterns and relative motion. In contrast, prior approaches either apply a *scene-level global randomization* strategy, sharing the same transformation to all objects [56], or are limited to augmenting object and arm poses only [53], or rely on simulation replaying to acquire novel demonstrations [28].

4.2 Visuomotor Distillation via Gaussian Splatting Rendering

We further distill the low-level expert into a vision-based policy by training on Gaussian splatting renderings of the reconstructed scene. To bridge the representation gap between simulation and real-world observations and enhance robustness to visual variability, we additionally apply lighting augmentation to simulate diverse illumination conditions.

Photorealistic Simulation. Rather than rendering the simulated environment directly, we employ the 3D Gaussian representations of the scene and objects to yield photorealistic observations, while contacts and dynamics are computed from the MuJoCo assets used by the privileged expert. The robot links and objects are treated as independent rigid bodies, for which the corresponding Gaussian models have been obtained during real-to-sim reconstruction. For each rigid body, we can obtain the relative transformation between consecutive frames from the simulator or via robot forward kinematics, parameterized by a translation $\mathbf{t}$ and a rotation $\mathbf{R}$. Using this transformation, we update the geometric attributes by: $\boldsymbol{\mu}' = \mathbf{R}\boldsymbol{\mu} + \mathbf{t}, \boldsymbol{\Sigma}' = \mathbf{R}\boldsymbol{\Sigma}\mathbf{R}^T$, where $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and $(\boldsymbol{\mu}', \boldsymbol{\Sigma}')$ are current and transformed Gaussian means and covariances, respectively.

Lighting Augmentation. Despite the high visual fidelity of Gaussian splatting–based rendering, discrepancies in lighting conditions may still arise due to variations in real-world illumination or changes in camera parameters. To mitigate this effect, we augment the diffuse color of each Gaussian in the reconstructed scene via random offsetting and additive noise. Concretely, given an original color vector $\mathbf{c} = (r, g, b)$, the augmented color $\tilde{\mathbf{c}}$ is obtained as: $\tilde{\mathbf{c}} = \mathbf{c} + \boldsymbol{\beta}_{\text{scene}} + \boldsymbol{\beta}_{\text{object}} + \boldsymbol{\epsilon}_{\text{point}}$, where $\boldsymbol{\beta}_{\text{scene}}$ and $\boldsymbol{\beta}_{\text{object}}$ denote the scene-level and object-level color bias terms shared across all points within the same scene and rigid body, respectively, and $\boldsymbol{\epsilon}_{\text{point}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ is a point-wise Gaussian noise term accounting for local color perturbations. In contrast to pixel-level 2D augmentations (e.g., color jittering), our method enables semantic-level augmentation for both scenes and objects. By leveraging explicit, consistent labels, it preserves global structure while introducing local variability.

Policy Architecture. We employ Diffusion Policy[7] as the policy backbone. Specifically, RGB observations are encoded using a DINOv2 [31] visual encoder, while proprioceptions are embedded via a multi-layer perceptron (MLP) to form an observation representation. This latent representation is then passed to a diffusion action head, which generates an action chunk as end-effector poses.

Sim-to-Real Transfer. To reduce the disparity between simulation and real-world deployment, we adopt a leader-follower control strategy that functions as a physics-level sim-real alignment mechanism, visualized in Figure 2. Within this closed-loop process, real-world observations and proprioceptions are first provided to the distilled policy to infer end-effector poses as actions, which are then executed in the simulator to perform inverse kinematics and further forward dynamics steps. The resulting joint states of the simulated robot are subsequently transmitted to the real robot for execution, after which the updated visual and proprioceptive observations are captured and used as input for the next inference loop. By synchronizing the inverse kinematics solver and consistent

Table 1: **Ablation study of tri-level 3DGS optimization.**

Scene	**Franka + LEAP**				**Galaxea + OYMotion**			
Type	Background		Objects		Background		Objects	
Criteria	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Ours	**30.16**	**0.9540**	**40.88**	**0.9964**	**26.25**	**0.9334**	**39.04**	**0.9944**
w/o *composed*	29.50	0.9511	39.80	0.9960	25.88	0.9205	37.92	0.9939
w/o *objects*	29.85	0.9538	30.26	0.9379	26.02	0.9274	25.58	0.9003
w/o *background*	17.46	0.7578	40.22	0.9964	15.91	0.7571	38.23	0.9941
3DGS	19.09	0.7930	39.97	0.9925	17.77	0.7750	34.86	0.9713

system dynamics across both simulation and the real world, this framework maintains a coherent internal simulation model while allowing the policy to respond to real-world observations, thereby effectively reducing the sim-to-real discrepancy.

5 Experiments

5.1 Experimental Setup

Hardware Platforms. Although DexMirror is designed to generalize to arbitrary combinations of dexterous hands and robotic arms, we evaluate it primarily on two representative platforms. The first is a single-arm setting consisting of a 7-DoF Franka Emika Panda manipulator paired with a 16-DoF LEAP Hand[38], a widely adopted setup in dexterous manipulation research. The second is a bimanual setting composed of two 6-DoF Galaxea A1 arms equipped with two OYMotion 6-DoF hands, enabling the execution of more complex and highly dexterous manipulation tasks. Moreover, we employ Intel L515 cameras to capture RGB observations. In the single-arm setting, a third-person camera is used, while in the bimanual setup, a head-mounted camera provides an egocentric view.

Task Design. We conduct our experiments on six dexterous manipulation tasks, as depicted in Figure 5, including three single-arm tasks and three bimanual tasks: (1) *PourBeer*: Franka grasps the beer can and pours the beer into a cup; (2) *ServeCoffee*: Franka holds the coffee pot and pours coffee into a filter funnel; (3) *HangMug*: Franka picks up a mug and hangs it on a rack by its handle; (4) *BottleHandover*: Bimanual Galaxea picks up a bottle, performs a right-to-left handover, and finally places it into a box; (5) *ScanBottle*: Bimanual Galaxea grasps a handheld scanner and uses it to scan the bottom of a bottle; (6) *PlateGrasp*: Bimanual Galaxea braces a plate against a wall before it executes a successful grasp, ***which is a non-prehensile manipulation task***. For the single-arm scene, our tri-level reconstruction takes approximately 27 minutes and interactive calibration takes approximately 4 minutes; the full timing breakdown is given in Section B.4.

5.2 Real-to-Sim Evaluation

Ablation on Tri-Level Optimization. We conduct ablation studies to evaluate the contribution of each supervision level in the proposed tri-level optimization framework. We also add a standard 3DGS pipeline as an extreme ablation without tri-level supervision. Results are summarized in Table 1. Removing composed-scene supervision degrades reconstruction, while removing object-level supervision significantly harms object quality, highlighting the necessity of both global and object-level supervision. Removing background supervision severely overfits to the composed configuration. The original 3DGS yields much weaker background reconstruction due to entangled scene and object optimization. Our tri-level formulation elegantly captures background and objects as independently movable entities, while maintaining faithful geometric consistency after composition. To test whether this reconstruction gain translates to real-world manipulation, we replace only the reconstruction backend with vanilla 3DGS while keeping calibration, policy architecture, and training budget fixed. As shown in Table 2, tri-level reconstruction improves success on all three Franka tasks. Vanilla 3DGS entangles objects and background, leaving cropped holes and baked-in shadows after object relocation. These artifacts matter more when transparent objects must be localized in *ServeCoffee* or precise contacts must be made in *HangMug*. By changing only the reconstruction backend, this comparison attributes the downstream gain to the compositional representation rather than to calibration or policy training.

Table 2: **Downstream real-world success under different reconstruction backends.** Calibration, policy, and training budget are held fixed.

Reconstruction backend	PourBeer	ServeCoffee	HangMug
Vanilla 3DGS	0.89	0.66	0.56
Ours (tri-level 3DGS)	**0.93**	**0.80**	**0.68**

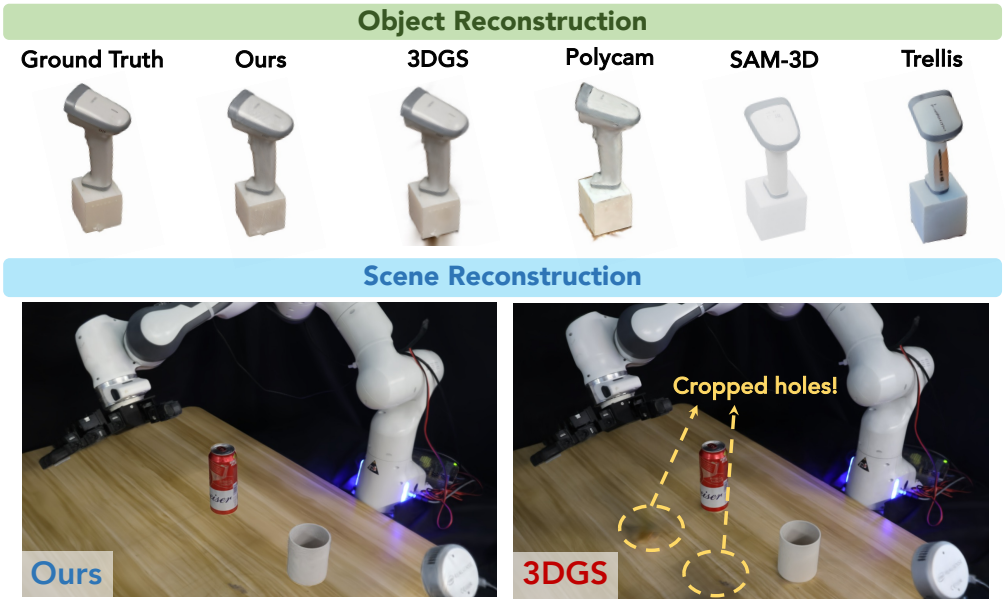


Figure 3: **Reconstruction comparison with baselines.** *Top*: object reconstruction results. *Bottom*: scene reconstruction results after object relocation.

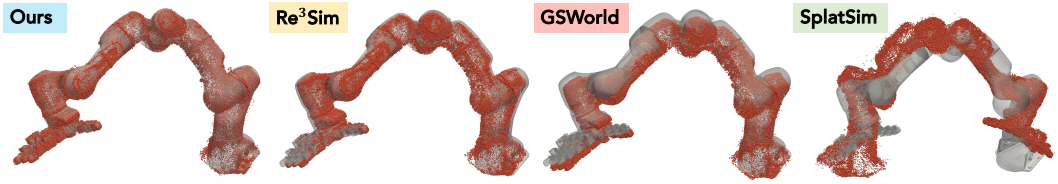


Figure 4: **Visualization of robot alignment results via different methods.** F-scores of robot-level and hand-level alignment across different methods are annotated.

Evaluation on Reconstruction Quality. As depicted in Figure 3, we evaluate the reconstruction quality of both the manipulated objects and the surrounding scene against several baselines:

- *Object reconstruction.* We compare DexMirror with a standard 3DGS pipeline trained on the composed scene with objects extracted via segmentation masks, the Polycam 3D scanning application, and two off-the-shelf image-to-3D generation methods, SAM-3D [41] and Trellis [51] to model a scanner. Our method reconstructs objects with high fidelity to the ground truth. In comparison, objects generated directly by 3DGS often retain residual artifacts due to incomplete separation, such as table colors and shadows. Polycam produces comparatively less precise reconstructions, while SAM-3D and Trellis, despite generating visually detailed 3D Gaussians, struggle to accurately preserve color, scale, and intra-object spatial consistency.
- *Scene reconstruction.* We compare our method with the original 3DGS in the Franka environment. Specifically, objects are relocated beyond their initial positions, and the rendering quality under the novel configuration is evaluated to assess reconstruction consistency. We find that in 3DGS scenes, moving the objects leads to cropped holes at their original locations. In contrast, our method allows objects to be freely repositioned while effectively isolating artifacts such as shadows.

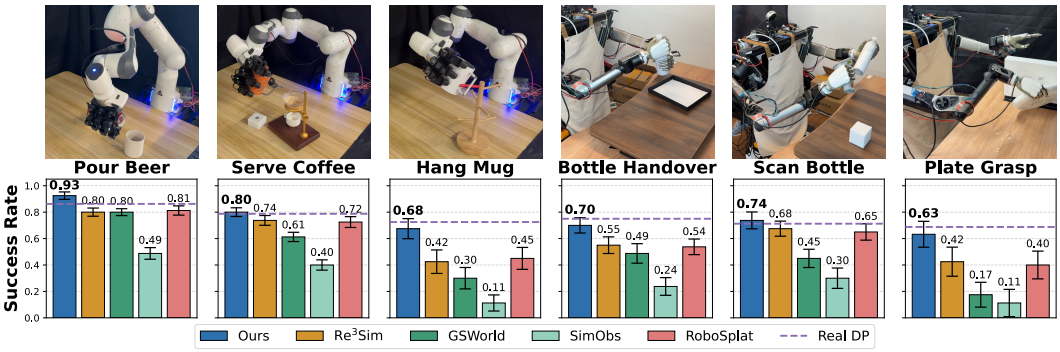


Figure 5: **Real-world experiment results.** DexMirror achieves 25% relative improvement across multiple types of tasks. All experiments are based on 5 seeds and 16 trials per seed.

Evaluation on Gaussian-Simulation Alignment. We compare our method with baseline approaches in terms of the alignment between the 3D Gaussians and the simulation in the Franka environment. Existing methods typically rely on ArUco markers and ICP-based registration for alignment. Accordingly, we implement three representative baselines:

- *GSWorld* [18]: Camera extrinsics estimated from ArUco marker detection are combined with COLMAP poses to compute a SIM(3) transformation for alignment.
- *SplatSim* [34]: ICP is directly applied for geometric alignment.
- *Re^{3}Sim* [13]: Marker-based alignment is employed to obtain a coarse initialization, followed by ICP refinement for fine-grained alignment.

Table 3: **Comparison of different methods on alignment using F-score.**

Method	Hand	Robot
Ours	**0.981**	**0.942**
Re^{3}Sim	0.864	0.882
GSWorld	0.436	0.784
SplatSim	—	0.433

The results are demonstrated in Figure 4. We observe that our method achieves the most precise alignment, particularly in the hand region. *GSWorld* provides only a relatively coarse alignment, while *Re^{3}Sim* improves accuracy to some extent with ICP refinement . In contrast, using ICP alone results in significant errors, as ICP is not well-suited for registering complex shapes and is highly sensitive to initialization. Moreover, to quantitatively evaluate the geometric consistency , we compute the F-score under a distance threshold τ (shown in Table 3). Let G denotes the point cloud from 3D Gaussians, and S denotes the point cloud obtained from the robot model. Precision, recall, and F-score are defined as:

$$P = \frac{1}{|G|}\sum_{g\in G}\mathbf{1}[\min_{s\in S}\|g-s\|_2 < \tau], R = \frac{1}{|S|}\sum_{s\in S}\mathbf{1}[\min_{g\in G}\|s-g\|_2 < \tau], F = \frac{2PR}{P+R}.$$

Specifically, we compute the F-score for both the whole robot and the hand separately, showing that our method outperforms the baselines in both cases. To assess manual annotation sensitivity, we independently perturb the selected keypoints with Gaussian noise and re-estimate the SIM(3) alignment using Umeyama's closed-form solution. At a distance threshold of $\tau = 0.015$, the hand F-score remains 0.972 and 0.936 under 2 mm and 5 mm keypoint perturbations, respectively; five independent annotators obtain 0.975 ± 0.007. These results indicate that the multiple correspondences absorb moderate click errors without repeated manual refinement. Full calibration details and noise-sensitivity results are provided in Section B.3.

5.3 Sim-to-Real Evaluation

Comparison with Baselines. We compare DexMirror with several baselines, including methods using reconstructions from *Re^{3}Sim* and *GSWorld*, and a policy trained in simulation with random overlay augmentation on observations, denoted as *SimObs*. We further include two non-sim-to-real references: *RoboSplat* [18] augments the reconstructed reference demo directly without simulation, and a diffusion policy trained on an equal amount (100 episodes) of real data, denoted as *Real DP*,

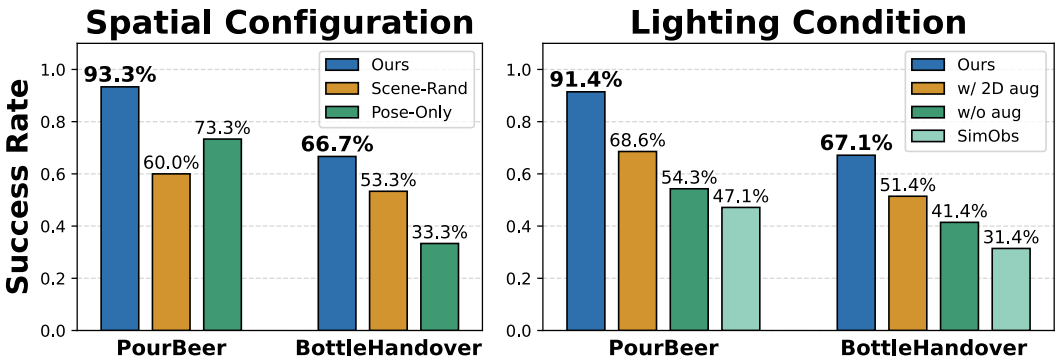


Figure 6: **Generalization under spatial and lighting variations.**. Our method shows improved robustness across both settings. *Left*: spatial configuration; *Right*: lighting conditions.

serving as an oracle. As shown in Figure 5, DexMirror consistently outperforms all baselines across six dexterous manipulation tasks, achieving an average success rate of 75.2%. In contrast, baselines suffer from inaccurate geometric correspondence, which becomes evident in contact-rich tasks such as *HangMug* and *PlateGrasp*. *SimObs* exhibits the weakest performance due to the substantial visual and geometric discrepancy between the simulation and the real world. These results show that precise reconstruction and consistent modeling are critical for reliable transfer. Moreover, compared to *RoboSplat*, our method benefits from RL training with physics-aware simulation, enabling more physically consistent data generation beyond direct demo augmentation. We also observe performance comparable to *Real DP*, while being more scalable than collecting real-world data, thanks to the data amplification and augmentation enabled in simulation (see B.4 for human effort analysis).

Generalization. We further conduct generalization tests and ablation studies of our method with respect to spatial configurations and lighting conditions on tasks *PourBeer* and *BottleHandover*.

Spatial Configuration. We compare our proposed object-level independent randomization (*Ours*) with the scene-level global randomization strategy (*Scene-Rand*) adopted in [56], as well as a variant that only augments object and hand poses without preserving hand–object interactions (*Pose-Only*), to evaluate spatial generalization. As shown in Figure 6 left, both *Scene-Rand* and *Pose-Only* exhibit noticeable degradation when encountering varied object configurations. In contrast, our approach maintains consistently strong performance, indicating improved robustness and generalization to diverse spatial arrangements. Detailed breakdown analyses are provided in Section B.6.

Lighting Condition. We evaluate the impact of our 3DGS-based lighting augmentation on policy generalization under varying illumination conditions, comparing with a variant using standard 2D image-space augmentation (*w/ 2D aug*), a variant without lighting augmentation (*w/o aug*), and the *SimObs* baseline. As shown in Figure 6 right, our method achieves the highest average success rate, indicating strong overall robustness to lighting variations. Detailed breakdowns (refer to Section B.7) across different lighting conditions reveal that the performance of the variant without augmentation is primarily driven by settings similar to the original scene and degrades under larger illumination shifts, while the variant with standard 2D augmentation improves over it but still exhibits noticeable sensitivity to illumination changes. In contrast, *SimObs* shows uniformly weak performance.

6 Conclusion

In this paper, we present DexMirror, a comprehensive real-to-sim-to-real framework that delivers photorealistic rendering, a streamlined calibration pipeline, and robust real-world generalization for dexterous manipulation. Across six tasks spanning diverse manipulation regimes, DexMirror consistently outperforms strong baselines by over +25% relative success rate, largely attributed to the improved visual fidelity and precise geometric consistency provided by our real-to-sim pipeline. A controlled downstream ablation further shows that tri-level reconstruction improves task success when calibration and policy training are held fixed. Looking ahead, we plan to extend DexMirror to broader robot platforms and environments, while incorporating more complex objects to further diversify simulation data and enable scalable robot data collection.

References

[1] Arthur Allshire, Hongsuk Choi, Junyi Zhang, David McAllister, Anthony Zhang, Chung Min Kim, Trevor Darrell, Pieter Abbeel, Jitendra Malik, and Angjoo Kanazawa. Visual imitation enables contextual humanoid control. *arXiv preprint arXiv:2505.03729*, 2025.

[2] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. Spie, 1992.

[3] Junhao Cai, Zetao Cai, Jiafei Cao, Yilun Chen, Zeyu He, Lei Jiang, Hang Li, Hengjie Li, Yang Li, Yufei Liu, et al. Internvla-a1: Unifying understanding, generation and action for robotic manipulation. *arXiv preprint arXiv:2601.02456*, 2026.

[4] Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025.

[5] Yuanpei Chen, Chen Wang, Yaodong Yang, and C Karen Liu. Object-centric dexterous manipulation from human motion data. *arXiv preprint arXiv:2411.04005*, 2024.

[6] Zoey Chen, Aaron Walsman, Marius Memmel, Kaichun Mo, Alex Fang, Karthikeya Vemuri, Alan Wu, Dieter Fox, and Abhishek Gupta. Urdformer: A pipeline for constructing articulated simulation environments from real-world images. *arXiv preprint arXiv:2405.11656*, 2024.

[7] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

[8] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

[9] Tianyuan Dai, Josiah Wong, Yunfan Jiang, Chen Wang, Cem Gokmen, Ruohan Zhang, Jiajun Wu, and Li Fei-Fei. Automated creation of digital cousins for robust policy learning. *arXiv preprint arXiv:2410.07408*, 2024.

[10] Prithwish Dan, Kushal Kedia, Angela Chao, Edward Weiyi Duan, Maximus Adrian Pace, Wei-Chiu Ma, and Sanjiban Choudhury. X-sim: Cross-embodiment learning via real-to-sim-to-real. *arXiv preprint arXiv:2505.07096*, 2025.

[11] Hongwei Fan, Hang Dai, Jiyao Zhang, Jinzhou Li, Qiyang Yan, Yujie Zhao, Mingju Gao, Jinghang Wu, Hao Tang, and Hao Dong. Twinaligner: Visual-dynamic alignment empowers physics-aware real2sim2real for robotic manipulation. *arXiv preprint arXiv:2512.19390*, 2025.

[12] Michael Goesele, Noah Snavely, Brian Curless, Hugues Hoppe, and Steven M Seitz. Multi-view stereo for community photo collections. In *2007 IEEE 11th international conference on computer vision*, pages 1–8. IEEE, 2007.

[13] Xiaoshen Han, Minghuan Liu, Yilun Chen, Junqiu Yu, Xiaoyang Lyu, Yang Tian, Bolun Wang, Weinan Zhang, and Jiangmiao Pang. Re^{3}sim: Generating high-fidelity simulation data via 3d-photorealistic real-to-sim for robotic manipulation. *arXiv preprint arXiv:2502.08645*, 2025.

[14] Tairan He, Zi Wang, Haoru Xue, Qingwei Ben, Zhengyi Luo, Wenli Xiao, Ye Yuan, Xingye Da, Fernando Castañeda, Shankar Sastry, et al. Viral: Visual sim-to-real at scale for humanoid loco-manipulation. *arXiv preprint arXiv:2511.15200*, 2025.

[15] Pu Hua, Minghuan Liu, Annabella Macaluso, Yunfeng Lin, Weinan Zhang, Huazhe Xu, and Lirui Wang. Gensim2: Scaling robot data generation with multi-modal and reasoning llms. *arXiv preprint arXiv:2410.03645*, 2024.

[16] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *SIGGRAPH 2024 Conference Papers*. Association for Computing Machinery, 2024.

[17] Arhan Jain, Mingtong Zhang, Kanav Arora, William Chen, Marcel Torne, Muhammad Zubair Irshad, Sergey Zakharov, Yue Wang, Sergey Levine, Chelsea Finn, et al. Polaris: Scalable real-to-sim evaluations for generalist robot policies. *arXiv preprint arXiv:2512.16881*, 2025.

[18] Guangqi Jiang, Haoran Chang, Ri-Zhao Qiu, Yutong Liang, Mazeyu Ji, Jiyue Zhu, Zhao Dong, Xueyan Zou, and Xiaolong Wang. Gsworld: Closed-loop photo-realistic simulation suite for robotic manipulation. *arXiv preprint arXiv:2510.20813*, 2025.

[19] Simar Kareer, Dhruv Patel, Ryan Punamiya, Pranay Mathur, Shuo Cheng, Chen Wang, Judy Hoffman, and Danfei Xu. Egomimic: Scaling imitation learning via egocentric video. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13226–13233. IEEE, 2025.

[20] Simar Kareer, Karl Pertsch, James Darpinian, Judy Hoffman, Danfei Xu, Sergey Levine, Chelsea Finn, and Suraj Nair. Emergence of human to robot transfer in vision-language-action models. *arXiv preprint arXiv:2512.22414*, 2025.

[21] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, George Drettakis, et al. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

[22] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

[23] Xuanlin Li, Kyle Hsu, Jiayuan Gu, Karl Pertsch, Oier Mees, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, Sergey Levine, Jiajun Wu, Chelsea Finn, Hao Su, Quan Vuong, and Ted Xiao. Evaluating real-world robot manipulation policies in simulation. *arXiv preprint arXiv:2405.05941*, 2024.

[24] Toru Lin, Kartik Sachdev, Linxi Fan, Jitendra Malik, and Yuke Zhu. Sim-to-real reinforcement learning for vision-based dexterous manipulation on humanoids. *arXiv preprint arXiv:2502.20396*, 2025.

[25] Toru Lin, Zhao-Heng Yin, Haozhi Qi, Pieter Abbeel, and Jitendra Malik. Twisting lids off with two hands. *arXiv preprint arXiv:2403.02338*, 2024.

[26] Haozhe Lou, Yurong Liu, Yike Pan, Yiran Geng, Jianteng Chen, Wenlong Ma, Chenglong Li, Lin Wang, Hengzhen Feng, Lu Shi, et al. Robo-gs: A physics consistent spatial-temporal model for robotic arm with hybrid representation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 15379–15386. IEEE, 2025.

[27] Zhao Mandi, Yijia Weng, Dominik Bauer, and Shuran Song. Real2code: Reconstruct articulated objects via code generation. *arXiv preprint arXiv:2406.08474*, 2024.

[28] Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. *arXiv preprint arXiv:2310.17596*, 2023.

[29] mikedh. trimesh. `https://github.com/mikedh/trimesh`, 2026. Accessed: 2026-02-28.

[30] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[31] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[32] Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024.

[33] PlayCanvas. supersplat. `https://github.com/playcanvas/supersplat`, 2026. Accessed: 2026-01-30.

[34] M Nomaan Qureshi, Sparsh Garg, Francisco Yandun, David Held, George Kantor, and Abhisesh Silwal. Splatsim: Zero-shot sim2real transfer of rgb manipulation policies using gaussian splatting. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6502–6509. IEEE, 2025.

[35] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024.

[36] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[37] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016.

[38] Kenneth Shaw, Ananye Agarwal, and Deepak Pathak. Leap hand: Low-cost, efficient, and anthropomorphic hand for robot learning. *arXiv preprint arXiv:2309.06440*, 2023.

[39] Ritvik Singh, Arthur Allshire, Ankur Handa, Nathan Ratliff, and Karl Van Wyk. Dextrah-rgb: Visuomotor policies to grasp anything with dexterous hands. *arXiv preprint arXiv:2412.01791*, 2024.

[40] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM siggraph 2006 papers*, pages 835–846. 2006.

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

[42] Tencent Hunyuan3D Team. Hunyuan3d 1.0: A unified framework for text-to-3d and image-to-3d generation, 2024.

[43] Tencent Hunyuan3D Team. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation, 2025.

[44] Tencent Hunyuan3D Team. Hunyuan3d 2.5: Towards high-fidelity 3d assets generation with ultimate details, 2025.

[45] Marcel Torne, Anthony Simeonov, Zechu Li, April Chan, Tao Chen, Abhishek Gupta, and Pulkit Agrawal. Reconciling reality through simulation: A real-to-sim-to-real approach for robust manipulation. *arXiv preprint arXiv:2403.03949*, 2024.

[46] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 13(4):376–380, 2002.

[47] Homer Rich Walke, Kevin Black, Tony Z Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, et al. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning*, pages 1723–1736. PMLR, 2023.

[48] Lirui Wang, Yiyang Ling, Zhecheng Yuan, Mohit Shridhar, Chen Bao, Yuzhe Qin, Bailin Wang, Huazhe Xu, and Xiaolong Wang. Gensim: Generating robotic simulation tasks via large language models. *arXiv preprint arXiv:2310.01361*, 2023.

[49] Kun Wu, Chengkai Hou, Jiaming Liu, Zhengping Che, Xiaozhu Ju, Zhuqin Yang, Meng Li, Yinuo Zhao, Zhiyuan Xu, Guang Yang, et al. Robomind: Benchmark on multi-embodiment intelligence normative data for robot manipulation. *arXiv preprint arXiv:2412.13877*, 2024.

[50] Jianfeng Xiang, Xiaoxue Chen, Sicheng Xu, Ruicheng Wang, Zelong Lv, Yu Deng, Hongyuan Zhu, Yue Dong, Hao Zhao, Nicholas Jing Yuan, and Jiaolong Yang. Native and compact structured latents for 3d generation. *Tech report*, 2025.

[51] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. *arXiv preprint arXiv:2412.01506*, 2024.

[52] Guowei Xu, Ruijie Zheng, Yongyuan Liang, Xiyao Wang, Zhecheng Yuan, Tianying Ji, Yu Luo, Xiaoyu Liu, Jiaxin Yuan, Pu Hua, et al. Drm: Mastering visual reinforcement learning through dormant ratio minimization. *arXiv preprint arXiv:2310.19668*, 2023.

[53] Zhengrong Xue, Shuying Deng, Zhenyang Chen, Yixuan Wang, Zhecheng Yuan, and Huazhe Xu. Demogen: Synthetic demonstration generation for data-efficient visuomotor policy learning. *arXiv preprint arXiv:2502.16932*, 2025.

[54] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *CVPR*, 2024.

[55] Zhecheng Yuan, Tianming Wei, Shuiqi Cheng, Gu Zhang, Yuanpei Chen, and Huazhe Xu. Learning to manipulate anywhere: A visual generalizable framework for reinforcement learning. *arXiv preprint arXiv:2407.15815*, 2024.

[56] Zhecheng Yuan, Tianming Wei, Langzhe Gu, Pu Hua, Tianhai Liang, Yuanpei Chen, and Huazhe Xu. Hermes: Human-to-robot embodied learning from multi-source motion data for mobile dexterous manipulation. *arXiv preprint arXiv:2508.20085*, 2025.

[57] Ruijie Zheng, Dantong Niu, Yuqi Xie, Jing Wang, Mengda Xu, Yunfan Jiang, Fernando Castañeda, Fengyuan Hu, You Liang Tan, Letian Fu, et al. Egoscale: Scaling dexterous manipulation with diverse egocentric human data. *arXiv preprint arXiv:2602.16710*, 2026.

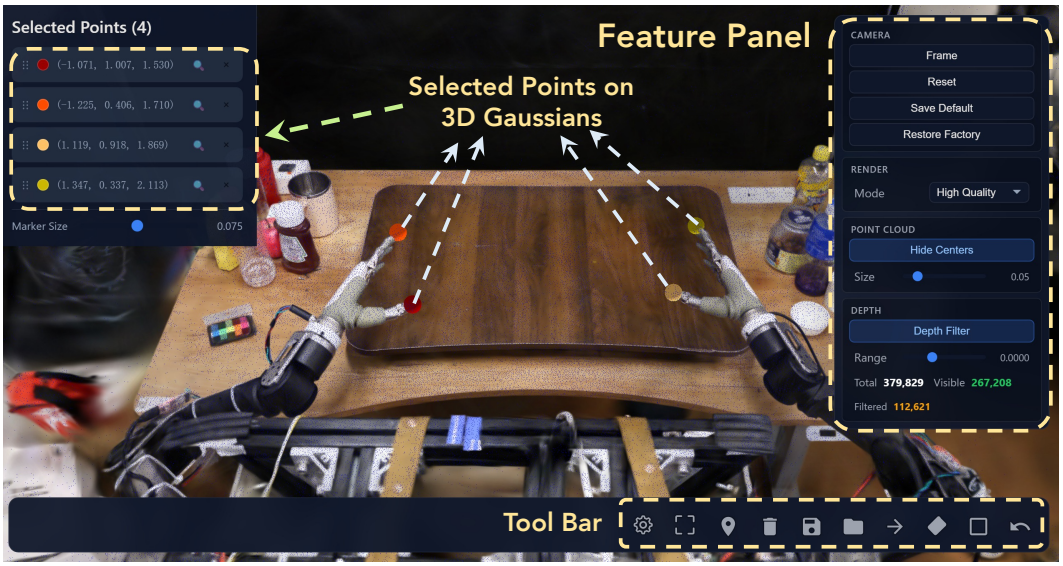


Figure 7: **Online annotation platform.** Users can jointly visualize point clouds and 3DGS rendering to select semantic keypoints for reliable registration.

A Additional Details on Method

A.1 Physics-Aware Asset Construction and Gaussian Rendering

Physics and rendering use separate representations. No predefined object meshes are required: for each manipulated object, we use AR Code (`https://ar-code.com/`) to extract a mesh, following the same mesh-extraction approach used by the Re^{3}Sim baseline. For regular objects such as cans, the extracted mesh serves as both the MuJoCo visual mesh and collision mesh. For objects with more complex geometry, such as the scanner, an automated primitive-fitting procedure approximates the mesh with MuJoCo primitives to obtain collision geometry. A vision-language model (Gemini 3.5 Flash) estimates a real-world prior for each physical parameter as a mean and variance; we sample from these priors for domain randomization during privileged-policy RL training. This avoids per-object parameter optimization while exposing the policy to plausible physical variation.

All contacts, rigid-body dynamics, and privileged-policy training use the generated MuJoCo assets. The simulator then provides each rigid body's pose, which we apply to its associated Gaussians before rendering the RGB observations used to distill the visuomotor policy. Gaussians do not participate in collision detection or dynamics. We retain Gaussian rendering because the mesh-rendered observations have a larger visual gap to real images, consistent with the weak *SimObs* results in Figure 5.

A.2 An Example of the Demonstration Synthesis Method

We take the task *PourBeer* as an example, in which the franka robot should first grasp a beer can and pour some beer into a cup. We draw a schematic diagram in Figure 8 Given a reference demonstration, we first decompose the full trajectory into several motion clips, including approaching the beer, grasping the beer, approaching the cup, and pouring beer into the cup. These clips are then categorized into *object-agnostic* and *object-interactive* motions. According to the definitions in Section 4.1, the two approaching phases are object-agnostic (colored in blue), while the grasping and pouring phases are object-interactive (colored in red). Given a new object configuration, we first transform the object-interactive clips according to the object transformation while preserving the relative pose between the hand and the object, and then synthesize the object-agnostic clips by interpolating the transformed hand (and object) trajectories between the corresponding starting and ending frames.

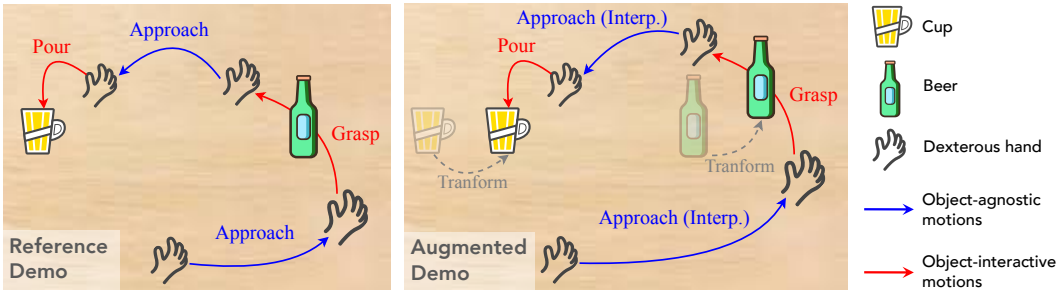


Figure 8: **Schematic diagram of the task *PourBeer*.** *Left*: Motion segmentation of the reference demonstration; *Right*: Motion segmentation of the augmented demonstration, which is derived from the reference. The original object configurations are shown with slight transparency, while the new configurations are displayed with solid filling.

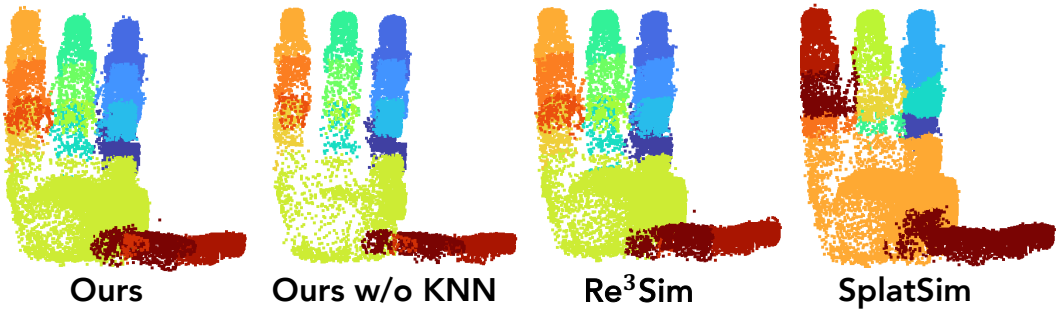


Figure 9: **Visualization of link-wise Gaussian segmentation results.**

B Additional Details on Experiments

B.1 Multi-View Data Collection

For background reconstruction, we capture approximately 200 images in the single-arm scene and 400 in the dual-arm scene along a 360-degree trajectory around the workspace. These sets include close-up views near the robot, especially the hand, to improve coverage under occlusion. For the object dataset, we use approximately 20 images when the scene contains few objects and 50 when it contains more objects. Images are captured at 1920×1080 resolution. Separately reconstructing the background and objects reduces occlusion artifacts because regions hidden in one dataset may be visible in the other.

B.2 Evaluation on Link-wise Segmentation

Moreover, we evaluate whether the proposed calibration pipeline can accurately segment Gaussians corresponding to individual robot links, especially on the dexterous hand. We compare our method against the aforementioned *GSWorld* and *Re*^{3}*Sim* baselines, as well as an ablated variant that removes the KNN refinement. The segmentation results of a LEAP Hand, shown in Figure 9, indicate that our method accurately assigns points to their corresponding links. Due to imperfect alignment, *GSWorld* and *Re*^{3}*Sim* can misassign points to neighboring links, with the former showing the most severe errors. In addition, removing KNN further causes many points to remain unassigned and thus be classified as background.

B.3 Calibration Details and Robustness

We annotate approximately 10 corresponding keypoints per arm, including the fingertips, the base of the palm, and three to four reference points along the arm. The hand and robot F-scores in Table 3 use a distance threshold of $\tau = 0.015$. To measure sensitivity to manual selection, we add Gaussian noise independently to each annotated keypoint and re-solve the closed-form Umeyama alignment.

Table 4: **Hand-alignment robustness to annotated-keypoint noise.** Each keypoint is perturbed with independent Gaussian noise before re-solving the SIM(3) alignment.

Noise standard deviation	Hand F-score
0 mm	0.981
2 mm	0.972
5 mm	0.936
10 mm	0.842

Table 5: **Per-stage cost of our single-arm pipeline.** Object asset generation scales with the number of objects.

Stage	Human time	Wall-clock time
Multi-view capture	$\sim$10 min	$\sim$10 min
Tri-level 3DGS	Unattended	$\sim$27 min
Object asset generation	Unattended	$\sim$5 min/object
Calibration with 3DGS viewer	$\sim$4 min	$\sim$4 min
Calibration without 3DGS viewer	$>$15 min	$>$15 min
Reference demonstration	$\sim$2 min	$\sim$2 min
Policy training	Unattended	$\sim$200 min

Table 6: **Wall-clock comparison of real-to-sim stages.** Times are approximate and refer to the corresponding single-arm setup.

Method	Scene reconstruction	Object reconstruction / asset	Calibration
Ours	Tri-level 3DGS: $\sim$27 min	AR Code: $\sim$5 min/object	$\sim$4 min
Re^{3}Sim	3DGS: $\sim$27 min	AR Code: $\sim$5 min/object	$\sim$10 min (marker) $\sim$0 min (ICP)
GSWorld	3DGS: $\sim$27 min	2DGS: $\sim$15 min/object	$\sim$10 min (marker)
SplatSim	3DGS: $\sim$27 min	3DGS: $\sim$10 min/object	$\sim$0 min (ICP)

The hand F-score remains 0.972 with 2 mm keypoint noise and 0.936 with 5 mm noise (Table 4). Five annotators independently calibrated the same scene, yielding a hand F-score of 0.975 ± 0.007. Multiple correspondences reduce sensitivity to individual click errors.

B.4 Human Effort Analysis on Real-to-Sim Pipeline

For a single-arm scene and task, Table 5 separates human effort from elapsed wall-clock time. Multi-view capture takes approximately 10 minutes, followed by approximately 27 minutes of unattended tri-level 3DGS reconstruction (30,000 iterations on one A100). Automatically generating physical object assets takes approximately 5 minutes per object. The real-time 3DGS viewer reduces interactive calibration to approximately 4 minutes, compared with more than 15 minutes without 3DGS guidance. Capturing and categorizing the reference demonstration takes approximately 2 minutes; subsequent simulation and augmentation add negligible human effort. Policy training takes approximately 200 minutes on eight A40 GPUs and runs unattended. Unlike real-world collection, the human effort for our one-time setup does not scale linearly with the number of training demonstrations.

Table 6 compares reconstruction, object-asset construction, and calibration across real-to-sim pipelines. Our tri-level model jointly reconstructs the scene and objects; Re^{3}Sim extracts object meshes with AR Code, GSWorld reconstructs objects with 2DGS, and SplatSim uses separate 3DGS models for scenes and objects. Re^{3}Sim and GSWorld use marker-based calibration, whereas SplatSim uses ICP without manual calibration. These entries describe the corresponding pipeline stages, rather than claiming that the independent stages run sequentially. For the downstream sim-to-real comparisons, the policy backbone (Diffusion Policy with a DINOv2 encoder), 100 training demonstrations, 1,000-epoch budget, and closed-loop leader–follower deployment interface are matched; policy training takes approximately 200 minutes on eight A40 GPUs and inference runs at 10 Hz. Real DP, trained on 100 real episodes, is a near-oracle reference rather than a matched real-to-sim baseline.

Table 7: **Spatial generalization under different object configurations.** Results are reported as successful trials over total trials.

Task	Method	Normal	Near	Far	All
PourBeer	**Ours**	**5/5**	**5/5**	**4/5**	**14/15**
	Scene-Rand	**5/5**	3/5	1/5	9/15
	Pose-Only	4/5	4/5	3/5	11/15
Bottle Handover	**Ours**	**4/5**	**3/5**	**3/5**	**10/15**
	Scene-Rand	4/5	3/5	1/5	8/15
	Pose-Only	2/5	2/5	1/5	5/15

B.5 Training Details

For the real-to-sim part, we train the 3DGS models on a single A100 GPU, following the default optimization settings of the original repository. The training runs for 30,000 iterations. For our tri-level formulation, we set the reconstruction weights for the three datasets as $\lambda_{\text{bg}} = 1.0$, $\lambda_{\text{comp}} = 0.1$, and $\lambda_{\text{obj}} = 10.0$. For the sim-to-real part, we train each policy using 8 A40 GPUs based on 100 demonstration trajectories, for a total of 1,000 epochs.

B.6 Detailed Generalization Analysis on Spatial Configuration

We compare our proposed object-level independent randomization against the scene-level global randomization (*Scene-Rand*) adopted in [56], as well as a pose-only augmentation variant (*Pose-Only*) that perturbs object and hand poses without preserving hand–object interactions, to evaluate the spatial generalization capability of the trained policies. The comparison is conducted on two tasks, *PourBeer* and *BottleHandover*, under three types of object configurations: *Normal*, where the relative distance between objects is approximately preserved ($\sim$20cm); *Near*, where objects are initialized in closer proximity (<12cm); and *Far*, where objects are placed farther apart (>30cm). The results (in Table 7) show that all methods perform similarly under the *Normal* configuration, where object distances remain close to the training distribution. However, under the *Near* and *Far* settings, *Scene-Rand* exhibits a clear performance drop, indicating limited spatial generalization. This is because keeping the relative poses between objects fixed during environment initialization limits variation in the collected action sequences. Consequently, large changes in object configuration at test time may result in out-of-distribution situations. In contrast, our independent randomization maintains strong performance, demonstrating improved robustness to variations in object placement. We further observe that *Pose-Only* also suffers from performance degradation under configuration shifts, especially on the more contact-rich *BottleHandover* task. This suggests that preserving consistent interaction and contact information is crucial for robust spatial generalization.

B.7 Detailed Generalization Analysis on Lighting Condition

We evaluate the impact of the 3DGS-based lighting augmentation on policy generalization under different illumination conditions. Specifically, we compare our full method against a variant without lighting augmentation (*w/o aug*), a variant with standard 2D random overlay augmentation (*w/ 2D aug*), and the *SimObs* baseline on the *PourBeer* and *BottleHandover* tasks. The scene-level bias in Section 4 is sampled as $\boldsymbol{\beta}_{\text{scene}} \sim \mathcal{U}([-0.25, 0.25]^3)$ once per episode. The object-level bias is sampled independently for each rigid body as $\boldsymbol{\beta}_{\text{object}} \sim \mathcal{U}([-0.15, 0.15]^3)$, and point-level noise follows $\boldsymbol{\epsilon}_{\text{point}} \sim \mathcal{N}(\mathbf{0}, 0.01\mathbf{I}_3)$. All perturbations are resampled each episode, kept fixed within that episode, and applied throughout visuomotor distillation. As shown in Table 8, our method achieves consistently high success rates across diverse lighting conditions, demonstrating strong robustness to illumination changes. The variant without lighting augmentation performs well under lighting similar to the original scene but degrades significantly under large illumination shifts, although it retains partial robustness due to the semantic consistency of the pretrained DINOv2 encoder. The variant with 2D augmentation improves over training without augmentation but still shows noticeable degradation under illumination changes. In contrast, *SimObs* exhibits uniformly low success rates, indicating that simple 2D overlay augmentation is insufficient to handle the gap between simulation and real-world illumination variations.

Table 8: **Illumination generalization under different lighting conditions.** *Original* denotes illumination similar to the collected multi-view images, *Natural* corresponds to daily lab lighting, and *Varying* represents illumination with varying hue values.

Method	Original	Natural	Red	Yellow	Green	Blue	Varying	All
Task: PourBeer								
Ours	**10/10**	**10/10**	**10/10**	**10/10**	**9/10**	**8/10**	**9/10**	**64/70**
w/ 2D aug	9/10	8/10	7/10	9/10	6/10	5/10	4/10	48/70
w/o aug	9/10	6/10	5/10	8/10	5/10	3/10	2/10	38/70
SimObs	5/10	4/10	5/10	6/10	6/10	4/10	3/10	33/70
Task: BottleHandover								
Ours	**8/10**	**7/10**	**8/10**	**8/10**	**6/10**	**5/10**	**5/10**	**47/70**
w/ 2D aug	6/10	5/10	7/10	6/10	5/10	4/10	3/10	36/70
w/o aug	7/10	5/10	3/10	6/10	4/10	2/10	2/10	29/70
SimObs	4/10	3/10	3/10	5/10	3/10	2/10	2/10	22/70

Table 9: **Real-world success in a directly scanned cluttered scene.**

Task	Clean scene	Cluttered scene
PourBeer	0.93	0.87
ServeCoffee	0.80	0.72
HangMug	0.68	0.63

Table 10: **Reconstruction quality with a complex background.**

Setting	PSNR (bg)	SSIM (bg)	PSNR (obj)	SSIM (obj)
Studio backdrop	30.16	0.9540	40.88	0.9964
Complex background	30.09	0.9535	40.67	0.9952

Table 11: **Real-world success under complex-background training and zero-shot transfer.** The zero-shot policy was trained with the studio backdrop.

Task	Studio backdrop	Complex background	Complex background (zero-shot)
PourBeer	0.93	0.92	0.85
ServeCoffee	0.80	0.81	0.72
HangMug	0.68	0.65	0.56

B.8 Evaluation in Cluttered and Complex-Background Scenes

To test multi-object clutter, we directly scan a workspace containing all objects used across the three Franka tasks, with the objects present on the table simultaneously. As shown in Table 9, real-world success decreases by 0.05–0.08 relative to the clean scene. Accidental collisions with irrelevant objects account for much of this drop.

We also remove the studio backdrop around the robot and reconstruct the resulting natural, complex background. The background and object reconstruction metrics in Table 10 remain close to those of the studio scene. When the complex background is included in training, task success likewise remains close to the original setting (Table 11). Zero-shot transfer to the unseen background yields a moderate decrease, indicating residual sensitivity to visual distribution shifts. The compositional Gaussian models, 3DGS-based lighting augmentation, and DINOv2 visual encoder jointly support this robustness.

Table 12: **Primary causes of the 16 failed diagnostic trials.**

Failure mode	Count	Fraction of failures
Object misalignment	6	37.5%
Post-grasp slip	3	18.75%
Collision	5	31.25%
Observed reconstruction artifact	2	12.5%

B.9 Articulated-Object Case Study

We evaluate an *OpenDrawer* task to test whether the pipeline can extend beyond rigid objects. We use URDFormer [6] to reconstruct the articulated asset, which is stable for the drawer's regular geometry. We capture the 3DGS model with the drawer open to expose more of its surface, then apply the robot link-wise segmentation procedure to bind drawer Gaussians to their corresponding links. These Gaussians move with the simulated drawer joint. The real-world task success rate is 0.75. This case study demonstrates support for a representative articulated task, while broader automatic mesh and joint reconstruction remains challenging.

B.10 Failure Analysis

We analyze a separate diagnostic set of 10 trials for each of the six tasks (60 trials in total), of which 16 fail. This diagnostic set is distinct from the evaluation protocol summarized in Figure 5. We assign each failed trial to one primary execution-failure category (Table 12). Object misalignment describes cases where two objects appear aligned in a single-view image but are separated in 3D because depth cues are unavailable. We assign the reconstruction-artifact category only when an artifact is directly observed, such as an error involving a transparent object or a detail missed in a capture blind spot and revealed after motion; generic static visual defects are not inferred from policy failure alone.

Object misalignment and collisions together account for nearly 70% of these failures, pointing to single-view spatial ambiguity and contact safety as the main remaining challenges. Post-grasp slip motivates improved contact control, while directly observed reconstruction artifacts motivate broader capture coverage.

B.11 Scope and Limitations

The articulated *OpenDrawer* case study in Section B.9 does not establish broad coverage of articulated categories: automatic mesh and joint reconstruction remains difficult, and deformable-object manipulation is limited by simulation fidelity and particle-based reconstruction. Multi-view reconstruction requires sufficient coverage; even with a 360-degree trajectory and targeted close-ups, blind spots can leave artifacts. Semantic segmentation is also required. Although moderate mask noise is tolerated, severe errors can leave residual artifacts or cause alignment error.

Manual keypoint selection and robot-camera calibration still impose setup costs. The 3DGS-guided interface reduces keypoint calibration to approximately 4 minutes, but each new scene requires capture and alignment. Robot-camera calibration is a standard step, and random-crop augmentation in the DINOv2 encoder mitigates small camera-viewpoint perturbations; neither removes the need to check camera alignment. Automated primitive fitting and VLM-prior domain randomization reduce physical mismatch, but simulated contacts can still differ from real contacts. Finally, closed-loop robot deployment requires appropriate safety checks.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Section 5.2, Section 4, Section 5, Section B

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Section 6

   Guidelines:

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

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: [N/A]

Guidelines:

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

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 3, Section 4

   Guidelines:

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

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: [Yes]

Guidelines:

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

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section B.5

   Guidelines:

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

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Section 5.3

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section B.5

   Guidelines:

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

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: [Yes]

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: Section 6

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: [N/A]

    Guidelines:

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

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: [Yes]

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [N/A]

    Justification: [N/A]

    Guidelines:

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

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: [N/A]

    Guidelines:

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

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: [N/A]

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: [N/A]

Guidelines:

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